Using Deep Reinforcement Learning to Coordinate Autonomous Maritime Search and Rescue Drones

Pedro H. B. A. Andrade[1], Ricardo R. Rodrigues[1], Renato L. Falcão[1], Joras C. C. Oliveira[1], José F. B. Brancalion[3], and Fabrício J. Barth[2]

[1]Insper, São Paulo, Brazil
Email: {pedroa3, ricardorr7, renatolf1, jorascco}@al.insper.edu.br

[2]Insper, São Paulo, Brazil
Email: fabriciojb@insper.edu.br

[3]Embraer, São José dos Campos, Brazil
Email: jose.brancalion@embraer.com.br

***Abstract*—This paper explores the use of deep reinforcement learning (DRL) to improve the efficiency and effectiveness of maritime search and rescue (SAR) operations using autonomous drones. This paper compares the performance of two RL algorithms, Proximal Policy Optimization (PPO) and Deep Q-Network (DQN), against a greedy search algorithm in a simulated SAR environment. The results show that PPO and DQN outperform the greedy search approach, particularly in more complex environments with higher dispersion increments (faster expansion of the search area). PPO demonstrates significantly faster learning and achieves higher success rates compared to DQN.**

***Index Terms*—Deep Reinforcement Learning, Simulation, Search and Rescue**

I. Introduction

The issue of missing persons-in-water (PIW) is as old as humankind itself, and given the chaotic nature of the ocean, search and rescue (SAR) operations have never been optimal, with limitations on the human ability ranging from the creation of proper search paths to visibility and recognition of PIW. According to several authors, using drones allows for continuous search over extended periods of time and distances. While the capabilities of artificial intelligence (AI) with reinforcement learning for problem-solving are still in their infancy, it is theorized that introducing AI for SAR applications may significantly improve the efficacy of search operations and reduce the time needed to find and save PIW [1, 2]. Reinforcement Learning (RL) is believed to enable the development of new, more efficient search patterns tailored to specific applications. This is based on the hypothesis that reward maximization is sufficient to foster generalization abilities, thereby creating powerful agents [3]. Such advancements could potentially lead to the saving of more lives.

Ai [2] explores reinforcement learning (RL) in maritime SAR, using boats for the search. The study focuses on decision-making, search area determination, and maritime vehicle deployment, using metrics like the probability of detection (POD), probability of containment (POC), and probability of success (POS). A drift prediction component forecasts the trajectory of distressed objects, constructing a search grid with cells assigned a POC. Using Q-learning, the algorithm selects actions to maximize POS based on the current state. Experimental results show effective coverage with minimized redundancy.

Wu [1] builds on this with deep reinforcement learning (DRL), using the DQN algorithm, where deep neural networks replace the Q-table, allowing the algorithm to approximate values and choose the best action based on the environment. Experiments show this approach matches or exceeds the Q-learning results, completing SAR missions more efficiently. However, both methods assume a constant search area, not accounting for dynamic factors like wind or waves during the search.

The LSAR [4], Q-learning [2], UAS [5], and DQN [1] approaches all use probability regions to locate targets. LSAR consistently centers the highest probability region, while other methods use maritime data and statistical methods. LSAR and UAS employ multiple agents, with LSAR drones communicating upon detection and UAS agents operating independently. Each approach aims to optimize SAR missions and enhance success rates using distinct strategies. In this paper, the success rate refers to the likelihood of successfully locating and rescuing subjects within a specific timeframe, minimizing the search duration.

Abreu [6] implemented a Reinforce algorithm that considers a dynamic map of probabilities, representing the chances of a person being found and the position of other agents. This work compares a reinforcement learning algorithm with a parallel sweep [7] algorithm with a pre-defined behavior. This work provided a proof of concept, supporting the notion that reinforcement learning strategies outperform predefined paths in efficiency and effectiveness. However, this work did not compare reinforcement learning algorithms with other algorithms used in search and rescue missions, such as algorithms that explore heuristics to find the best path to the target.

This paper compares deep reinforcement learning (DRL) algorithms with greedy search algorithms. We aim to improve

real-world SAR operations using reinforcement learning (RL). Previous studies [1, 2, 6] show promising results for RL in SAR, and we will build on this by demonstrating how RL can help achieve faster search and high recovery rates in tested scenarios. We also will use a simulation tool [8] that is more complex and realistic than the simulation tools used by [2] and [1].

This paper is structured as follows: in the section II, we provide an overview of rescue and search strategies; in the section III, we describe the environment used to train and validate the algorithms; in the section IV, we present our experiments and results; finally, we discuss our findings.

II. Rescue and Search Strategies

Implementing efficient algorithms in autonomous maritime search and rescue operations is paramount to ensuring timely and effective responses to emergencies. Two distinct approaches could be explored in this domain: greedy search algorithms and reinforcement learning methodologies [1, 2, 5, 7]. The greedy search algorithm operates on the principle of immediate reward maximization, making decisions based solely on the information available at the current moment. While this algorithm may yield satisfactory results in specific scenarios, its deterministic nature often leads to suboptimal solutions, particularly in complex and dynamic environments such as maritime search and rescue missions.

In contrast, reinforcement learning offers a promising alternative by allowing autonomous agents to learn and adapt their behaviors through interaction with the environment. By leveraging a system of rewards and punishments, reinforcement learning enables agents to explore various strategies and gradually refine their decision-making processes over time. In maritime search and rescue, reinforcement learning empowers agents to navigate unpredictable conditions, dynamically allocate resources, and optimize search patterns to maximize the chances of locating and rescuing survivors.

A. Greedy search approach

Informed algorithms have some form of knowledge about the task at hand, such as locations to search, time, or other factors assumed necessary for solving the task. In this case, the informed algorithm selected was a greedy search.

The greedy search was constructed using a simple yet effective heuristic. For N drones, each drone receives a point from the list of highest probabilities, moves toward the end, and searches for it. This algorithm was selected based on its ease of implementation, non-stochastic behavior, and its effectiveness in the current environment used for the simulations.

B. Deep Reinforcement Learning implementation

To effectively compare and evaluate multiple search algorithms, we utilized the RLlib [9] library to implement the reinforcement learning algorithms Proximal Policy Optimization (PPO) [10] and Deep Q-Network (DQN) [11].

DQN and PPO were chosen to represent the main algorithms from the value-based and policy-based families, respectively. Value-based algorithms, like DQN, approximate the value function for the environment through a trained Q-table or Q-network. In contrast, policy-based algorithms, like PPO, approximate an ideal policy by mapping states to actions, often using neural networks. These algorithms do not directly map all possible actions and states to rewards but instead optimize policy parameters to maximize cumulative rewards [12].

The two algorithms differ in their learning and exploration strategies, which are crucial for the reinforcement learning process. PPO selects actions by sampling from the probability distribution output by its policy network, facilitating higher exploration when the agent is uncertain and greater exploitation as the agent learns. In contrast, DQN uses an ϵ-greedy algorithm, balancing exploration by sampling random actions and exploitation by choosing actions with the highest Q-value.

III. Environment

For this project, the Drone Swarm Search Environment (DSSE) framework, developed by the authors and available as a Python package [8], was used to study the viability of using reinforcement learning in searching for shipwrecked people.

This environment is designed to train RL agents to locate PIW and simulate its movement in real time. It concludes when all PIW are found or a certain time step limit is reached. This environment incorporates relevant variables to the search scenario, such as the POD, number of PIW, and vector values that guide the movement of both the PIW and the probability matrix.

The environment is a 2D grid with a probability matrix representing the probability of containing a PIW. The probability matrix is based on a Gaussian distribution and a dispersion increment, which controls the Gaussian expansion rate. The distribution is calculated using the bi-dimensional Gaussian function presented in the equation 1.

$$f(x,y) = A \cdot \exp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right) \quad (1)$$

where A is the amplitude of the Gaussian function, x_0 e y_0 are the coordinates of the supposed position of the PWI. σ_x e σ_y define how the function will be stretched along the matrix, and correspond to the dispersion increment, determining how quickly the Gaussian expands in both directions. Finally, x e y represent the horizontal and vertical positions within the grid.

The dispersion increment dictates the rate at which the Gaussian expands. For instance, 0.1 allows faster expansion, suitable for cases where the PIW moves rapidly, while 0.05 provides a slower, more precise spread. These values were chosen to balance coverage and accuracy in different scenarios.

A vector representing ocean currents and the wind controls the direction of movement. The PIW's movement is influenced by the probabilities in the matrix, which select a pseudo-random direction weighted by the probabilities of adjacent cells and an independent movement vector.

The environment states are represented as a tuple of two boxes: one for the drone's position (x, y) and another for

the probability matrix. This allows the agents to use the probabilities and their own positions to decide the best action, facilitating the orchestration of complex search patterns to find the PIW. The environment's action space is discrete, consisting of 9 actions: moving in the cardinal and intercardinal directions and searching the current cell. With a Probability of Detection (POD) of 1.0, the environment is deterministic, but with lower values, it becomes stochastic. The problem is a sparse reward problem, as the goal is reached only when the PIW is found. The reward scheme represents this, as seen in the equation 2.

$$R(Ts) = \begin{cases} 2 - \frac{Ts}{Ts_{limit}}, & \text{if found} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where Ts is the time step that the PIW was found and Ts_{limit} is the configurable time step limit.

IV. Experiments and Results

The training data was collected through simulated SAR missions in the DSSE environment, employing DQN and PPO algorithms and the baseline greedy search for comparison. The training process for the algorithms involves various configurations of the environment, each using a 40x40 navigation grid in a multi-agent scenario with four agents. A key variable in these configurations is the dispersion increment, set to 0.1 and 0.05. Dispersion is the rate at which the probability matrix spreads within the environment. A higher dispersion rate means the search area expands quickly.

The validation data was collected by running the SAR mission simulation 5,000 times. This involved the greedy algorithm and the trained DQN and PPO algorithms across two neural network modes in the same environment.

The training routine using RLlib [9] collected rewards received, the number of actions taken, and whether the agents found the PIW in each trial. The training data analysis involved calculating the moving average of the rewards and actions, with the window size depending on the training batch. Data was collected in parallel using RLlib and stored by the Learner, the class within RLlib responsible for defining neural network training parameters and data gathering for later analysis.

Tests using RL algorithms on a 40x40 grid with dispersion increments of 0.1 and 0.05 show that RL outperforms the greedy approach in more complex environments. PPO significantly outperformed the greedy approach, with a 75.44% success rate versus 35.84% for 0.1 dispersion increment and 83% versus 50.18% for 0.05. Remember, the success rate refers to the likelihood of successfully locating and rescuing subjects within a specific timeframe.

Furthermore, reinforcement learning algorithms can learn complex search patterns, coordinate to encircle the probability matrix in its front and rear and intercept the PIW, which is suitable for unpredictable SAR missions.

Figures in 1 and 2 display PPO learning curves for these configurations and compare PPO and DQN learning. According to these results, PPO learns a better policy much quicker than DQN, achieving rewards between 1.4 and 1.75, while DQN reaches only 0.4. PPO training took 3 hours, whereas DQN took 23 hours. PPO's advantage is attributed to its action probability distribution, allowing better exploration of states the agent has less knowledge about. Thus, PPO was chosen for further tests.

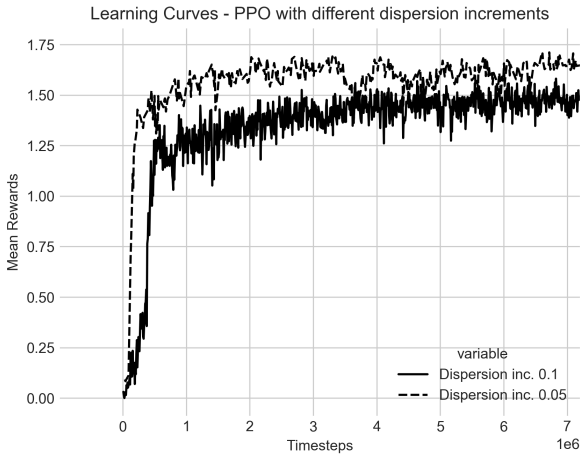

Fig. 1. Learning curves for PPO on 0.1 and 0.05 dispersion increment configuration

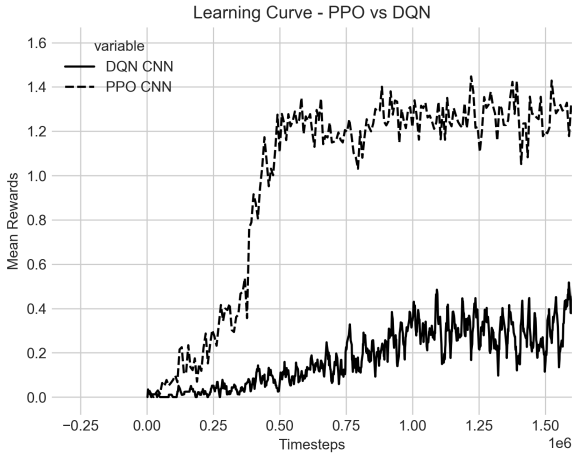

Fig. 2. Learning curve comparison of PPO and DQN on 0.1 dispersion increment configuration

V. Conclusions and Future work

This paper proposes using PPO and DQN algorithms to implement drones' behavior in search and rescue activities. We showed that RL techniques have superior adaptability and performance compared to a pre-determined, informed algorithm, especially when PIWs deviate from regions of highest probability. The results confirm that RL algorithms outperform the greedy approach in all metrics. PPO achieved a 75.44% success rate with a 0.1 dispersion increment and 83%

with a 0.05 increment, compared to the greedy algorithm's 35.84% and 50.18%, respectively. DQN failed to learn an effective strategy.

Furthermore, the agents trained using the PPO algorithm consistently demonstrated the ability to locate the target swiftly across various configurations. This aspect aligns with the critical need for prompt responses in scenarios involving shipwrecked individuals. Scaling up the number of drones resulted in faster and more accurate responses, justifying the utilization of multiple agents.

Further investigations are imperative to thoroughly evaluate the effectiveness of reinforcement learning in addressing Search and Rescue (SAR) challenges. It is hypothesized that as the training environment for the agents incorporates heightened realism, featuring additional factors such as wind currents, a more faithful simulation of Person in Water (PIW) movement, expanded search areas, increased dispersion numbers, and the integration of probability of detection based on diverse climate scenarios (e.g., fog, nocturnal conditions, storms, and strong waves), reinforcement learning agents are poised to develop varied and potentially more efficient search patterns.

Moving from simulation to real-world deployment introduces critical challenges that must be addressed to fully realize DRL's potential in SAR operations. Key considerations include drone hardware limitations, sensor accuracy, reliable communication in maritime conditions, and robust safety protocols. Additionally, adapting RL models to real-world variability, such as unpredictable weather and complex water dynamics, will be essential. Tackling these practical issues is vital to advancing DRL applications and ensuring that SAR missions benefit from enhanced reliability and efficiency in real-world scenarios.

References

[1] J. Wu, L. Cheng, S. Chu, and Y. Song, "An autonomous coverage path planning algorithm for maritime search and rescue of persons-in-water based on deep reinforcement learning," *Ocean Engineering*, vol. 291, p. 116403, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801823027877

[2] B. Ai, M. Jia, H. Xu, J. Xu, Z. Wen, B. Li, and D. Zhang, "Coverage path planning for maritime search and rescue using reinforcement learning," *Ocean Engineering*, vol. 241, p. 110098, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801821014220

[3] D. Silver, S. Singh, D. Precup, and R. S. Sutton, "Reward is enough," *Artificial Intelligence*, vol. 299, p. 103535, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370221000862

[4] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "Lsar: Multi-uav collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55 817–55 832, 2019.

[5] D. W. Schuldt and J. A. Kurucar, "Maritime search and rescue via multiple coordinated uas," Defense Technical Information Center, Tech. Rep., 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:177273

[6] L. D. M. Abreu, L. F. S. Carrete, M. Castanares, E. F. Damiani, J. F. Brancalion, and F. J. Barth, "Exploration and rescue of shipwreck survivors using reinforcement learning-empowered drone swarms," in *XXV Simpósio de Aplicações Operacionais em Áreas de Defesa*, 2023, pp. 64–69.

[7] K. CHOUTRI, M. LAGHA, and L. DALA, "A fully autonomous search and rescue system using quadrotor uav," *International Journal of Computing and Digital Systems*, vol. 10, pp. 2–12, 2021.

[8] R. L. Falcão, J. C. C. de Oliveira, P. H. B. A. Andrade, R. R. Rodrigues, F. J. Barth, and J. F. B. Brancalion, "DSSE: An environment for simulation of reinforcement learning-empowered drone swarm maritime search and rescue missions," *Journal of Open Source Software*, vol. 9, no. 99, p. 6746, 2024. [Online]. Available: https://doi.org/10.21105/joss.06746

[9] E. Liang, R. Liaw, P. Moritz, R. Nishihara, R. Fox, K. Goldberg, J. E. Gonzalez, M. I. Jordan, and I. Stoica, "Rllib: Abstractions for distributed reinforcement learning." International Conference on Machine Learning, 2018.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[12] S. V. Albrecht, F. Christianos, and L. Schäfer, *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press, 2024. [Online]. Available: https://www.marl-book.com