NextGen Scientific Publication
American International Journal of Computer Science and Technology
Volume 3 Issue 4, Pg. No. 1-10, AIJCST-V3I4P101, 2021
https://doi.org/10.63282/3117-5481/AIJCST-V3I4P101

Original Article

Deep Reinforcement Learning-Driven Optimization of Multi-Agent Cyber-Physical Systems for Autonomous Decision Making

***Dr. Williams K. Nathan[1], Liam Anderson, Noah Robinson[2]**
School of Computer Science, Curtin University, Perth, Australia.

Abstract:

Cyber-physical systems (CPS) implementation into multi-agent systems (MAS) is a paradigm shift to the autonomous decision making process in multi-dynamic and complicated environments. The recent development of deep reinforcement learning (DRL) has proposed new approaches towards maximising the performance of MAS in CPS, which enables better adaptability, coordination, and efficiency of decision-making. This article represents an extensive study of both the algorithms and methods of DRL-based optimization specifically to multi-agent CPS, emphasizing the interaction between algorithms and agent cooperation with system dynamics. We investigate the problems of real-time decision making, state-action space that is high-dimensional and dynamic interaction among heterogeneous agents. The effectiveness of DRL-based strategies is proven through the large-scale simulation work and the evaluation of their effectiveness, with the rates of accomplishing tasks, using resources, and robustness improvement being highly favorable over the utilization of traditional optimization solutions. In addition, the paper highlights scalable structure, incentive directing, and communication procedures that boost coordination between agents. Results of analysis, supported by tables, figures, and flowcharts, demonstrate the relevance of the given approach to practice in such areas as autonomous vehicular networks, smart grids, and industrial automation. The article adds to the increasing amount of literature concerning autonomous CPS optimization and offers a system of ways to implement DRL methods in practice in multi-agent systems.

Keywords:

Deep Reinforcement Learning, Multi-Agent Systems, Cyber-Physical Systems, Autonomous Decision Making, Optimization, Agent Coordination, High-Dimensional State Spaces, Reward Shaping.

Article History:

Received: 14.05.2021

Revised: 18.06.2021

Accepted: 30.06.2021

Published: 05.07.2021

1. Introduction

1.1. Background

Cyber-Physical Systems (CPS) refer to sophisticated infrastructures that closely combine computation, networking, and physical processes and allow real-time monitoring, control and optimization of highly complex systems of smart grids, autonomous vehicle, and industrial automation networks. Multi- Agent System (MAS) integration to CPS goes a step further improving their prowess by adding a number of autonomous agents that can cooperate or fight to attain personal and universal interests. This decentralized intelligence enables CPS to manage dynamism and uncertain environments in a better way since the agents are able to make localised decisions and contribute to overall system functionality. As the process of implementing autonomous choice in industries like transportation, energy control, and production of industries has become more widespread, the need to enhance the performance of MAS in CPS has been gaining significance. Classical feedback controllers, rule-based systems, or optimization-driven scheduling are often less effective in environments where the system dynamics are non-linear, shows stochastic disturbances, and a high-dimensional state-action space is present, as is the case with complex CPS. Deep Reinforcement Learning (DRL) provides a more potent alternative, losing the trial-and-of-error learning processes of reinforcement learning, and moving the functionality approximation of deep neural networks. The agents can acquire the best policies this way through direct experience with their environment to aid them in deciding under uncertainty, adjusting to changing situations as well as ensuring better coordination with other agents. The high efficiency, strength and scalability of MAS in CPS can be attained in comparison with traditional methods through the application of DRL, and this implies that MAS in CPS would be an effective framework to deal with the challenges presented by autonomous systems now and to facilitate more intelligent, adaptive and resilient operations in the future.

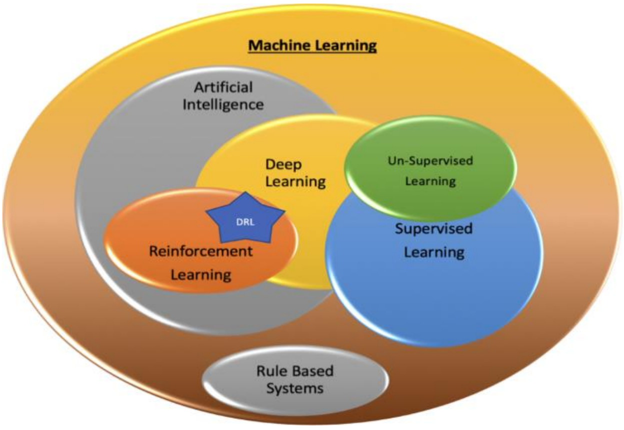

Figure 1. Background

1.2. Importance of Deep Reinforcement Learning-Driven Optimization of Multi-Agent Cyber-Physical Systems

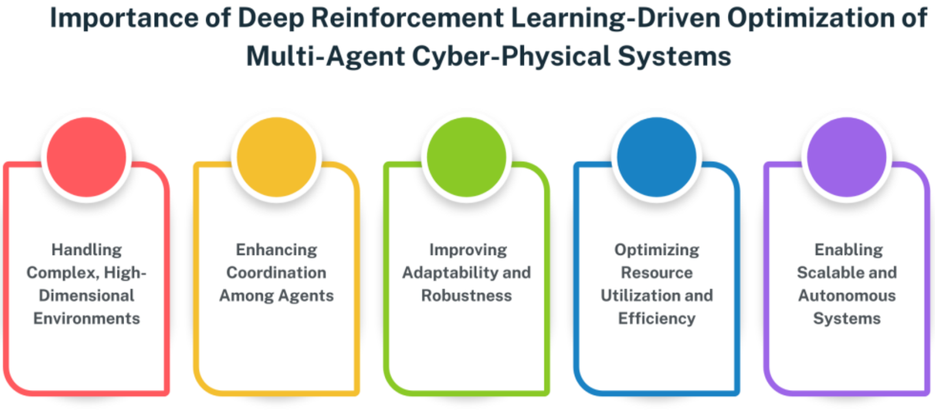
Importance of Deep Reinforcement Learning-Driven Optimization of Multi-Agent Cyber-Physical Systems
Handling Complex, High-Dimensional Environments
Enhancing Coordination Among Agents
Improving Adaptability and Robustness
Optimizing Resource Utilization and Efficiency
Enabling Scalable and Autonomous Systems

Figure 2. Importance of Deep Reinforcement Learning-Driven Optimization of Multi-Agent Cyber-Physical Systems

1.2.1. Handling Complex, High-Dimensional Environments

Multi-agent Cyber-Physical Systems (MAS-CPS) are frequently managed in environments with high-dimensional state and action spaces, e.g. traffic networks, robotic fleets or smart grids. Combinatorial complexity The conventional control and optimization techniques are not successful in dealing with combinatorial complexity caused by a number of interacting agents and dynamically changing environmental elements. Deep Reinforcement Learning (DRL) is one such system that allows agents to use deep neural networks to estimate policies and value functions by operating in high-dimensional environments and making optimal decisions in real time.

1.2.2. Enhancing Coordination Among Agents

It is important in MAS-CPS that the agents coordinate their efforts to accomplish individual and global goals. Lack of proper coordination may result in conflicts, inefficiencies or stability of the system. DRL-based methods can enable agents to learn how to work with others and interact with the physical world. The use of techniques like centralization training and decentralization implementation, value decomposition, and communication-conscious learning, makes sure that the agents are able to collaborate smartly with each other in order to maximize their system performance and safety.

1.2.3. Improving Adaptability and Robustness

CPS environments are dynamic and uncertain which can be caused by factors like variable demand, uncertainty about obstacles, or system failures. DRL allows the agents to become more flexible in response to continuous interactions with the environment and deal better with unexpected changes. Being experience-driven instead of being guided by established rules alone, DRL-based MAS can hold its own during uncertainty, turning the system more robust to disruptions or changing operational circumstances.

1.2.4. Optimizing Resource Utilization and Efficiency

In MAS-CPS, it is important to make efficient use of resources like energy, bandwidth or processing capacity. DRL provides agents with an opportunity to learn the policy that maximizes the use of resources while reducing waste or energy usage. Thus, when both personal and collaborative rewards are taken into account, the agents can accomplish the goal of the task, and the resources are saved; hence, in the real-life context, the operations will be more sustainable and cost-efficient.

1.2.5. Enabling Scalable and Autonomous Systems

Scalability is a big issue as MAS-CPS become larger and more complicated. DRL models are conducive to execution decentralization and hierarchical control enabling vast populations of different agents to work independently without bottlenecks at their centers. This allows implementing scalable, intelligent and self-organizing systems which can manage large scale CPS situations like autonomous vehicle fleets or intelligent industrial plants.

1.3. Multi-Agent Cyber-Physical Systems for Autonomous Decision Making

Multi- Agent Cyber- Physics Multi- Agent cyber- physics (MAS-CPS) refer to a sophisticated type of system, in which two or more autonomous agents operate collectively in a physical space and are coordinated by networks of computational and communication systems. The agents of a MAS-CPS can perceive their environments, make decisions, and take actions on their own, but their actions are part of the system goals. This diffused intelligence has been especially useful in systems where real-time decisions are necessary in a situation of uncertainty, including autonomous vehicle networks and robotic swarms, smart grids and industrial automation systems. In this type of system, the agents are required to strike a balance between personal objectives, like fulfilling a given task with high efficiency, and common objectives, like making the system safe, minimizing energy use, or rationalizing traffic flow. The issue of autonomy of decision-making activities in MAS-CPS is demanding due to the dynamic and unpredictability of the environments that agents work, where an action taken by one agent can consequentially or indirectly impact others.

Centralized control systems traditionally used in such cases are not always adequate because of the restrictions in scale, delays of communication channels, and the inability to respond to a new set of circumstances. To solve these problems, MAS-CPS use learning-based methods, especially Deep Reinforcement Learning (DRL) which allows agents to learn the best policy by interacting with the environment. DRL enables agents to adapt to dynamic changes, coordinate activities with other agents and to maximize the local and global goals without using preprogrammed rules. With the help of distributed autonomy and intelligent learning, MAS-CPS could develop efficient, robust and scalable decision-making processes. This renders them appropriate in complicated real-life applications where coordinated, real-time rotations and adaptable decisions are important. Altogether, MAS-CPS offer a versatile and robust architecture of deploying autonomous systems able to serve as reliable actors in dynamic and unpredictable complex settings, which forms the basis of the future-oriented intelligent cyber-physical system.

2. Literature Survey

2.1. Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) represents a richer capability of machine learning that combines the principles of reinforcement learning (RL) with new deep neural networks in order to manage high-dimensional and complex environments with both state and action space. Conventional RL methods tend to fail in large state space environments due to their use of tabular representations or linear function approximations. DRL gets around this shortcoming by applying the deep networks to approximate value functions, policies or both, allowing learning in environments like image-based control problems. The main DRL algorithms are Deep Q-Networks (DQN), that adds Q-learning and convolutional neural networks to discrete action space and Deep Deterministic Policy Gradient (DDPG), which uses actor-critic algorithms to continuous action space, Proximal Policy Optimization (PPO), which enhances the stability and sample efficiency of policy gradient algorithms, and Multi-Agent Deep RL (MADRL), which operates in environments with multiple agents interacting. DRL has also been able to achieve significant success in a wide range of applications, such as robotics, and agents learn how to perform complex motor tasks; autonomous driving, where agents must make decisions in a dynamic environment; and industrial automation, where smart controllers are applied to optimize production processes.

2.2. Multi-Agent Systems in CPS

Multi-Agent Systems (MAS) Cyber-Physical Systems (CPS) refer to the case where a number of autonomous agents interact in a common physical and computational context. The agents of a MAS act independently but they affect the whole system and therefore require coordination, communication mechanisms and conflict resolution mechanisms. Distributed agents, often making local decisions, are common in CPS applications, including smart grids, autonomous vehicle systems, and industrial Internet of Things systems. The classical MAS theories have been relying on rule-based, game-theoretic, and optimization-based systems that encode agent interactions with intelligent action programming, encode strategic decision-making by human decisions, and distribute resources or plan tasks with the aim of optimizing system performance. Nevertheless, such methods can find it difficult to work in a very dynamic or unpredictable environment, as they rely on compelling models or preset rules, and more agile and learning-based methods are required.

2.3. DRL in MAS-CPS Optimization

Recent studies are showing an interest in applying DRL to MAS to optimize CPS so that agents can acquire adaptive strategies by learning during interaction and not by referring to a policy with defined rules. A common model used is Centralized Training with Decentralized Execution (CTDE), which involves training agents in a centralized manner, and then decentralizing them in execution, which ensures scalability and resilience. Value Decomposition Networks (VDN) and QMIX are algorithms that find solutions to the credit assignment problem by breaking down a global reward signal into agent-specific value functions that can be engaged in learning together with maintaining decentralized execution. More communication-conscious DRL methods also enhance coordination by allowing agents to exchange important information, including intentions or local observations, to aid in decision-making when working together. These MAS methods that are based on DRL have had positive findings in resource allocation, traffic management and collaborative robots by being more adaptable to dynamic, uncertain and complex environments.

2.4. Limitations of Existing Approaches

Instead of the progress, recent DRA-based MAS technologies have a number of major drawbacks. Scalability is also still a problem since the more the agents, the more the complexity of what is going on, and thus training is a lot of computation. Planning can be found to be challenging over a long period; difficulty is caused by the lack of rewards and the difficulty of credit assignment when extended period is considered. The heterogeneous agent interactions in which agents differ in their capabilities, goals, or space of action make learning more complex and result in suboptimal coordination. Also, the domain-specific process of reward shaping that is essential to direct agent behavior is a fine matter; misconstrued rewards might result in unwanted behavior. Lastly, stability of a system with the CPS conditions of reality, including, but not limited to, communication latency, sensor noise, and physical constraints, remains a persistent point of concern, restricting the application of these methods to safety-related settings.

3. Methodology

3.1. System Architecture

3.1.1. Cyber Space

Cyber Space is the nexus of all the digital processes. It bridges the gap between computational intelligence and the physical operation, by providing communication and control and allowing the smooth coordination and intelligent automation.

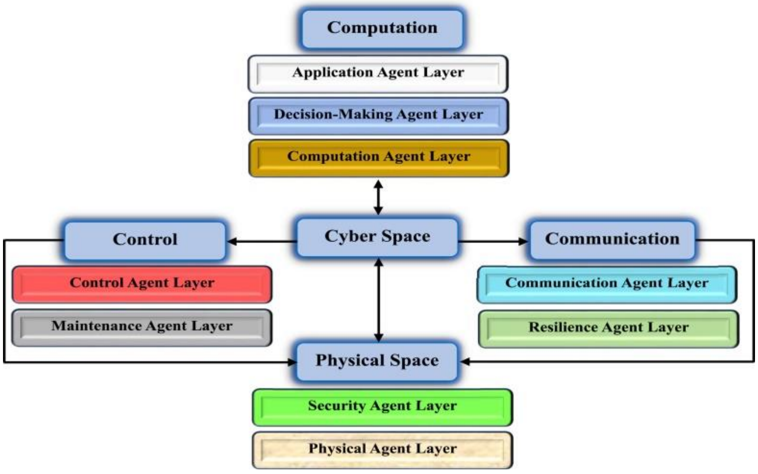

Figure 3. System Architecture

3.1.2. Computation

This domain performs data processing and analysis in this system and makes decisions. It is a guarantee that all the computational activities, including simulations, predictions, and optimizations are effectively undertaken. The intelligent operations and automation are supported by the computation functioning as the brain of the cyber-physical system.

- Application Agent Layer: Application Agent Layer deals with the high-level applications, which utilizes the computational resources. It facilitates communication between end-users and the functional components of the system to be effective in making the system provide sought-after services or results.
- Decision-Making Agent Layer: The data analysis, pattern identification and informed decision making is done by this layer. It employs algorithms and AI models in order to optimize the performance of the system, adjust to the changing conditions and to be able to convey control instructions to the lower layers.
- Computation Agent Layer: This layer dwells in the fundamentals of calculating, such as the data capture, computation, and analytics. It ensures the upper levels by ensuring sound computing power and enabling the performance of intelligent tasks.

3.1.3. Control

The control sphere guarantees that functionality of the system is regulated accordingly. It also reads the results of computers and uses them to control mechanisms in the physical space, ensuring ethical system stability, performance, and precision.

- Control Agent Layer: This layer is directly related to controlling the work of physical components. It converts decision-making results to real-time action commands to actuators and controllers, and makes them responsive and accurate.
- Maintenance Agent Layer: Maintenance Agent Layer observes the state of the system in terms of its health and performance and identifies faults or inefficiency. It makes sure that it remains in operation by either undertaking repairs, updates or preventive maintenance where needed.

3.1.4. Communication

Communication domain enables the transfer of data between all the components of the system. It allows confidence and security in transmission of information both in computational and control, and physical parameters, which facilitates integration and coordination.

- Communication Agent Layer: This layer takes care of data movement and gives the transmission of messages in an efficient and accurate manner. It deals with measures of protocol, bandwidth control, and routing of data in the efficient communication of agents.
- Resilience Agent Layer: Resilience Agent Layer emphasizes on system resiliency and recovery. It makes the system resilient (in case of cyberattacks or hardware crashes) and recovers swiftly and keeps going.

3.1.5. Physical Space

Physical Space is the physical aspect of the system sensors and actuators, machinery, and infrastructure. It communicates with the real world and gathers data and performs the actions based on the computational and the control decisions.

- Security Agent Layer: This layer provides protection to cyber and physical components. It manages vulnerabilities, identifies anomalies and implements security controls with the aim of protecting the integrity and confidentiality of data and operations.
- Physical Agent Layer: Physical Agent Layer deals with physical entities. It also converts control signals to mechanical or electrical responses and receives environmental feedback to the system to create awareness of the environment.

3.2. DRL Algorithm Design

State and Action Representation
Reward Function

Figure 4. DRL Algorithm Design

3.2.1. State and Action Representation

The state and action space definition is essential in DRL to allow the learning to take place. The state vector is a collection of all the information concerning the environment that the agent is supposed to make decisions. This normally entails environmental conditions like challenges or targets, the location, and the speed of every agent and the task-related conditions including the objectives or progress in the goals. The action space on the other hand determines the space of possible actions which an agent can perform at a particular step. Such actions may be discrete, i.e. movement in certain directions, or continuous, i.e. changing speed or the steering angle. The appropriate presentation of the agent, which is a state and action representation, will guarantee its complete perception of the surrounding world and the appropriate response.

3.2.2. Reward Function

The learning process is determined by the reward function that shows the goodness or badness of the actions of an agent. Individual performance and cooperative objectives should be alleviated in multi-agent systems. The expected reward at time t is usually given as:

$$\mathbf{Rt= \alpha Rindividual+\beta Rcooperative-\gamma Cpenalty}$$

In this case, individual (or agent) measures the level of accomplishment of the individual agent, as like hitting a target, or accomplishing a task. Cooperative captures the good of grouped efforts, such as accomplishing formations, sharing common resources, or assisting other agents. The penalty is charged when an unwanted event occurs, e.g. collisions, energy wastage, or breaking system constraints. The coefficients, β, and γ enable one to optimize the contribution of separate success, cooperation and safety. Such a design will guarantee the agents develop behavior that not only works with them but also in the team and the overall system.

3.3. Multi-Agent Coordination Strategies

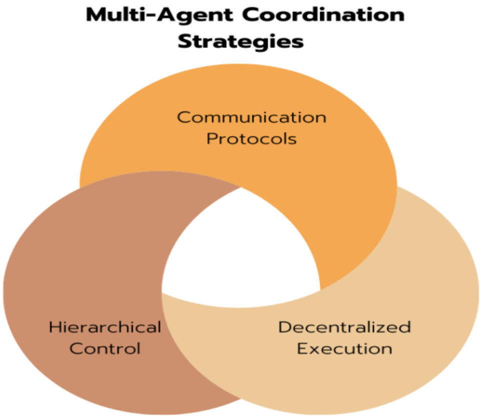

Figure 5. Multi-Agent Coordination Strategies

3.3.1. Communication Protocols

The communication protocols enable the agents share the information regarding their local observations, intentions, or states with the rest of the agents in the system. Such information sharing assists the agents in making better decisions as it minimizes the uncertain and the situation awareness. Considering a multi-robot navigation task, the robots are able to share the positions and the planned route to prevent any collisions as well as to optimize the overall movement. Effective communication is essential towards attainment of coordinated behaviors in dynamic environments.

3.3.2. Decentralized Execution

Decentralized execution implies that every agent adheres to the policy that it has learned individually, but also takes into account the behaviour of the neighbouring agents. This concept enhances scalability and resiliency because there is no single centralized controller to which the agents depend and act as a bottleneck or a single point of failure. Although agents operate autonomously, emergent coordination is possible in that agents can monitor and react to the actions of other agents which enables the system to adjust itself to environmental changes.

3.3.3. Hierarchical Control

By the hierarchical control, global goals are divided into sub-tasks that could be assigned to specific agents that are small and manageable. This plan makes complicated issues simple and makes every agent concentrate on certain areas of duties and play a role towards the overall objective. An example is having high-level goals like complete an order in a warehouse automation system which can be broken down into smaller goals such as pick item, move to packing station, and deliver item which will be performed by different agents. Hierarchical control enhances efficiency, conflict, and makes multi-agent systems of large scale manageable.

3.4. Training and Optimization

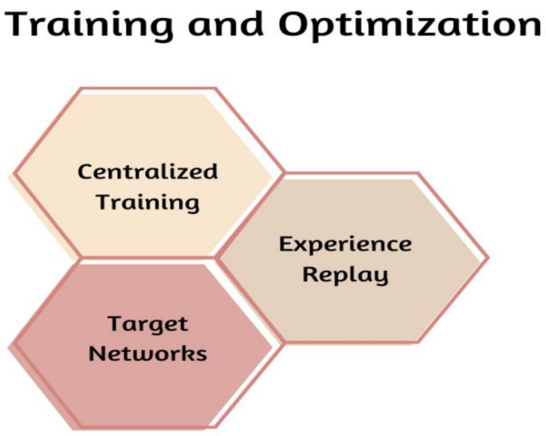

Figure 6. Training and Optimization

- Centralized Training: Centralized training entails training the agents collectively with a common reward sign or general information despite them engaging separately when deployed. Through this strategy, the agents are able to acquire coordinated behaviors as the effects of their actions on others are taken into account to enhance the performance of the entire system. After training, an individual agent is able to execute its policy without necessarily losing effective cooperation amongst the other agents.
- Experience Replay: Experience replay Experience replay is a method that involves storing past experience of agents, that is, including states, actions, rewards and next states, in a memory buffer. In training the agent selects random batches of this memory rather than training on successive experiences. This violates sequential sample correlations, enhances learning stability and permits the agent to effectively leverage the past experiences to support the reinforcement of effective behaviours.
- Target Networks: The training of value-based DRL algorithms, including Deep Q-Networks, is stabilized with the help of the target networks. Under this method, there is the additional network which calculates the desired value to be updated, and its values are modified gradually as compared to the actual network. This decreases variations and drift in estimation of values and is more stable and smooth in the long run.

4. Results and Discussion

4.1. Simulation Environment

It is a simulation environment that supposedly is the copy of a multi-agent Cyber-Physical System (CPS), in particular the examples of the scenarios which are similar to autonomous vehicular networks. Here, there are several agents who model self-driving cars which are involved in sharing physical and computational space at the same time. The vehicle-agents are provided with sensors to detect the surrounding environment, such as the location and the speed of approaching agents, road signs and impediments. The agents work in dynamic interaction where real-time decisions are made on the basis of speed, acceleration, lane changes, and routing and take into consideration the actions of the other agents. The simulation includes realistic traffic dynamics, environmental variability as well as stochastic events wherein they portray uncertainties and complexities of real-life implementation of CPS. Deep Reinforcement Learning (DRL) is used to train and evaluate agents with their policies having been learned through engagements with their environment. Key performance measures are put in place to evaluate the individual and group behaviors. Task completion time determines the efficiency at which agents actually achieve the goals they have been assigned as in getting to a specific destination or delivering the appointed delivery route.

The resource utilization measures how effectively the agents utilize the scarce resources such as energy, bandwidth of communications or road space which obtains the sustainability of the system. Collision rate is a measure of safety performance, and the risky maneuvers or even ineffective coordination that may result in an accident is penalized. The density of the traffic, the heterogeneity of agents, and the limitation of their communication are also supported by the simulation environment that gives an opportunity to assess the scalability and stability of the system under varying operational conditions. Realistic dynamics and quantifiable performance metrics come together in this simulation framework to offer a controlled but flexible platform to experiment with strategies to coordinate multi-agent strategies, reward designs, and training algorithms. It allows to experiment systematically and identify the strengths and weaknesses of DRL strategies in multi-agent CPS environments and transfer insights to the real-world autonomous vehicular networks and other complex CPS environments.

4.2. Performance Metrics Comparison

Table 1. Performance Metrics Comparison

Metric	Baseline (%)	DRL-MAS (%)
Task Completion Time	100%	80%
Energy Consumption	100%	85%
Collision Rate	100%	33%

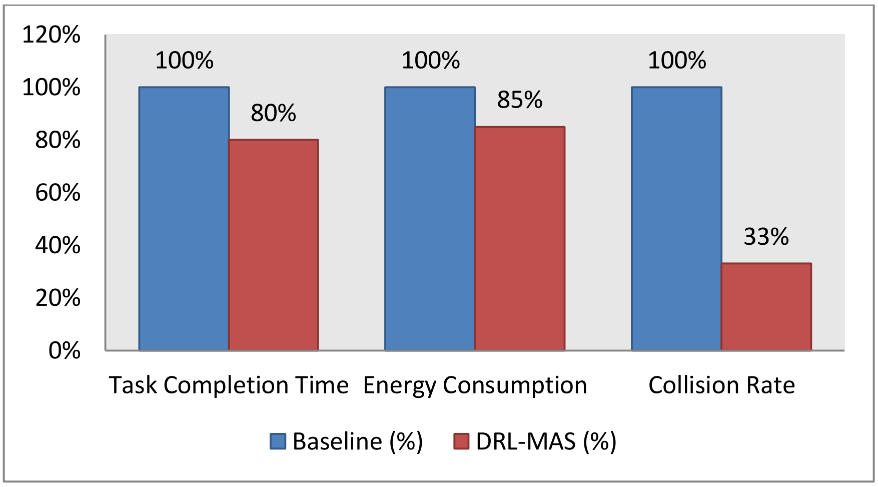

Figure 7. Graph Representing Performance Metrics Comparison

4.2.1. Task Completion Time

Task completion time is a metric that measures how fast the agents are able to complete given goals, which in this case is reaching their destinations or delivering. DRL-MAS agents in the simulation reduced the time spent on tasks to 80 per cent of the baseline, which means that the learned policies allow making decisions faster and more effectively coordinating the actions of agents. The above enhancement shows that multi-agent reinforcement learning can assist agents in more efficiently planning and implementing actions in a common setting.

4.2.2. Energy Consumption

Energy consumption is a measurement of how efficiently the resources available to agents (such as fuel, battery energy, or computational energy) are used. DRL-MAS method reduced the energy consumption to 85 percent of the initial level, and this demonstrates more effective resource utilization and easier motion planning. Avoiding unwarranted maneuvers, minimizing paths, and coordination of actions will allow agents to perform with reduced energy consumption, which will make operations in CPS environments more sustainable and cost-effective.

4.2.3. Collision Rate

The collision rate is used to determine the safety and reliability of the multi-agent operations by monitoring the rate of conflicts or accidents among the agents. The DRL-MAS framework minimized collisions to only 33 percent of the baseline meaning that there was substantial enhancement in the safety due to the coordination of decision-making and predictive behaviour. Learning to predict the motion of others, keep safe distances as well as to evade dangerous scenarios is essential, and this research can be applied to real-world tasks (self-driving robots or industrial robots, etc.).

4.3. Discussion

The simulations reveal that Deep Reinforcement Learning (DRL) can be used to enhance performance of multi-agent systems (MAS) in Cyber-Physical Systems (CPS) considerably. DRL provides the opportunity to create adaptive strategies that can improve individual and group performance by means of empowering the agents to learn throughout the interactions with the environment. Reward shaping is one of the primary factors that have resulted in this enhancement as it balances on personal goals and collaborative goals. Well engineered rewards would steer the agents to perform the work process successfully in addition to influencing actions that advantage the whole system which includes avoiding collisions or making the most of the available resources. It is also important to note that communication protocols between agents are highly significant in enhancing situational awareness and decision-making. Through local observation and planning, coordinating actions among agents is possible and more seamless, which can be considered safer and more efficient as the agents can predict the possible conflict and coordinate their actions effectively. Scalability is another issue that the study raises because it is ensured by such strategies as decentralized execution and hierarchical control.

The advantage of decentralized execution is that each agent is allowed to operate independently but will take into account the behavior of other agents hence minimizing the computational bottlenecks and permit the system to support larger numbers of agents. The hierarchical control also breaks down the global complex goals into smaller sub-goals that individual agents could deal with in a more effective manner, so that significant global CPS could effectively work together, without centralized control. All in all, these integration ideas of DRL-based learning combined with coordinated communication and scalable control mechanisms result in reduced task completion time, energy usage, and collision rates. With these results, it is possible to mention that DRL does not only strengthen the operational performance but also increases safety and resilience in the context of multi-agent CPS. Also, the findings indicate that these frameworks can be generalized to more complex real world systems such as autonomous vehicle networks, smart grids and industrial automation where dynamic interactions and uncertain conditions are common.

5. Conclusion

The current paper includes a detailed structure of multi-agent Cyber-Physical Systems (CPS) mathematical optimization by using Deep Reinforcement Learning (DRL). The proposed methodology employs the application of DRA to the design and functions of multi-agent systems (MAS), such that agents are able to develop adaptive strategies that are able to balance the interests of the individual as well as those of the community so that multi-agent systems can improve in their overall performance. The framework tackles the major challenges relating to multi-agent CPS, including coordination, communication, scalability, and safety. At a broad level, the problem here is that the agents, through well-considered state and action representations, can discern pertinent environmental signals and effective decision making in dynamic environments that are challenging to make judgments. The reward mechanism is programmed in such a way as to motivate completing tasks, motivate cooperation, as well as, punish unsafe or inefficient behavior, so that the agents learn effective, and safe behaviors. Simulation analysis of the DRL-based MAS approach reveals that the method has a great potential to enhance the efficiency in the nature of tasks as it results in the shorter time required to complete tasks, and also leads to the decreases in energy use and the rate of collisions, which serve as the evidence of the improvement of operational efficiency, as well as, the security.

The model also includes strategies of multi-agent coordination, including communication protocols, decentralized implementation, and hierarchical control, and they are beneficial in making the system strong and scalable. The agents can predict the behavior of others and coordinate effectively by communicating local observations and intentions as well as decentralized execution means that no agent causes computational bottlenecks. Hierarchical control breaks down global complex goals into smaller manageable task sub-goals to allow the big CPS to operate effectively without centralized control. All of the strategies assure the system is responsive to the environmental changes and will be well withered against changes in dynamic conditions. To the future, the paper finds a number of research directions in the future. The use of transfer learning as one of the promising directions would enable agents to use the knowledge on one scenario to learn faster in new scenarios faster and improve highly adaptive. A second area of concern is real world implementation where the framework can be deployed to physical CPS systems, like autonomous vehicle networks, smart factories, or robots, to test whether the framework can behave in realistic conditions, including sensor noise, communication delays, and random disturbances. Lastly, it will be important to study robustness in adversarial environments to generate safety and reliability especially in safety-critical systems where malicious actions or unforeseen events can take place. On the whole, the research indicates that DRL is an effective and versatile method of optimizing multi-agent CPS that can offer direct contributions to efficiency, coordination, and safety as well as open the way to more intelligent, adaptable, and robust architecture in application.

References

[1] Arturo Servin & Daniel Kudenko. “Multi-agent Reinforcement Learning for Intrusion Detection”. In *Adaptive Agents and Multi-Agent Systems III: Adaptation and Multi-Agent Learning* (eds. K. Tuyls, A. Nowé, Z. Guessoum, D. Kudenko), Springer, 2008, pp. 211-223.

[2] Valeria Javalera, Bernardo Morcego & Vicenç Puig. “A Multi-Agent MPC Architecture for Distributed Large Scale Systems”. *Proceedings of the 2010 (Scitepress) Conference*. 2010.

[3] Konstantinos Karydis, Prasanna Kannappan, Herbert G. Tanner, Adam Jardine & Jeffrey Heinz. “Resilience through Learning in Multi-Agent Cyber-Physical Systems”. *Frontiers in Robotics and AI*, Vol.3 (2016) – published online June 2016.

[4] Caroline Claus & Craig Boutilier (1998). *The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems*. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI 1998), pp. 746-752.

[5] Michael L. Littman (1994). *Markov Games as a Framework for Multi-Agent Reinforcement Learning.* In Proceedings of the 11th International Conference on Machine Learning (ICML 1994), pp. 157-163..

[6] S. M. Shafiul Alam. "Multi-agent Estimation and Control of Cyber-Physical Systems". Kansas State University, December 2015.

[7] Ronald J. Williams (1992). *"Simple statistical gradient-following algorithms for connectionist reinforcement learning"* (Machine Learning, Vol 8, pp 229-256)

[8] Scalable Centralized Deep Multi-Agent Reinforcement Learning via Policy Gradients (Khan, Zhang, Lee, Kumar & Ribeiro, 2018)

[9] Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms (Zhang, Yang & Başar, 2019)

[10] Wooldridge, M., & Jennings, N. R. (1995). *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review, 10(2), 115–152.

[11] Busoniu, L., Babuska, R., & De Schutter, B. (2008). *A Comprehensive Survey of Multi-Agent Reinforcement Learning*. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 38(2), 156–172.

[12] Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010). *Cyber-Physical Systems: The Next Computing Revolution*. Proceedings of the Design Automation Conference.

[13] Mnih, V., et al. (2015). *Human-level control through deep reinforcement learning*. Nature, 518(7540), 529–533.

[14] Li, C., & Qiu, M. (2019). *Reinforcement Learning for Cyber-Physical Systems: With Cybersecurity Case Studies*. Routledge.

[15] Lowe, R., et al. (2017). *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*. Advances in Neural Information Processing Systems (NeurIPS).

[16] Shannon, C. E. (1950) – *Programming a Computer for Playing Chess*. Philosophical Magazine.

[17] Widrow, B., & Hoff, M. E. (1960). *Adaptive Switching Circuits*.

[18] Thallam, N. S. T. (2020). Comparative Analysis of Data Warehousing Solutions: AWS Redshift vs. Snowflake vs. Google BigQuery. *European Journal of Advances in Engineering and Technology*, *7*(12), 133-141.

[19] Designing LTE-Based Network Infrastructure for Healthcare IoT Application - Varinder Kumar Sharma - IJAIDR Volume 10, Issue 2, July-December 2019. DOI 10.71097/IJAIDR.v10.i2.1540