Swarm Robotics Optimization Using Deep Q-Learning for Cooperative Search and Rescue Missions

Abdallah jarrah[1] and Ghadeer abu asfar[2]

[1] Department of computer science, Faculty of Information Technology and computer science, Yarmouk university, Irbid, Jordan
[2] Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Jordan

ABSTRACT

Swarm robotics has emerged as a promising field for addressing complex real-world challenges by enabling coordinated behavior among multiple autonomous robots. In search and rescue operations, traditional methods often face limited scalability, inefficient exploration, and a lack of adaptability to dynamic and unpredictable environments. Rule-based and heuristic approaches frequently struggle with real-time decision-making and coordination in large, multi-agent systems. To overcome these limitations, this paper proposes a method DQLSRO integrates Deep Q-Learning (DQL) into swarm robotics optimization (SRO) for cooperative search and rescue missions (SRM) to develop an adaptive, decentralized framework where robots learn optimal policies. Each robot utilizes a Deep Q-Network to learn and adapt its actions autonomously. Communication among robots enables information sharing, allowing dynamic policy updates and coordinated decision-making. The framework incorporates multi-agent reward functions to maximize coverage, minimize time to locate victims and avoid obstacles. Experimental results demonstrate that the DQL-based swarm outperforms traditional methods, achieving a 30% reduction in mission completion time and a 25% increase in victim detection rate. The DQLSRO also exhibits resilience to robot failures and communication disruptions. In conclusion, integrating Deep Q-Learning into swarm robotics provides a robust and efficient solution for cooperative search and rescue operations, addressing key limitations of traditional methods in emergency response scenarios.

Keywords: Swarm Robotics, Deep Q-Learning, Search and Rescue Operations, Multi-Agent Systems, Cooperative Decision-Making, Adaptive Framework, Autonomous Robots.

1. Introduction

Inspired by the collective behaviour in natural systems, such as ant colonies and bird flocks, swarm robotics has emerged as a revolutionary field in robotics [1]. It consists of deploying multiple autonomous robots that can perform complex tasks which are difficult or impossible for a single robot to achieve collaboratively [2]. By exploiting the decentralized control and local interactions, swarm robotics systems achieve scalability, robustness, and adaptability, becoming suitable for dynamic, uncertain, and large-scale environments [3]. In the last few years, swarm robotics has been extended to agriculture, environmental monitoring, industrial automation, and, most importantly, search and rescue missions (SRM) [4]. Search and rescue operations generally occur in hazardous environments, like disaster scenes, where response must be fast, flexible, and efficient in exploration [5].

Though swarm robotics showed promising applicability in such scenarios, the existing approaches are dominantly based on heuristic methods or rule-based frameworks that can hardly handle the arising complexities in real-time decision-making, coordination, and adaptability [6]. The traditional approach also deals with the problems of being non-scalable, resource-consuming, and

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250205

communication-constrained environments being quite fragile towards communications [7]. Increasing demands on effectiveness and robustness have compelled the scientific world to incorporate artificial intelligence in swarm robotics [8]. Deep reinforcement learning, or deep Q-Learning, has emerged as an immense ability to endow agents with learning and autonomous policies that adapt to complicated and dynamic tasks [9]. The conventional approach for search and rescue in swarm robotics comes with key limitations regarding scalability, adaptability, and coordination. More generally, rule-based and heuristic methods are suboptimal for decision-making in dynamic or unpredictable environments, resulting in inefficient exploration and low victim detection rates [10].

The problem statement is developing adaptive, decentralized frameworks for efficient autonomous learning of optimal policies in cooperative search and rescue missions. It also aims to scale up the problem size, maximize coverage, decrease the time to complete missions, and exhibit resilience in this dynamic environment. The DQLSRO approach combines the Deep Q-Learning algorithm with swarm robotics optimization, which is applied to cooperative search and rescue missions. Every robot in this approach has its Deep Q-Network, which it uses to learn autonomously optimal actions based on environmental feedback. Robots also use decentralized communication to update policies and make coordinated decisions. Multi-agent reward functions are incorporated into mission objective optimization, such as maximizing victim detection, minimizing exploration time, and obstacle avoidance. The framework's effectiveness is demonstrated in simulation for experimental missions under challenging conditions. The work has key meaning for

- To introduce a novel Deep Q-Learning-based framework for adaptive, decentralized search and rescue operations.
- To improve mission efficiency by reducing completion time and enhancing victim detection rates.
- To enhance resilience against robot failures and communication disruptions in dynamic environments.
- To demonstrate the scalability and robustness of DQLSRO through comprehensive experimental evaluations.

The paper is organized as follows: Section 1 gives an introduction and reviews related work in swarm robotics and deep reinforcement learning. Section 2 elucidates the methodology proposed. Section 3 presents the experimental setup and results. Finally, Section 4 concludes with key insights and future research directions.

de Carvalho, José Pedro Ferreira Pinheiro [11] proposed a Deep Reinforcement Learning (DRL) framework to enable cooperative robotic navigation in dynamic environments. DRL techniques were applied to train robots to learn optimal navigation policies autonomously in simulated multi-agent setups. Results showed significant improvements in navigation efficiency and adaptability. However, the research remains limited to simulated environments and lacks real-world validation. Challenges related to hardware constraints and environmental complexities must be addressed for practical application.

Chitikena et al. [12] introduced an ethical and design framework for search and rescue robotics during the response phase. The framework integrated ethical principles with design considerations, emphasizing safety, adaptability, and effective human-robot interaction. It provided valuable insights into the development of SAR robots that meet safety and ethical standards. However, the study is mainly conceptual, with minimal experimental validation to demonstrate its practical applicability in real-world scenarios.

Phadke and Medrano [13] presented an agent-focused framework to improve operational resiliency for UAV swarms in search and rescue missions. This framework uses decentralized, agent-based modeling and fault-tolerance strategies to optimize UAV coordination and performance. Results showed improved fault tolerance and efficiency in simulated SAR missions. However, this study is bounded to UAV systems and does not address broader applicability, integrating ground or hybrid robotic systems.

Solmaz et al. [14] developed a robust robotic framework for search and rescue in harsh environments, addressing challenges in scalability and autonomy. Robots with advanced sensors, mobility, and decision-making capabilities were tested in controlled environments, demonstrating improved robustness and adaptability. However, scalability and limited autonomy remain unresolved, with difficulties persisting in large-scale or resource-constrained scenarios.

Han et al. [15] proposed a collaborative task allocation and optimization solution for UAVs in SAR missions. A hybrid optimization approach combining genetic algorithms and auction-based methods efficiently distributed tasks. Results showed reduced mission completion times and optimized UAV resource utilization. However, the focus is exclusively on UAV systems, with no consideration of heterogeneous robotic teams that could broaden the method's applicability.

Inspired by biological systems, Sivaraman et al. [16] proposed a pack-hunting strategy for heterogeneous robots in rescue operations. Bioinspired algorithms simulated predator-prey dynamics to enhance coordinated task allocation among diverse robots. Results indicated improved task efficiency and adaptability in dynamic rescue scenarios. However, as the system scale increases, complexity becomes a significant challenge. Additionally, practical testing is limited, leaving questions about real-world robustness unanswered.

Queralta et al. [17] introduced a collaborative multi-robot framework for search and rescue, emphasizing planning, coordination, and active perception. The framework utilized real-time planning and robust perception techniques to improve multi-robot coordination. Results revealed significant improvements in search efficiency and victim detection rates in simulated scenarios. However, the framework lacks extensive testing in dynamic real-world SAR environments, limiting its immediate applicability to practical challenges.

Sanjay Sarma et al. [18] explored the impact of heterogeneity on collective behaviours in multi-robot systems during search and rescue missions. Simulation-based analysis highlighted how heterogeneity improves system robustness, adaptability, and task efficiency. While the results emphasized the benefits of diversity in robot systems, the study focused on simulations and lacked real-world validation, leaving its practical applicability in heterogeneous multi-robot scenarios unverified.

2. Proposed Methodology

The problem is formulated within a dynamic and partially observable grid environment, representing a disaster-stricken area. The grid is characterized by $G = (X, Y)$, where $X \; and \; Y$ denote the spatial dimensions of the area. Robots are deployed with sensors and actuators, enabling them to detect victims, avoid obstacles, and communicate with other robots. Each robot R_i is defined as $R_i = (S_i, A_i, C_i)$ where S_i represents the sensor capabilities (e.g., range r_s, victim detection probability P_d), A_i denotes actuator functions (e.g., mobility and obstacle avoidance), and C_i defines communication parameters (e.g., range r_c, latency L_c). The mission involves optimizing three key objectives: as Maximizing Coverage (C_v), Minimizing Rescue Time (T_r) and Ensuring Fault-Tolerant Communication (F_c). The mathematical expressions for these are shown in equation 1.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250205

$$C_v = \frac{|G_v|}{|G|}$$

$$T_r = \frac{1}{|V|} \sum_{v \in V} (t_{rescue}(v) - t_{detect}(v)) \quad (1)$$

$$P_c = \frac{M_{success}}{M_{total}}$$

where C_v Coverage is defined as the fraction of the total grid G visited by at least one robot within a given time T. G_v is the set of visited cells. T_r refers to the rescue time for each victim v∈V. V is the set of victims, defined as the time elapsed after detection until rescue. The objective of T_r is to minimize the mean rescue time. P_c communication reliability is measured by successfully delivering messages between robots within a time threshold. The fault-tolerance objective is to maximize the communication success rate (P_c). $M_{success}$ is the number of successfully delivered messages and M_{total} is the total number of messages. Observability and environmental dynamics call for real-time decision-making and robust communication protocols. Achieving optimality on these objectives involves trade-offs since high coverage might increase rescue time or strain communication reliability. Figure 1 shows the process of the proposed DQLSRO method.

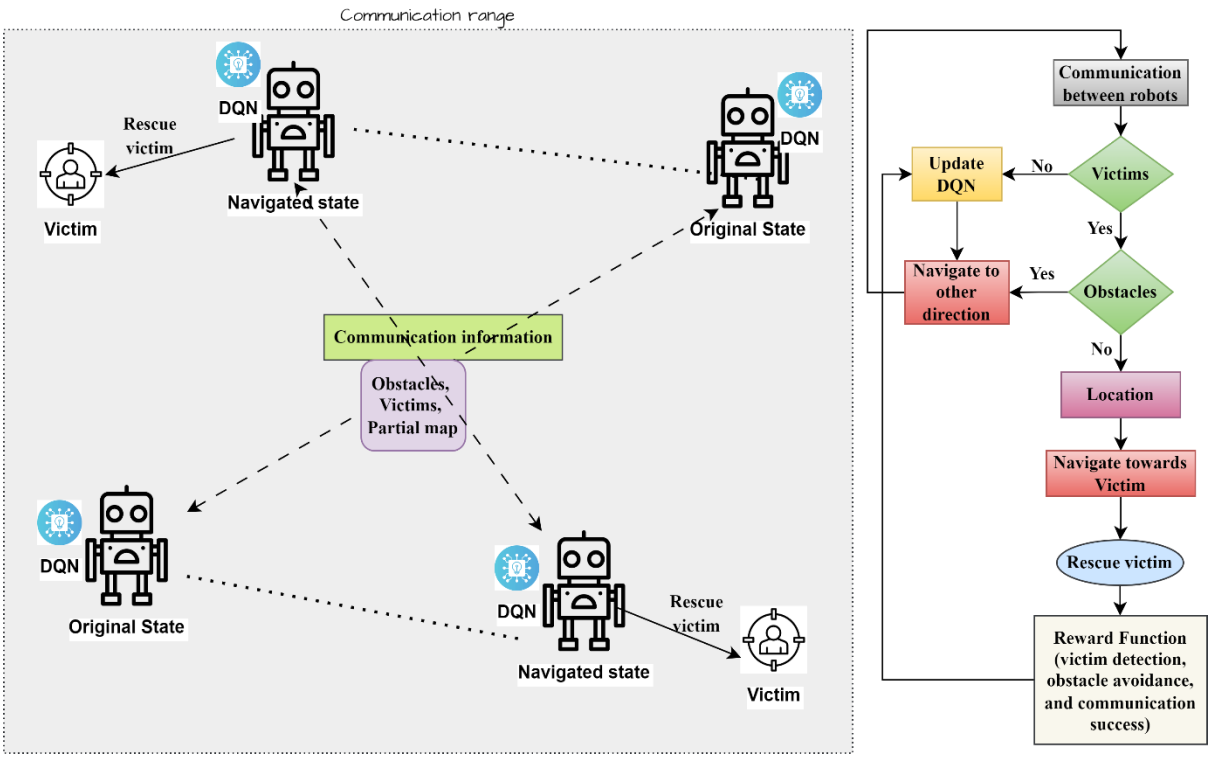

Figure.1. Process of the Proposed DQLSRO Method

a) Robot Representation and State Definition

In a multi-agent system, each robot is an independent agent that can decide based on what it perceives and interacts with. A robot's state at any time t is represented as an extensive vector S_t embodies critical information needed in decision-making and coordination. It is defined in equation 2. Figure 2 shows the robot representation and state definition.

$$S_t = \{P_t, O_t, V_t, C_t, E_t\} \quad (2)$$

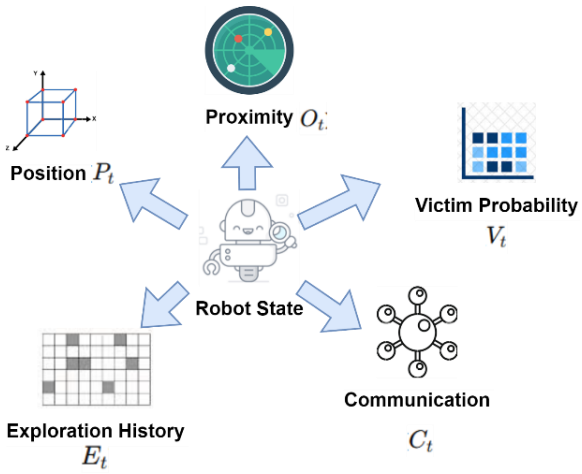

Figure.2. Robot representation and state definition

The state of the robot is defined by its position for navigation and coordination (x, y, z), proximity to obstacles (O_t) measured using sensors for safe navigation and the probability of finding a victim nearby ($V_t = P(victim|sensors)$). It also monitors the communication state (C_t) with signal strength and connection status parameters and keeps track of an exploration history (E_t) to avoid revisiting the same area. The state will help make efficient decisions, collaborate, and adapt to dynamic environments.

b) Action Space

In a multi-agent system, each robot is acting in some discrete action space denoted by A_t, comprising all possible actions the robot might take at some time t. Such actions depend on the time and the robot's state S_t along with environmental conditions. The robot's possible actions at time t can be defined as $A_t = \{a_1, a_2, \dots, a_n\}$ each action a_i corresponds to a particular task the robot can execute. If the current position $P_t = (x, y)$, the new position P_{t+1} after moving in a direction d. The action space A_t is updated dynamically based on the robot's environment and state. Figure 3 shows the robots' action spaces.

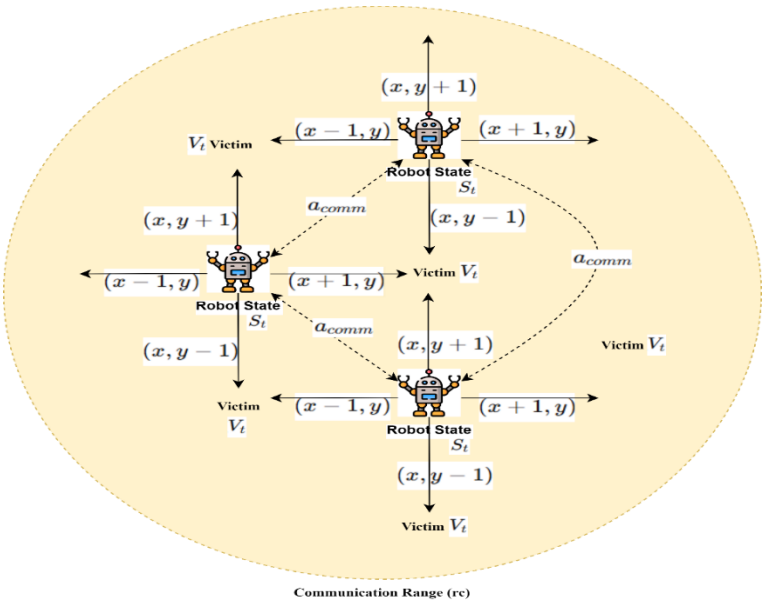

Figure.3. Robot's Action Space

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250205

The action set contains **moving** (equation 3). The robot moves to an adjacent cell in the grid environment. **Staying in the current position** (equation 4). The robot does not move and remains at P_t. **Initiating victim rescue** (a_{rescue}). If the robot detects a victim ($V_t > \theta$, where θ is a detection threshold), it initiates a rescue protocol. This may include signaling other robots for assistance. **The robot communicates with neighbouring robots** (a_{comm}) - The robot shares information, such as victim locations and areas explored, with other robots in its communication range r_c.

$$P_{t+1} = \begin{cases} (x, y+1) & if\ d = North \\ (x, y-1) & if\ d = South \\ (x+1, y) & if\ d = East \\ (x-1, y) & if\ d = South \end{cases} \quad (3)$$

$$P_{t+1} = P_t \quad (4)$$

Table 1 Reward Function Design

Reward Component	**Description**	**Mathematical Expression**	**Purpose**
Positive Rewards			
Locating a Victim	The reward for successfully detecting a victim based on sensor data.	$R_{victim} = \alpha \cdot victim\ sensor$	Locates a victim
Avoiding Obstacles	The reward for avoiding obstacles during movement (safe navigation).	$R_{obstacle-free} = \beta \cdot (1 - \frac{d_{min}}{d_{safe}})$	Promotes safe navigation by avoiding collisions.
Sharing Information	Reward for successful communication with neighboring robots (data exchange).	$R_{communication} = \gamma$ Information Shared	Fosters collaboration and information sharing between robots.
Negative Rewards			
Collisions	Penalty for colliding with obstacles or other robots.	$R_{collision} = -\delta$	Discourages collisions and unsafe actions.
Wasted Exploration	Penalty for revisiting previously explored areas (redundant exploration).	$R_{redundant} = -\epsilon$ Redundant Cells Visited	Ensures efficient area coverage by penalizing redundant exploration.
Cumulative Reward		$R_t = R_{victim} + R_{obstacle-free} + R_{communication} - R_{collision} - R_{redundant}$	Summation of all components to define the robot's overall performance at any time t.

Where $\alpha, \beta, \gamma, \delta, \epsilon$ are the weight factors to adjust the importance of each reward or penalty, d_{min} is the minimum distance to obstacles detected. d_{safe} refers to the predefined safe distance to maintain from obstacles. $Information\ Shared$ is the amount of data communicated with neighbouring robots. $Redundant\ Cells\ Visited$ is the number of already-visited grid cells revisited unnecessarily.

c) Deep Q-Learning Framework for Robot Action Selection

Each robot uses DQN to make decisions and improve its behaviour over time autonomously. The DQN model is a neural network approximating the Q-function, which estimates the expected future reward for taking a given action in a specific state.

i. *Input Layer - Encodes the Robot's State (S_t):* The input layer of the Deep Q-Network (DQN) receives the robot's state (S_t) at time (t), which encapsulates all the relevant information about the robot's environment and current condition. This state comprises the position of the robot in the grid, its distance from obstacles measured using sensors such as LiDAR, the estimated probability of a victim being close, based on thermal or sound detection, the communication state, whether the robot can communicate with its neighbours and its exploration history that shows which areas have already been covered. For processing in the DQN, this state is typically encoded as a vector or, in some cases, image-like data (mainly if visual sensors are used), which is then fed into the neural network for further processing and decision-making.

ii. *Hidden Layers - Processing the State Information:* The hidden layers of the DQN consist of convolutional layers (in the case of using image-like input) and fully connected layers to process the robot's state and extract relevant features. In this case, the convolutional layers are used for automatically detecting spatial patterns directly from visual or spatial data, such as thermal images or LiDAR scans, so that it can understand the environment's layout. After that, the features are passed on to the fully connected layers, which assemble all this information, learning complex relationships between the state variables and the expected rewards. This will enable the network to recognize the most important patterns and features for any action selection in a given state, contributing to better decision-making.

iii. *Output Layer: Produces $Q(S_t, a)$:* The output layer of the DQN is where the Q-values are generated, representing the expected future reward for each possible action (a) in the given state (S_t). Each Q-value represents a different action the robot could take, such as moving in a specific direction or rescuing a victim. The Q-value $Q(S_t, a)$ represents the expected total future reward the robot will accumulate if it acts (a) in the state (S_t) and thence onward follows the optimal policy. The output layer emits one Q-value for each action, which is used to choose the best action via an epsilon-greedy strategy. This strategy is that the robot explores random actions with some small probability of selecting a (ϵ), usually an action whose Q-value is maximised. The training goal is the optimization of those Q-values to make the robot consistently decide what actions to select to maximize its long-term rewards.

The DQN learns to update its Q-function over time using the Q-learning update rule as in equation 5.

$$Q(S_t, a) \leftarrow Q(S_t, a) + \alpha(R_t + \gamma \boldsymbol{max}_{a'} Q(S_{t+1}, a') - Q(S_t, a)) \quad (5)$$

where α is the learning rate (how quickly the network adapts). R_t is the immediate reward after acting a in state S_t. γ is the discount factor (how much future rewards are valued). $\boldsymbol{max}_{a'} Q(S_{t+1}, a')$ is the estimated future reward (for the next state S_{t+1}). The robot uses this update rule to adjust the Q-values and improve its policy over time.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250205

d) Communication and Coordination

Local communication allows robots to communicate with each other to share vital information. Robots share partial maps of the environment that have been explored, locations of victims, and information about observed obstacles. These shared data assist each robot in updating its DQN in a decentralized fashion so that the robots can adapt their behaviours and make informative decisions. By disseminating this information, the robots can coordinate their exploration and rescue efforts efficiently and effectively to optimize the global mission without any central control.

3. Result and Discussion

a) Dataset Explanation

The Rescue Object Detection dataset [19] is designed to train and evaluate object detection models in emergency and rescue scenarios. It contains annotated images of objects such as fire extinguishers, first aid kits, helmets, and other critical rescue-related equipment. Each image is labelled with bounding boxes and class information, making it suitable for machine-learning tasks like object detection and classification. This dataset is ideal for applications in safety compliance, robotic rescue missions, and augmented reality in disaster management.

b) Performance Metrics

The proposed DQLSRO is compared with other traditional approaches: Agent-Based Modeling with Fault Tolerance Strategies (ABMFTS) [13], Hybrid Optimization Approach (HOA) [15], and Collaborative Multi-Robot Framework (CMRF) [17] in terms of key performance metrics, including the Task Completion Time, Victim Detection Rate, and Resilience to Failures (Fault Tolerance), are presented to confirm the effectiveness of DQLSRO in most of these areas.

Task Completion Time: The total time the robot swarm takes to accomplish the SRM, that is, the time taken to find all victims, avoid obstacles, and navigate optimally in the environment. The proposed DQLSRO framework minimizes TC by considering deep Q-learning for the adaptive optimization of robot policies. This can be given by equation 6.

$$TCT = \max_{r \in R}\left(\sum_{t=1}^{T} \Delta t_r\right) \quad (6)$$

where R refers to the set of all robots in the swarm. Δt_r is the time step in which robot r completes a task (e.g., finding a victim, avoiding an obstacle). T is the total number of time steps until all robots complete all tasks. max Ensure that the slowest robot dictates the total time it takes to complete its part of the mission. Each robot learns optimal policies using a Deep Q-Network (DQN) for real-time decision-making within the DQLSRO framework to reduce time spent (Δt_r) on each task to avoid redundant exploration and ensure efficient coordination among robots. It further consists of multi-agent reward functions, minimizing the overall mission completion time (T) by coverage efficiency, allowing faster victim detection through collaboration.

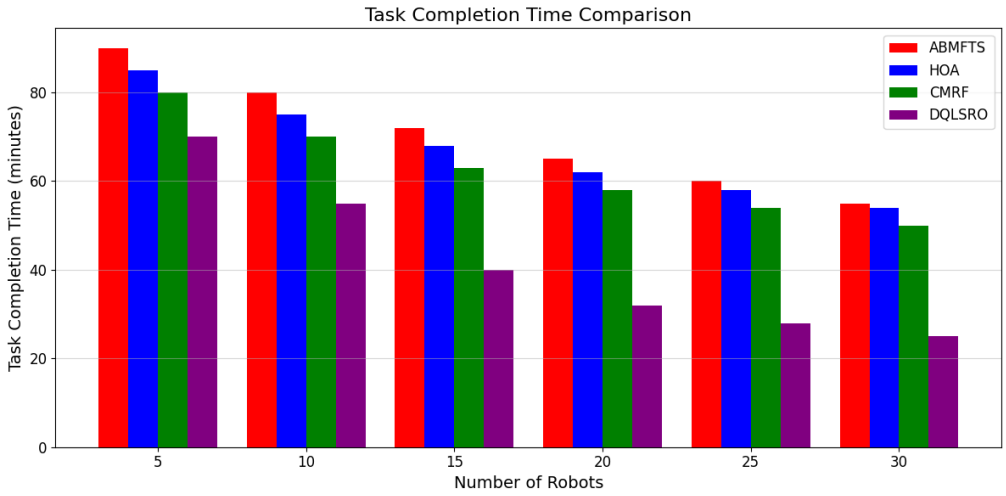

Figure.4. Task Completion Time Analysis

Figure 4 compares the task completion time among different methods, including ABMFTS, HOA, CMRF, and DQLSRO, for various robots. DQLSRO has the shortest bars in all cases, meaning it effectively finishes the mission ahead of traditional methods. By increasing the number of robots, DQLSRO shows high scalability by reducing completion time by 30% compared to others. Conventional methods, such as ABMFTS and HOA, have slower improvements with more robots, emphasizing their inefficiencies in coordination and adaptability to dynamic environments.

Victim Detection Rate (VDR): The VDR measures the proportion of victims successfully detected by the swarm of robots during a search and rescue mission. It is a key metric for evaluating the framework's effectiveness in efficiently locating victims. The VDR is expressed in equation 7.

$$VDR = \frac{N_d}{N_t} \times 100 \tag{7}$$

where N_d is the number of victims successfully detected. N_t is the total number of victims in the environment. The result is expressed as a percentage (%).

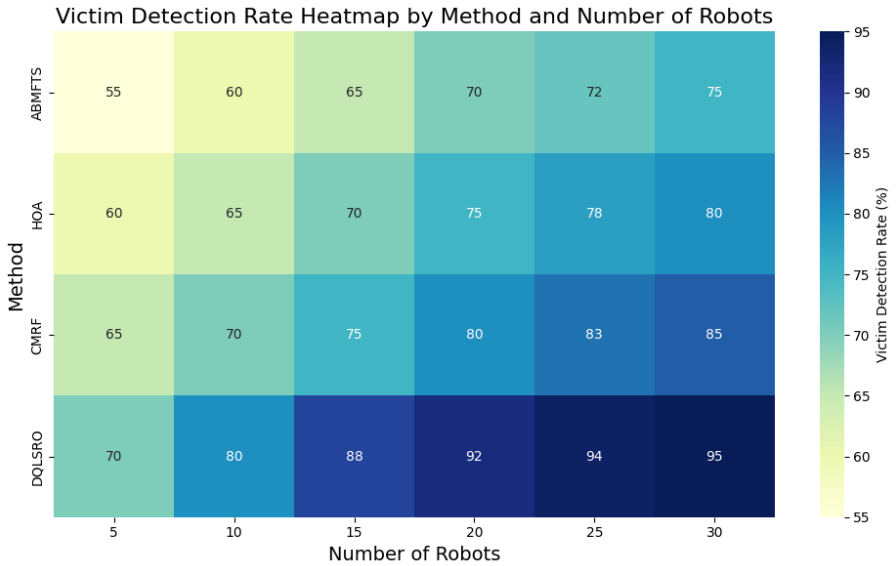

Figure.5. Victim Detection Rate Analysis

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250205

Figure 5 visualizes the Victim Detection Rate for four methods, ABMFTS, HOA, CMRF, and DQLSRO, under various robots (5, 10, 15, 20, 25, 30). The colour intensity represents the VDR, where darker shades show higher detection rates. DQLSRO constantly shows higher values of VDR for all robot counts, proving to be more efficient in victim detection. In contrast, the traditional approaches of ABMFTS and HOA show lower and spread-out detection rates; hence, these approaches are limited in their scalability and performance in more enormous robot swarms. This further improved the effectiveness of DQLSRO on large-scale search and rescue operations.

Resilience to Failures (Fault Tolerance): In swarm robotics, resilience to failures means that the system should not stumble in its tasks even if a subset of the robots breaks or communications among the robots are lost. The fault tolerance metric measures how well the swarm's performance is maintained with any breakdowns of its robots. Fault Tolerance (FT) can be defined as the ratio of task completion times, showing how well the system performs despite failures. Equation 8 shows how well the system performs despite failures.

$$FT = \frac{T_{complete} - T'_{complete}}{T_{complete}} \times 100 \quad (8)$$

where $T_{complete}$ be the task completion time when all robots are operational. $T'_{complete}$ be the task completion time when some robots fail. N be the number of robots in the swarm. This equation provides a percentage value indicating the decrease in task performance due to robot failures. A higher FT value indicates better fault tolerance.

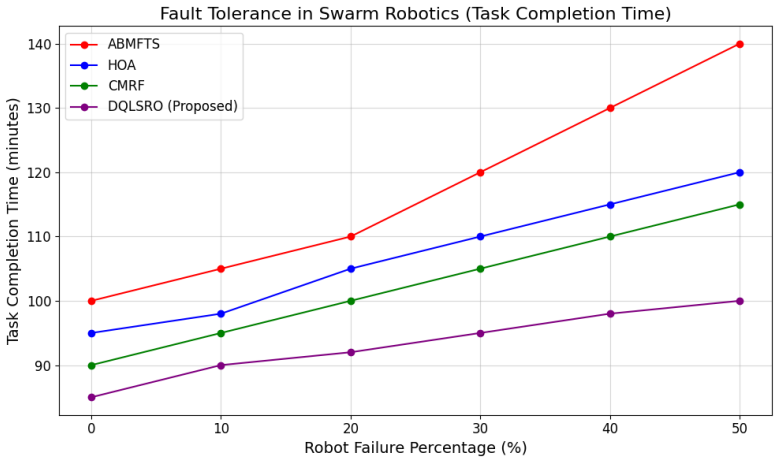

Figure.6. Fault Tolerance Analysis

Figure 6 shows the fault tolerance of four swarm robotics methods: ABMFTS, HOA, CMRF, and DQLSRO. It shows the task completion time when the percentage of robot failure increases. The X-axis represents the percentage of failure from 0% to 50%, and the Y-axis shows the corresponding task completion time in minutes. DQLSRO has the best robustness with the smallest increment of task completion time as robot failures increase, which indicates it has better fault tolerance than traditional approaches like ABMFTS and HOA.

4. Conclusion

The DQLSRO framework achieves the goal of integrating Deep Q-Learning in swarm robotics to address complex problems of search and rescue applications. The work gives robots independence in learning through decentralized decisions made by a swarm of robots. It attains better performance on mission execution by reducing failure times and robot victim detection rates. Most importantly,

coordination between robots creates a robust execution that is effective against failures caused by dynamic environments. The multi-agent reward structure incentivises collaboration and optimizes exploration, victim location, and obstacle avoidance. Experimental results prove the superiority of DQLSRO over traditional rule-based and heuristic methods by showing its scalability and resilience in different scenarios. DQLSRO has the possibility of revolutionizing swarm robotics to provide a transformative solution for critical applications such as disaster management and search and rescue. Future work will concentrate on designing lightweight learning algorithms that could improve scalability and incorporate advanced fault-tolerant communication mechanisms to increase the robustness of decentralized operations. Real-world validation through deployment on physical robots and the integration of advanced sensor systems can further advance the practicality of DQLSRO in emergency response scenarios.

References

[1]. Debie, Essam, Kathryn Kasmarik, and Matt Garratt. "Swarm robotics: A Survey from a Multi-tasking Perspective." ACM Computing Surveys 56.2 (2023): 1-38.

[2]. Dias, Pollyanna G. Faria, et al. "Swarm robotics: A perspective on the latest reviewed concepts and applications." Sensors 21.6 (2021): 2062.

[3]. Abouelyazid, Mahmoud. "Adversarial Deep Reinforcement Learning to Mitigate Sensor and Communication Attacks for Secure Swarm Robotics." Journal of Intelligent Connectivity and Emerging Technologies 8.3 (2023): 94-112.

[4]. Hoang, Maria-Theresa Oanh, et al. "Drone swarms to support search and rescue operations: Opportunities and challenges." Cultural Robotics: Social Robots and Their Emergent Cultural Ecologies (2023): 163-176.

[5]. Kumar, Girish, et al. "Obstacle avoidance for a swarm of unmanned aerial vehicles operating on particle swarm optimization: A swarm intelligence approach for search and rescue missions." Journal of the Brazilian Society of Mechanical Sciences and Engineering 44.2 (2022): 56.

[6]. Calderón-Arce, Cindy, Juan Carlos Brenes-Torres, and Rebeca Solis-Ortega. "Swarm robotics: Simulators, platforms and applications review." Computation 10.6 (2022): 80.

[7]. Drew, Daniel S. "Multi-agent systems for search and rescue applications." Current Robotics Reports 2 (2021): 189-200.

[8]. Orr, James, and Ayan Dutta. "Multi-agent deep reinforcement learning for multi-robot applications: A survey." Sensors 23.7 (2023): 3625.

[9]. Yang, Jian, and Xuejun Huang. "Intelligent Route Planning Method for UAV Based on Swarm Intelligence and Deep Learning Technology." Computing and Informatics 43.4 (2024): 874-899.

[10]. Koradiya, Gauravkumar A. Reinforcement Learning Based Planning and Control for Robotic Source Seeking Inspired by Fruit Flies. MS thesis. San Jose State University, 2024.

[11]. de Carvalho, José Pedro Ferreira Pinheiro. "Deep reinforcement learning methods for cooperative robotic navigation." (2023).

[12]. Chitikena, Hareesh, Filippo Sanfilippo, and Shugen Ma. "Robotics in search and rescue (SAR) operations: An ethical and Design Perspective Framework for Response Phase." Applied Sciences 13.3 (2023): 1800.

[13]. Phadke, Abhishek, and F. Antonio Medrano. "Increasing Operational Resiliency of UAV Swarms: An Agent-Focused Search and Rescue Framework." Aerospace Research Communications 1 (2024): 12420.

[14]. Solmaz, Selim, et al. "Robust robotic search and rescue in harsh environments: An example and open challenges." 2024 IEEE International Symposium on Robotic and Sensors Environments (ROSE). IEEE, 2024.

[15]. Han, Dan, et al. "Collaborative Task Allocation and Optimization Solution for Unmanned Aerial Vehicles in Search and Rescue." Drones 8.4 (2024): 138.

[16]. Sivaraman, Dileep, et al. "A pack hunting strategy for heterogeneous robots in rescue operations." Bioinspiration & Biomimetics 20.1 (2025): 016029.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250205

[17]. Queralta, Jorge Pena, et al. "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision." Ieee Access 8 (2020): 191617-191643.

[18]. OV, Sanjay Sarma, Ramviyas Parasuraman, and Ramana Pidaparti. "Impact of heterogeneity in multi-robot systems on collective behaviors studied using a search and rescue problem." 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2020.

[19]. https://www.kaggle.com/datasets/julienmeine/rescue-object-detection