

Review of Decentralized Reinforcement Learning for Multi-Agent Robotic Task Scheduling

Timileyin Okoya
Department of Computer
Science, Lead City
University, Ibadan,
Nigeria

Waliu Apena
Department of Computer
Engineering, The Federal
University of Technology,
Akure (FUTA), Nigeria

Folasade Dahunsi
Department of Computer
Engineering, The Federal
University of Technology,
Akure (FUTA), Nigeria

Olufunso Alowolodu
Department of
Cybersecurity, The
Federal University of
Technology, Akure
(FUTA), Nigeria

ABSTRACT

Multi-robot task scheduling is fundamental to managing multi-agent robotic systems (MARS) in unpredictable environments such as logistics, disaster management and autonomous exploration. Centralized task allocation scales poorly and adapts slowly to change, and reinforcement learning (RL) offers an alternative in which robots learn scheduling strategies through interaction. This review analyses 75 studies on multi-agent task allocation with RL, with particular attention to PettingZoo's Pistonball environment as a multi-agent RL (MARS) benchmark. It combines a bibliometric profile of the corpus, a side-by-side comparison of representative studies, a qualitative comparison of learning paradigms and a graded appraisal of the evidence behind five recurring claims. Fully decentralized learning remains difficult because concurrent policy updates produce non-stationarity, whereas centralized training with decentralized execution (CTDE) has proved more practicable; hardware-validated schedulers are the weakest-supported area. The review identifies the need for systematic comparison of CTDE-based algorithms for vision-based multi-robot task scheduling under identical conditions in the Pistonball environment.

General Terms

Artificial Intelligence, Robotics, Algorithms, Performance

Keywords

Multi-agent reinforcement learning, decentralized control, task scheduling, multi-robot systems, centralized training decentralized execution, PettingZoo, Pistonball

1. INTRODUCTION

Robots function as perception-information-processing-action systems that operate independently or under human control and have become essential in industries that demand accuracy, repetition and risk tolerance. While single-robot research is mature, current academic interest centres on Multi-Agent Robotic Systems (MARS), in which robots collaborate toward collective or individual goals. MARS offer scalability, fault tolerance and the capacity to address multiple facets of a task concurrently, although recent reviews also stress their open challenges [1]. Historical roots reach to the 1980s and 1990s through behaviour-based robotics and the ALLIANCE framework for cooperation and fault tolerance [2], building on earlier work on motion stability and control [3].

As MARS enter rapidly changing settings such as logistics, disaster management and exploration, task scheduling (the assignment of tasks to robots) has become a central concern. Traditional methods that rely on centralized planners or fixed rule sets encounter coordination bottlenecks and struggle with change [4], [5]. Reinforcement learning provides an alternative

in which an agent selects actions and receives rewards, thereby learning strategies through direct interaction with the environment [6]–[8].

Multi-agent reinforcement learning (MARL) extends these techniques to systems in which agents learn independently yet must coordinate to optimize a shared goal, typically a collective reward. MARL suits collaborative robotics in dynamic environments where pure programming fails, yet success depends on how non-stationarity, partial observability, scalability, credit assignment and communication limits are handled [9]–[13]. Across robotics, scheduling, cloud/edge robotics, path planning and swarm systems, MARL has moved from centralized or rule-based formulations toward decentralized execution combined with centralized training as the configuration best matched to real-world constraints [10], [14]–[17].

Reviews of flexible shop scheduling, robotic networks, cloud robotics, UAV task assignment and multi-robot task allocation stress the need for adaptive algorithms capable of online reassignment while respecting completion-time, throughput, energy and communication constraints [16], [18]–[21]. Decentralized MARL appears promising yet remains less explored than coordination and navigation [18], [22], [23].

Open problems persist. Most MARL approaches still rely on central control during training, remain domain-specific, or learn slowly in visually complex settings [24]–[26]. Mainstream actor-critic algorithms such as PPO [27] depend largely on extrinsic rewards and therefore suffer sample inefficiency and weak temporal coordination under continuous dynamics. This review maps the gaps in decentralized reinforcement learning for multi-robot task scheduling and supplies the evidentiary background for subsequent PhD research on MARL CTDE architectures in the PettingZoo Pistonball environment [28].

The review makes four contributions: a profile of the 75-source corpus (Section 4.1), a side-by-side comparison of representative studies (Section 4.2), an evaluation of learning paradigms against five criteria (Section 4.3), and a graded appraisal of the evidence behind five recurring claims (Section 4.4). Section 2 reviews the literature, Section 3 describes the method and CTDE formulation, and Sections 5 and 6 discuss gaps and conclude.

2. LITERATURE REVIEW: TASK SCHEDULING AND REINFORCEMENT LEARNING

The thematic review follows the chronological development of multi-robot task scheduling along six lines: non-learning techniques, reinforcement-learning foundations, game-

theoretic and hybrid approaches, domain-specific applications, system-level constraints of communication and scalability, and benchmarking infrastructure.

2.1 Traditional Approaches to Multi-Robot Task Scheduling

Early solutions relied on centralized controllers and deterministic rules that proved inadequate under uncertainty and non-stationarity, prompting a shift first to decentralized methods and later to machine learning. Zhang and Parker [29] showed that processor-scheduling heuristics applied to multi-robot coalition tasks permitted formal decomposition yet lacked flexibility in unpredictable settings. Lerman et al. [30] demonstrated that coordinated behaviour can emerge when robots estimate the local environment and select tasks independently, enabling multirobot foraging without centralized management. Bischoff et al. [31] introduced a two-step procedure of task construction followed by local optimization that outperformed simpler algorithms when ordering constraints are present, and later formulations added time windows and sequencing constraints [32], cloud-based joint placement, assignment and scheduling [33], and scheduling of transport and pick robots in logistics [34]. Genetic algorithms have likewise been applied: Shakeri et al. [35] reported superior efficiency and completion times in flexible manufacturing, while Wang et al. [36] combined genetic search with mathematical programming for production scheduling. All of these classical methods share a common limitation: they cannot learn from interaction with the environment. Reinforcement learning addresses that shortcoming.

2.2 Reinforcement Learning Foundations for Multi-Robot Coordination

Deep reinforcement learning (DRL) addresses these limits. Li et al. [6] combined Model Predictive Control with Graph Neural Networks for real-time path planning and task allocation, and Dai et al. [7] used Deep Q-Networks for fleet management that reduced cost and completion time in simulation. Multi-agent variants include cooperative transport with task prioritization and global communication [8], multi-agent PPO on a partially observable stochastic game that surpasses rule-based schedulers [24], and cloud-based decentralized Q-learning for warehouse robots, consistent with the architecture pursued in the present research programme [26].

Broader taxonomies classify MARL techniques as independent learners, actor-critic models, value-based algorithms, value factorization, communication-learning, consensus-based and hierarchical approaches [10], [11], [37]. The dominant practical trend is Centralized Training with Decentralized Execution (CTDE), which exploits centralized information during training while preserving decentralized execution [10], [14], [38]. Fully decentralized learning imposes a stricter condition: no exchange of actions, experience, parameters or observations may occur in either training or execution, so each agent treats the others as part of the environment [13]. The resulting simultaneous policy updates render the environment non-stationary for every agent [13], [39]; best-response-based schemes are one route to learning without centralized information [40]. This dichotomy underpins the architectural choices examined throughout the review.

The governing formalism is the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [41]. Each agent receives only a local observation; the complete state

remains hidden, so robust feature extraction and memory are essential. Convolutional Neural Network (CNN) extractors apply learnable filters across image patches [42], and LSTM networks maintain a belief state over observation histories through gated forget, input and output mechanisms [43]. Policy optimization commonly relies on Proximal Policy Optimization (PPO), which maximizes expected return under a clipped surrogate objective that prevents destructively large updates [27].

2.3 Game-Theoretic and Hybrid Approaches

Park et al. [44] introduced a decision-revision algorithm that allows robots to adjust assignments dynamically as the environment changes, thereby reducing resource contention; experiments on rubbish-collection tasks confirmed resilience to individual robot failures. Chakraa et al. [4] surveyed market-based, behaviour-based and optimization-based assignment methods and devoted particular attention to reinforcement-learning solutions.

2.4 Scheduling Applications Across Domains

In robotics, task scheduling typically encompasses group formation, computational offloading and the joint allocation of communication, computation and robot resources [16], [19]. Table 1 summarizes how four application domains interpret the scheduling problem and the design pressures each places on MARL.

Table 1. Scheduling domains and MARL design pressures

Domain	Core problem; why MARL helps; main objectives	Typical limitation; promising direction
Flexible shop scheduling	Dynamic multi-resource scheduling; handles decentralized dynamics; completion time and scalability	Reward and state design; CTDE, decentralized and hierarchical methods
Robotic networks	Task scheduling and allocation; adapts to real-time conditions; throughput, delay, energy	Communication delays often omitted; better edge-task coupling
Cloud/edge robotics	Resource pooling and offloading; learns under uncertain demand; cost, latency, heterogeneous service	Data and training demands; DRL resource allocation
UAV task assignment	Real-time dynamic assignment; responds to sudden task changes; responsiveness and robustness	Communication overhead; scalable multi-objective methods

Sources: [16], [18]–[20].

Additional recurring themes include fault tolerance [45], energy-aware meta-heuristics for robotic swarms [46], cloud-based load balancing [47] and bio-inspired methods such as Ant Colony Optimization adapted to Antarctic conditions [48]. Adaptive approaches include meta-learning for rapid policy transfer [25], graph-attention networks for spatiotemporal scheduling [5] and hierarchical frameworks under time and uncertainty constraints [49]. Zhu et al. [50] formulated multi-task MARL with CTDE and attention mechanisms, demonstrating flexible cooperation in manufacturing while underscoring that centralized execution still hinders practical scaling.

2.5 System Constraints: Communication and Scalability

Communication is simultaneously a coordination resource and a bottleneck. MARL communication can enlarge each agent's observation and improve collaboration, yet energy and bandwidth constraints demand careful design [51], [52]. Poorly designed communication can impair credit assignment and inflate decision complexity [15], [53]. Scalability is an intrinsic MARL challenge: state-action spaces grow and robustness declines as the number of agents and environmental complexity increase [54]. Parameter sharing, factorization, curriculum learning, selective communication and decentralized aggregation are among the principal remedies [54], [55]. These constraints determine whether architectures validated in simulation can transfer to physical robot fleets.

2.6 Benchmarking Environments: PettingZoo and Pistonball

Rigorous comparison of centralized and decentralized systems requires reproducible benchmarks. PettingZoo supplies a unified Python API for multi-agent environments and employs an Agent-Environment-Cycle design that reduces ambiguity and hidden experimental error [28]. Environments can be interleaved with popular MARL tool-chains without simulator lock-in [28], [56]. Cooperative, competitive and hybrid tasks are available, supporting tests of decentralization across a range of interaction structures [57], and game-like settings such as maze and board games are routinely used to train and compare agents [58], [59]. Compatibility with libraries such as RLlib facilitates algorithmic prototyping [60].

Pistonball is a fully cooperative physics-based environment in which agents coordinate vertical piston actions to drive a ball toward the left boundary [28], [61]. Each agent receives only local or neighbourhood observations, so success depends on emergent coordination under partial observability [28], [62]. Realistic physics raises coordination complexity above that of grid-world tasks, and both discrete and continuous action spaces with flexible agent counts are supported [62]. Although Pistonball is not itself a task-scheduling domain, it exhibits the critical characteristics of decentralized robotic coordination: local sensing, action coupling, delayed team-level outcomes and alignment of local actions with a collective goal [61]–[63]. It therefore serves as a suitable test-bed for learning coordination structure before transfer to domain-specific schedulers [64]. Parameter sharing improves coordination and sample efficiency [59], [65], recurrence aids independent learning under partial observability [57], and reward shaping together with communication design further enhance performance [66].

3. REVIEW METHODOLOGY AND FRAMEWORK

3.1 Literature Search and Screening Protocol

The review draws on 75 sources located through structured searches of Semantic Scholar, PubMed, IEEE Xplore and related databases. Four literature streams were covered: classical and optimization-based multi-robot task allocation; theory and algorithms of MARL; applications in manufacturing, cloud/edge robotics, UAV coordination and swarm robotics; and MARL benchmarking platforms, principally PettingZoo and Pistonball. As few reviews focus exclusively on MARL for robotic scheduling, the search was broadened to communication-efficient distributed learning,

federated RL, path planning, swarm coordination and sim-to-real transfer. Core algorithmic references [27], [42], [43] and the PettingZoo/Pistonball literature [28] were included by design. After filtering and deduplication the corpus spans 1997 to 2026 and comprises peer-reviewed articles, conference papers, preprints and two book chapters [2], [9]. For the analysis in Section 4, each source was coded by the authors to one primary thematic stream and one venue type.

3.2 Problem Formalization and Engineering Framework

Multi-robot task scheduling is formalized as a Dec-POMDP following Bernstein et al. [41] and consistent with the CTDE paradigm that dominates practical work [10], [14], [38]. The Dec-POMDP is defined by the tuple of agent set, state space, individual action sets, transition function, shared reward function, individual observation sets and observation function. Under CTDE each robot selects its action according to a policy conditioned solely on its local observation, eliminating any requirement for real-time exchange of observations, policies or goals and thereby addressing the scalability and communication concerns of Section 2.5. During training only, a centralized critic receives the global state and joint actions and supplies advantage estimates that stabilize learning. After training the critic is discarded and only the decentralized policies remain for deployment (Fig. 1). Policies are updated with the clipped surrogate objective of PPO [27]. Team reward may be decomposed into a global term that encourages collective progress (for example successful scheduling or, in Pistonball, leftward ball movement) and weighted local terms that credit individual contributions [50], [62]. Local observations are encoded by a CNN-LSTM feature extractor before being passed to the decentralized policy network.

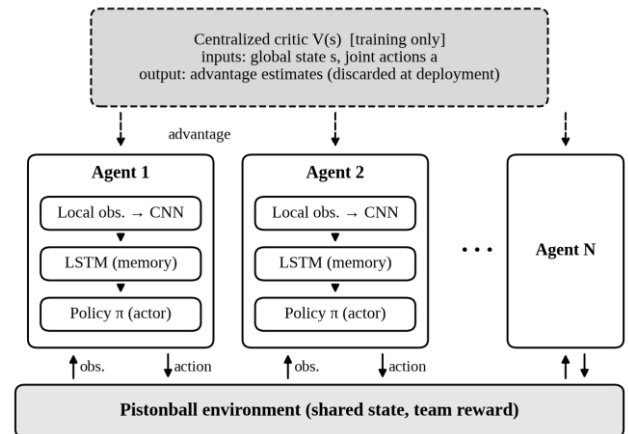


Fig 1: CTDE framework for decentralized task scheduling: the centralized critic is used only during training.

3.3 Evaluation Framework

Four evaluations structure Section 4. (a) A bibliometric profile counts sources by year, venue type and primary stream. (b) A side-by-side comparison of representative studies records method, setting, reported outcome and venue type. (c) A paradigm comparison rates centralized control, fully decentralized learning, CTDE and communication-based or hierarchical hybrids on five criteria: scalability, communication economy at execution, robustness to non-stationarity, handling of partial observability and deployment evidence. Ratings are ordinal (Low, Medium, High; higher is more favourable) and synthesize statements in the cited surveys rather than measured values. (d) Each of five recurring claims

receives an evidence-strength score from 1 to 10 reflecting the number and independence of supporting sources, the kind of evidence (survey, simulation or hardware validation) and the agreement among them. Scores are the authors' qualitative judgements.

4. RESULTS AND ANALYSIS

4.1 Corpus Profile

The corpus is recent: 63 of the 75 sources (84%) were published in 2020 or later and 46 (61%) in 2023 or later, with 2025 the single most represented year (Fig. 2). Journal articles dominate (46 sources, 61%), followed by conference papers (16, 21%), preprints (11, 15%) and book chapters (2, 3%). Preprints have not been peer reviewed. Applications of RL and MARL to robot scheduling and coordination form the largest stream (20 sources, 27%), followed by classical allocation methods and MARL theory (18 each, 24%), benchmarks and environments (11, 15%) and communication, scalability and sim-to-real work (8, 11%) (Fig. 3). The benchmark and sim-to-real streams are thinnest, mirroring the gaps in Section 5.2.

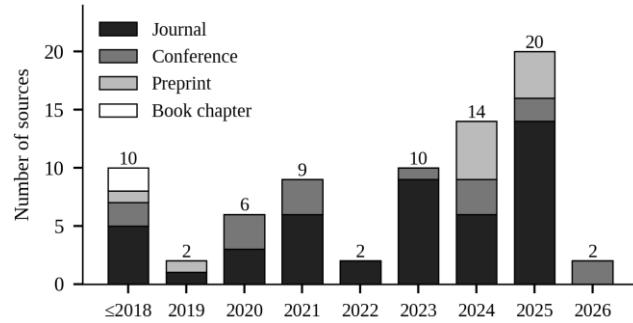


Fig 2: Publication year and venue type of the 75 sources.

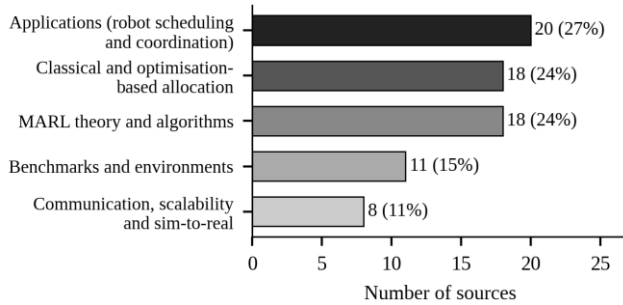


Fig 3: Sources by primary thematic stream.

4.2 Comparison of Representative Studies

Table 2 compares eleven representative studies discussed in Section 2 across classical, deep RL, MARL and game-theoretic families.

Table 2. Representative studies compared

Ref. (venue)	Family and method	Setting	Reported outcome
[31] (conf.)	Classical: task construction, then local optimization	Cooperative tasks, precedence constraints	Outperformed simpler algorithms
[35] (journal)	Classical: genetic algorithm	Flexible manufacturing	Higher efficiency, shorter completion times

Ref. (venue)	Family and method	Setting	Reported outcome
[44] (conf.)	Game-theoretic: decision revision	Rubbish collection	Less contention; resilient to robot failure
[5] (journal)	Deep RL: graph attention networks	Spatiotemporal scheduling	Learned scheduling policies
[6] (journal)	Deep RL: MPC with graph neural networks	Industrial automation, search and rescue	Real-time path and task adaptation
[7] (journal)	Deep RL: Deep Q-network fleet manager	Task assignment (simulation)	Lower cost and completion time
[8] (conf.)	MARL: task prioritization, global communication	Cooperative transport	Autonomy kept; unknown object weights handled
[24] (conf.)	MARL: multi-agent PPO	Collaborative scheduling	Beat rule-based schedulers on efficiency, throughput
[26] (preprint)	MARL: decentralized Q-learning, cloud-parallel	Warehouse robots	Efficiency gains
[50] (journal)	MARL (CTDE): multi-task, attention	Dual-resource job shop with robots	Flexible cooperation; centralized execution limits scaling
[25] (journal)	Meta-RL: meta-learning	Dynamic task scheduling	Rapid policy transfer

Three observations follow. First, the studies optimize different objectives (cost, completion time, throughput, contention, robustness), so reported gains are not directly comparable. Second, the studies summarized here do not share a benchmark, baseline or metric; each reports improvement over a rule-based or simpler method in its own setting. Third, among them only Zhu et al. [50] adopts CTDE explicitly, and it notes that centralized execution still hinders scaling. Direct comparison of CTDE variants under identical conditions is therefore missing, which motivates the Pistonball protocol in Section 5.2.

4.3 Comparison of Learning Paradigms

Fig. 4 rates four paradigms against the five criteria of Section 3.3. Centralized control scales poorly and loses robustness as agent numbers grow [10], [14], [67]. Fully decentralized learning needs no exchange of information but suffers non-stationarity [13], [39]. CTDE keeps execution decentralized while a centralized critic stabilizes training [10], [14], [38]. Communication-based and hierarchical hybrids improve cooperation under partial observability at a cost in bandwidth

and energy [51], [52]. Deployment evidence is weak for every learned paradigm [16], [68], [69]. No paradigm dominates on all criteria, but CTDE is the only one with no unfavourable rating other than deployment evidence, a weakness it shares with the other learned paradigms.

	Scalability	Comm. economy	Robust to non-stat.	Partial-obs. handling	Deployment evidence
Centralized control	Low	Low	High	Low	Med
Fully decentralized (independent learners)	High	High	Low	Med	Low
CTDE	Med	High	High	Med	Low
Communication-based / hierarchical hybrids	Med	Low	Med	High	Low

Fig 4: Qualitative comparison of paradigms (darker = more favourable).

4.4 Strength of Evidence for Key Claims

Five recurring claims emerge from the corpus, each supported to a different degree (Table 3, Fig. 5).

Table 3. Key claims and supporting evidence strength

Claim	Strength	Reasoning (representative sources)
Decentralized execution is the preferred deployment paradigm for multi-robot MARL	Strong (8/10)	Repeated across robotics, path-finding, connected-vehicle and MARL surveys, with strong practical rationale [10], [15], [70]
MARL is promising for dynamic task scheduling under uncertainty and heterogeneity	Moderate (7/10)	Supported across scheduling, robotic network, cloud robotics and UAV reviews, but mostly survey-level evidence [16], [18], [19]
Fully decentralized learning remains hard to optimize (non-stationarity, missing information)	Strong (8/10)	Strong conceptual agreement across decentralization and non-stationarity surveys [13], [39]
Communication design is critical to scheduling performance and scalability	Moderate (7/10)	Consistent across communication-MARL, distributed learning and swarm studies [51]–[53]
Real-world decentralized robotic schedulers are already well validated	Weak (2/10)	Corpus repeatedly highlights simulation dependence, reality-gap issues and limited generalization [16], [68], [69]

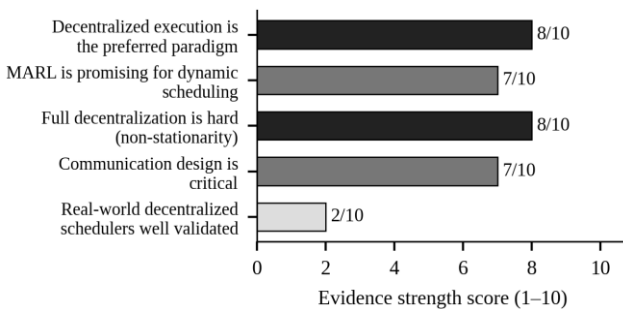


Fig 5: Evidence-strength scores for the five recurring claims

Decentralized execution and the persistence of non-stationarity under fully decentralized learning receive the strongest support. Claims about MARL’s utility for dynamic scheduling and the importance of communication design rest mainly on survey evidence. Validation of end-to-end, hardware-deployed decentralized schedulers is the weakest claim, which is consistent with the thin sim-to-real stream in Fig. 3.

5. DISCUSSION

5.1 Interpretation of the Evidence

The corpus supports the practical necessity of decentralization: centralized control introduces scalability and robustness problems that degrade coordination quality [10], [14], [67]. The principal limitation lies not in motivation but in the scarcity of deployable solutions. Most work consists of surveys, simulation studies or investigations of neighbouring domains such as path planning and wireless resource allocation; few benchmarks evaluate robotic scheduling on physical platforms [12], [16], [69]. Real-world successes of deep RL in robotics have been reviewed [71], yet the scheduling-specific evidence summarized here remains largely simulated. Sim-to-real transfer remains a persistent obstacle, and methods that mitigate the reality gap are still task-specific [68], [69]. Under fully decentralized learning the information bottleneck intensifies because agents lack knowledge of one another’s actions, experience and parameters [13]. Consequently, hybrid architectures that combine decentralized execution with centralized training, hierarchical decomposition, attention-based communication and distributed aggregation are repeatedly advocated [10], [22], [72], [73]. Because scheduling in heterogeneous robotic networks is inseparable from communication and resource management, a multi-layered decentralized control system that jointly addresses allocation, motion planning and edge/cloud resources is the most promising response [16], [19], [74], [75].

5.2 Research Gap and Future Directions

Three concrete directions emerge. First, hybrid decentralized-hierarchical methods that combine CTDE with attention-based or selective communication are frequently proposed yet rarely compared on standardized environments [10], [22], [73]. Second, scalable, communication-efficient coordination mechanisms must be evaluated under realistic bandwidth and latency constraints rather than idealized simulations [51], [52], [54]. Third, hardware-validated and standardized benchmarks remain scarce [68], [69], although PettingZoo and Pistonball are a useful starting point. The principal gap identified for the associated PhD programme is the absence of a decentralized RL framework augmented with self-supervised auxiliary-task learning for real-time task scheduling among heterogeneous multi-agent robots.

The planned response is a simulation of twenty heterogeneous agents in Pistonball. An independent-learner PPO baseline will be compared with CTDE PPO, with and without self-supervised auxiliary tasks, using five metrics: cumulative team reward, time for the ball to reach the goal boundary, sample efficiency (steps to a reward threshold), stability (variance across random seeds) and scaling with agent count.

5.3 Limitations of the Review

Four limitations apply. (i) Coverage is bounded by the databases and search terms used. (ii) Eleven sources (15%) are preprints that have not been peer reviewed. (iii) The ordinal paradigm ratings and 1–10 evidence scores are qualitative judgements, and each source is coded to a single primary stream although many span several. (iv) Heterogeneous

objectives and metrics across studies prevent a quantitative meta-analysis.

6. CONCLUSION

The review establishes the requirement for a decentralized RL technique for task scheduling in multi-agent robotic systems, grounded in MARL theory and reinforced by practical needs in robot scheduling, cloud and edge robotics, UAV cooperation and swarm systems. The corpus analysis shows a recent, journal-dominated literature in which CTDE is the most practicable paradigm, fully decentralized learning is limited by non-stationarity, and hardware-validated schedulers are the weakest-supported claim. The associated PhD programme will therefore build a simulation of twenty heterogeneous robots in the PettingZoo Pistonball environment, augment PPO with self-supervised auxiliary-task learning within a CTDE architecture, and assess performance and stability.

7. ACKNOWLEDGMENTS

The research enjoys the full support of the Institution-Based Research (IBR) intervention of the Tertiary Education Trust Fund (TETFund) Abuja to carry out this study in the Department of Computer Engineering, The Federal University of Technology, Akure (FUTA).

8. REFERENCES

- [1] Tejada, J. C., Toro-Ossaba, A., López-Gonzalez, A., et al. 2025. Multi-robot systems and soft robotics: Challenges and opportunities in a review. *Sensors*, 25(5), 1353.
- [2] Parker, L. E. 2008. Multiple mobile robot systems. In B. Siciliano & O. Khatib (Eds.), *Springer Handbook of Robotics* (pp. 921–941). Springer.
- [3] Owojori, A. O., Akingbade, K. F., Apena, W. O., and Ogunti, E. O. 2021. Stability control modelling under dynamic motion scenario of a differential drive robot. *Journal of Engineering Studies and Research*, 27(3), 64–73.
- [4] Chakraa, H., Guérin, F., Leclercq, E., and Lefebvre, D. 2023. Optimization techniques for multi-robot task allocation problems: Review on the state-of-the-art. *Robotics and Autonomous Systems*, 168, 104492.
- [5] Wang, Z., and Gombolay, M. 2020. Learning scheduling policies for multi-robot coordination with graph attention networks. *IEEE Robotics and Automation Letters*, 5(3), 4509–4516.
- [6] Li, Z., Shi, N., Zhao, L., and Zhang, M. 2024. Deep reinforcement learning path planning and task allocation for multi-robot collaboration. *Alexandria Engineering Journal*, 109, 408–423.
- [7] Dai, Y., Kim, D., and Lee, K. 2024. Development of fleet management system for task assignment to multiple robots using deep reinforcement learning. *Processes*, 12(12), 2921.
- [8] Shibata, K., Jimbo, T., Odashima, T., et al. 2023. Learning locally, communicating globally: Reinforcement learning of multi-robot task allocation for cooperative transport. *IFAC-PapersOnLine*, 56(2), 11436–11443.
- [9] Buşoniu, L., Babuška, R., and Schutter, B. 2010. Multi-agent reinforcement learning: An overview. In *Innovations in Multi-Agent Systems and Applications* (pp. 183–221). Springer.
- [10] Low, D., and Zhou, Y. 2025. Cooperative multi-agent reinforcement learning for robotic systems: A review. *Multiagent and Grid Systems*, 21, 96–123.
- [11] Liang, J., Miao, H., Li, K., et al. 2025. A review of multi-agent reinforcement learning algorithms. *Electronics*, 14(4), 820.
- [12] Canese, L., Cardarilli, G., Di Nunzio, L., et al. 2021. Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, 11, 4948.
- [13] Jiang, J., Su, K., and Lu, Z. 2024. Fully decentralized cooperative multi-agent reinforcement learning: A survey. *ArXiv*, abs/2401.04934.
- [14] Sharma, P., Fernandez, R., Zaroukian, E. G., et al. 2021. Survey of recent multi-agent reinforcement learning algorithms utilizing centralized training. In *Proceedings of SPIE*, 117462K–117462K-12.
- [15] Hu, J. 2025. A survey on reinforcement learning-based multi-agent path planning. *ITM Web of Conferences*, 78, 01015.
- [16] Alirezazadeh, S., and Alexandre, L. A. 2024. Survey on task allocation and scheduling in robotic network systems. *IEEE Internet of Things Journal*, 12(2), 1484–1508.
- [17] Evans, O., and Patel, M. 2025. Robotic swarm intelligence: Coordination and collaboration in multi-robot systems. *International Journal on Mechanical Engineering and Robotics*, 13(1), 17–22.
- [18] Xu, W., Gu, J., Zhang, W., et al. 2025. A survey on multi-agent reinforcement learning approaches for solving flexible shop scheduling problems. *Frontiers in Industrial Engineering*, 3, 1611512.
- [19] Afrin, M., Jin, J., Rahman, A., et al. 2021. Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23, 842–870.
- [20] Alqefari, S., and Menai, M. 2025. Multi-UAV task assignment in dynamic environments: Current trends and future directions. *Drones*, 9(1), 75.
- [21] Athira, K. A., Divya Udayan, J., and Subramaniam, U. 2024. A systematic literature review on multi-robot task allocation. *ACM Computing Surveys*, 57(3), 1–28.
- [22] Tarafdar, R. 2025. Reinforcement learning in embodied and multi-agent environments: Progress, challenges, and prospects. *International Journal of Innovative Science and Research Technology (IJISRT)*, IJISRT25MAR1376, 3183–3187.
- [23] Li, L., Yang, B., Chen, C., et al. 2025. Intelligent multi-robot exploration in non-exposed spaces: Methods and challenges. *Artificial Intelligence Review*, 58.
- [24] Gergely, M. I. 2024. Multi-agent deep reinforcement learning for collaborative task scheduling. In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART)* (Vol. 3, pp. 1076–1083).
- [25] Song, P., Chen, H., Cui, K., et al. 2025. Meta-learning for dynamic multi-robot task scheduling. *Computers & Operations Research*, 182.
- [26] Wu, S., Fu, L., Chang, R., et al. 2025. Warehouse robot task scheduling using reinforcement learning for maximizing operational efficiency [Preprint]. *TechRxiv*.

- [27] Schulman, J., Wolski, F., Dhariwal, P., et al. 2017. Proximal policy optimization algorithms. ArXiv, abs/1707.06347.
- [28] Terry, J. K., Black, B., Grammel, N., et al. 2021. PettingZoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 15032–15043.
- [29] Zhang, Y., and Parker, L. E. 2013. Multi-robot task scheduling. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2992–2998). IEEE.
- [30] Lerman, K., Jones, C., Galstyan, A., and Matarić, M. J. 2006. Analysis of dynamic task allocation in multi-robot systems. *The International Journal of Robotics Research*, 25(3), 225–241.
- [31] Bischoff, E., Meyer, F., Inga, J., and Hohmann, S. 2020. Multi-robot task allocation and scheduling considering cooperative tasks and precedence constraints. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 3949–3956). IEEE.
- [32] Suslova, E., and Fazli, P. 2020. Multi-robot task allocation considering time windows and sequencing constraints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 6909–6916). IEEE.
- [33] Zhang, C., and Shah, J. A. 2016. Multi-agent placement, task assignment, and scheduling in the cloud. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 3308–3314). AAAI Press.
- [34] Wang, H., and Chen, W. 2021. Task scheduling for transport and pick robots in logistics. *Autonomous Intelligent Systems*, 1(1), 17.
- [35] Shakeri, M., Yazdani, M., and Faal, H. 2025. A genetic algorithm for multi-robot task allocation and scheduling in flexible job shop environments. *International Journal of Production Research*, 63(8), 2673–2693.
- [36] Wang, H., Chen, W., and Wang, J. 2020. Coupled task scheduling for heterogeneous multi-robot systems. *Robotics and Autonomous Systems*, 131, 103560.
- [37] Oroojlooyjadid, A., and Hajinezhad, D. 2019. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53, 13677–13722.
- [38] Li, P., Li, S., and Wang, J. 2026. The cooperation and competition mechanisms in multi-agent reinforcement learning. *ITM Web of Conferences*, 84, 04004.
- [39] Papoudakis, G., Christianos, F., Rahman, A., and Albrecht, S. V. 2019. Dealing with non-stationarity in multi-agent deep reinforcement learning. ArXiv, abs/1906.04737.
- [40] Gabler, V., and Wollherr, D. 2024. Decentralized multi-agent reinforcement learning based on best-response policies. *Frontiers in Robotics and AI*, 11.
- [41] Bernstein, D., Givan, R., Immerman, N., and Zilberstein, S. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4), 819–840.
- [42] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- [43] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- [44] Park, S., Zhong, Y. D., and Leonard, N. E. 2021. Multi-robot task allocation games in dynamically changing environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 8678–8684). IEEE.
- [45] Kalempa, V. C., Piardi, L., Limeira, M., and de Oliveira, A. S. 2023. Multi-robot task scheduling for consensus-based fault-resilient intelligent behavior in smart factories. *Machines*, 11(4), 431.
- [46] Mokhtari, M., Sadati, S. M., Dorri-Nowkoohdan, M. M., and Shah-Hosseini, H. 2025. Energy-aware scheduling in robotic swarms: A meta-heuristic approach. *Journal of Ambient Intelligence and Humanized Computing*, 16(3), 1117–1131.
- [47] Sharma, K., Hassan, M. M., Pandey, S., et al. 2023. Cloud based multi-robot task scheduling using PMW algorithm. *IEEE Access*, 11, 146003–146013.
- [48] Kim, S., and Lee, H. 2023. Multi-robot task scheduling with ant colony optimization in Antarctic environments. *Sensors*, 23(2), 751.
- [49] Choudhury, S., Gupta, J. K., Kochenderfer, M. J., et al. 2022. Dynamic multi-robot task allocation in uncertain conditions and under temporal constraints. *Autonomous Robots*, 46(1).
- [50] Zhu, X., Xu, J., Ge, J., et al. 2023. Multi-task multi-agent reinforcement learning for real-time scheduling of a dual-resource flexible job shop with robots. *Processes*, 11(1), 26.
- [51] Zhu, C., Dastani, M., and Wang, S. 2022. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38, 1–48.
- [52] Cao, X., Başar, T., Diggavi, S., et al. 2023. Communication-efficient distributed learning: An overview. *IEEE Journal on Selected Areas in Communications*, 41, 851–873.
- [53] Cazenille, L., Toquebiau, M., Lobato-Dauzier, N., et al. 2025. Signalling and social learning in swarms of robots. *Philosophical Transactions of the Royal Society A*, 383, 20240148.
- [54] Liu, D., Ren, F., Yan, J., et al. 2024. Scaling up multi-agent reinforcement learning: An extensive survey on scalability issues. *IEEE Access*, 12, 94610–94631.
- [55] Myakala, P. K., and Kamatala, S. 2023. Scalable decentralized multi-agent federated reinforcement learning: Challenges and advances. *International Journal of Electrical, Electronics and Computers*, 8(6).
- [56] Oesterle, M., Grams, T., and Bartelt, C. 2024. DRAMA at the PettingZoo: Dynamically restricted action spaces for multi-agent reinforcement learning frameworks. In *Proceedings of the 57th Hawaii International Conference on System Sciences* (pp. 7810–7819).

- [57] Lee, K. M., Subramanian, S. G., and Crowley, M. 2021. Investigation of independent reinforcement learning algorithms in multi-agent environments. *Frontiers in Artificial Intelligence*, 5, 805823.
- [58] Arifi, H., and Zhi, G. 2025. Reinforcement learning based training for maze craze computer game. *Journal of Computer Sciences and Informatics*, 2(3), 191–202.
- [59] Adhikari, N., and Gu, A. 2024. Efficient learning in Chinese checkers: Comparing parameter sharing in multi-agent reinforcement learning. *ArXiv*, abs/2405.18733.
- [60] Kamthan, A. 2025. Learning to lead themselves: Agentic AI in MAS using MARL. *ArXiv*, abs/2510.00022.
- [61] Papadopoulos, G., Kontogiannis, A., Papadopoulou, F., et al. 2025. An extended benchmarking of multi-agent reinforcement learning algorithms in complex fully cooperative tasks. *ArXiv*, abs/2502.04773, 1613–1622.
- [62] Felten, F., Ucak, U., Azmani, H., et al. 2024. MOMAland: A set of benchmarks for multi-objective multi-agent reinforcement learning. *ArXiv*, abs/2407.16312.
- [63] Dai, W., Rai, U., Chiun, J., et al. 2025. Task allocation and scheduling of heterogeneous multiple robots using reinforcement learning. *IEEE Robotics and Automation Letters*, 10(3), 2654–2661.
- [64] Metelo, F., Racković, S., Ákos, P., and Soares, C. 2024. PeersimGym: An environment for solving the task offloading problem with reinforcement learning. *ArXiv*, abs/2403.17637.
- [65] Kölle, M., Erpelding, Y., Ritz, F., et al. 2024. Aquarium: A comprehensive framework for exploring predator-prey dynamics through multi-agent reinforcement learning algorithms. In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART)* (Vol. 1, pp. 59–70).
- [66] Kayal, A., and Ravimaran, S. 2026. Experimental evaluation of coordinated decision-making strategies based on multi-agent reinforcement learning. In *Proceedings of the International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSSES)* (pp. 1–9). IEEE.
- [67] Zhang, W. 2025. Cooperative path planning of multiple robots: Advances, challenges, and future trends. *Highlights in Science, Engineering and Technology*, 147, 359–364.
- [68] Zhao, W., Queralta, J. P., and Westerlund, T. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 737–744). IEEE.
- [69] Salvato, E., Fenu, G., Medvet, E., and Pellegrino, F. A. 2021. Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access*, 9, 153171–153187.
- [70] Hua, M., Qi, X., Chen, D., et al. 2023. Multi-agent reinforcement learning for connected and automated vehicles control: Recent advancements and future prospects. *IEEE Transactions on Automation Science and Engineering*, 22, 16266–16286.
- [71] Tang, C., Abbatematteo, B., Hu, J., et al. 2024. Deep reinforcement learning for robotics: A review of real-world success cases. *ArXiv*, abs/2408.03539.
- [72] Feriani, A., and Hossain, E. 2020. Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial. *IEEE Communications Surveys & Tutorials*, 23, 1226–1252.
- [73] Beltrán, E. T. M., Pérez, M. Q., Sánchez, P. M. S., et al. 2023. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*, 25(4), 2983–3013.
- [74] Chung, S.-J., Paranjape, A., Dames, P., et al. 2018. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4), 837–855.
- [75] Sun, L., Yang, Y., Duan, Q., et al. 2025. Coordination of multi-agents for various applications. *ArXiv*, abs/2502.14743.