Neurocomputing 653 (2025) 131211

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Search and rescue operations in wildfires using unmanned aerial vehicles: A multi-agent deep reinforcement learning approach

Maxime Collignon, Adolfo Perrusquía*, Antonios Tsourdos, Weisi Guo

Faculty of Engineering and Applied Sciences, Cranfield University, Bedford MK43 0AL, UK

ARTICLE INFO

Communicated by J. Xu

Keywords:
Wildfire
Search and rescue (SAR)
Multi-agent deep Q-network (MADQN)
Heuristic-based model
Score

ABSTRACT

Wildfires pose major challenges to natural ecosystems and smart living due to their destructive nature. Unmanned Aerial Vehicles (UAVs) or drones have been used to support fire fighter in identifying vulnerable areas and detecting people who need assistance. Most of the current solutions use path planning approaches under simple and deterministic environments that fail to model the dynamic nature of fire. Furthermore, the localisation of victims is assumed to be known which is unrealistic in disaster-like scenarios. To alleviate this issue, this paper proposes a novel search and rescue (SAR) application using drones. A multi-agent deep Q-network is designed to train a fleet of UAVs to search for people and evacuate them in a wildfire scenario. A realistic forest environment is designed that considers variations in vegetation and fire propagation. This helps to challenge RL algorithms to be more adaptive to changes in the environment due to the evolution of fire. Extensive simulation experiments are conducted to show the advantages and effectiveness of the proposed approach.

1. Introduction

Autonomous systems have been used as a key tool in search and rescue (SAR) missions due to the complex nature and risks in disaster scenarios [1–5]. Wildfires [6,7] have become increasingly common in recent years due to extreme weather conditions caused by climate change [8,9]. The threats of fires include: social [10–12], economic [13,14], and ecological (both in wildlife [15,16] and flora [17,18]).

Unmanned aerial vehicles (UAVs) also known as drones have been used in SAR missions involving wildfires due to their low cost and high manoeuvrability in complex and risky environments. The most common approaches for drones in SAR missions involve the design of path planning and tracking algorithms [19]. Traditional tracking applications using drones have prior knowledge of the environment (or map) such that finding the best path to reach a target location can be solved using a path optimisation technique. However, the environment changes after a disaster event, such that the previous route plan is not effective anymore. This raises two main questions that are concern of this paper: 1) how drones can effectively explore a wildfire zone, and 2) how drones know where the locations of the desired targets (e.g., people) are in order to assist and evacuate them before the fire propagates further.

1.1. Related work

The use of fleets of robots or drones working in swarms for SAR missions is becoming more common [20]. The key idea of using swarms is to exploit their enhanced perception and planning capabilities to perform SAR missions during or after wildfires autonomously.

Monte Carlo tree search [21] was applied to a swarm of UAVs for SAR missions, and multi-agent deep reinforcement learning (MADRL) algorithms [20] have been used to train individual agents or swarms of UAVs capable of performing SAR missions. A RL-trained maritime drone for SAR missions following maritime disasters is proposed in [22], which is capable of planning its mission by optimizing its trajectory lengths and being more efficient than current solutions.

Several researches have been conducted regarding the target localisation, e.g., in [23] it was proposed to minimize the time required to locate victims during natural disasters. Here, the swarm was trained to discover targets strategically in an obstacle-laden geographical area using Adversarial MADRL. Another approach introduces a MADRL-based framework for conducting Multi-Agent SAR (MASAR) missions in unknown, unpredictable environments where targets exhibit dynamic and unpredictable behaviour [24]. The research in [25] aimed to achieve

* Corresponding author.
Email addresses: maxime.collignon.250@cranfield.ac.uk (M. Collignon), adolfo.perrusquia-guzman@cranfield.ac.uk (A. Perrusquía), a.tsourdos@cranfield.ac.uk (A. Tsourdos), weisi.guo@cranfield.ac.uk (W. Guo).

https://doi.org/10.1016/j.neucom.2025.131211
Received 27 February 2025; Received in revised form 30 June 2025; Accepted 5 August 2025
Available online 7 August 2025
0925-2312/© 2025 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

efficient and communication fault-tolerant maritime SAR operations using MADRL by training heterogeneous agents to successfully perform trajectory planning and resource scheduling.

A MADRL was proposed for exploration SAR missions [26]. Here, the swarm was trained through distributed parallel threads developing the MADAC (Multi-Agent Distributed Actor Critic) framework, using a Long Short-Term Memory (LSTM) neural network [27] to account for imperfect communication between agents and partial, incomplete observations of the environment [28]. The research in [29] improved the efficiency of emergency response after a disaster in urban areas by proposing ResQ, a MADRL-based algorithm that can organize the rapid and effective deployment of rescuers to identify victims in a dynamic environment.

1.2. Contributions and outline

In view of the above, this paper proposes a novel solution for the localisation and evacuation of people in wildfire scenarios using multi-agent deep reinforcement learning (MADRL). The approach designs a relative complex model of wildfire that considers variable vegetation density, and a probabilistic fire model. In addition, we consider incorporating a new feature based on the temperature of the environment to avoid using cameras that are prone to high-false positives due to lightning conditions [30]. A multi-agent deep Q-Network (MADQN) algorithm is designed based on the environment model design which aims to detect people and evacuate them, whilst considering the fire propagation and recently visited zones that are not required to be visited again. In addition, a simple heuristic algorithm is designed under the same task philosophy for comparison purposes. Extensive simulation experiments are conducted to show the advantages of the proposed formulation and future research vectors.

The main contributions of this paper are,

- We model the forest as a grid where each cell has a specific vegetation density, whereas current research focuses on forests with constant density. This approach makes wildfires much more complex for the trained agents to manage, as their propagation becomes less predictable, with the fire front not being circular or continuous.
- We propose a visualization and decision-making approach based on temperature measurements. Previous authors such as in [31] designed the agents to observe the state of a cell in the grid directly: whether it is on fire, burned, or burning. This approach is not realistic because the state of these cells would be difficult to observe in reality due to smoke or lighting conditions. We propose modelling the temperature evolution around a wildfire and equipping our UAVs with temperature sensors, a technology that exists in the literature [32].
- A novel SAR solution is designed using a MADQN which considers the stochastic spread of wildfire under variable density vegetation. Each element of the MADQN (states, actions, sparse reward design) is well-defined to enhance the detection and evacuation of people in wildfires.

The outline of this paper is as follows. Section 2 introduces the methodology followed in this work, which includes the environment modelling, the MADQN algorithm design, and a heuristic algorithm design. Section 3 provides the results and discussions of the experiments conducted in this work. Conclusions are presented in Section 4.

2. Methodology

The methodology of this paper consists of two main parts dedicated to training and testing the proposed RL-based approach in SAR missions using drones. Fig. 1 summarizes the high-level block diagrams of the approach. Both models use a multi-agent deep Q-network (MADQN) architecture applied in a novel forest environment under variable density vegetation and fire propagation. Here, the environment provides both local and global information to the agents covering both fire locations, positions of other drones, and the information of the closest drones. Both the environment and MADQN algorithm design are discussed in the next sections.

We assume that the UAVs used in this paper have the basic capability, once a house is discovered, to locate people in need of urgent evacuation by firefighters (and potentially provide first aid or instructions for their evacuation). Therefore these drones can assist people who have not yet been reached by the fire. Here, we focus solely on training agents to explore the environment as efficiently as possible in search of people in danger, who need evacuation, in houses not yet reached by the fire. These agents will need to take into account the fire's progression and learn to avoid exploring unnecessary areas, such as already burned regions, as this is not part of their mission. We assume that the agents are equipped with house recognition technology via a standard camera as found in existing literature [33,34]. In this paper, we do not consider image processing and delays in gathering the data. We also assume that the agents have all the necessary systems to carry out their evacuation mission and/or transmit the location to the fire fighters once a house is discovered. The performance of the agents in an episode will correspond to the proportion of houses in the environment that were located and visited by the agents before being reached by the fire. Agents can observe their environment and identify houses within a radius of 2 blocks around them.

2.1. Proposed environment model design

Consider a grid G of size $N \times N$ with $N \in \mathbb{N} \setminus \{0\}$ denoted as $G = \mathbb{N}_N^2$ that models a forest environment. Each cell of the grid contains all the positions occupied by the forest. We propose $N = 100$ to effectively cover the complete grid.

2.1.1. Variable vegetation density

Previous approaches model forest models [31] with constant vegetation density such that wildfires will grow radially[35], which is not realistic due to the heterogeneous nature of forests. To be consistent with real wildfire scenarios, we propose a novel methodology to model forests with a variable vegetation density that adds more complexity and realism.

In each cell (i, j), we denote its variable vegetation density as $d_{ij} := d(i, j)$ where $d_{ij} \in [0, 1]$ and it is obtained by following the steps of Algorithm 1.

Algorithm 1 Variable Vegetation Density distribution algorithm.

Require: Approximate width of high/low density zones denoted as σ_d, the width of a Gaussian filter $K = 2\pi\sigma_d$, and $(m, n) \in [\![-K, K]\!]^2$.

1: Each cell is initialised with a uniform distribution between 0 and 1, i.e.,

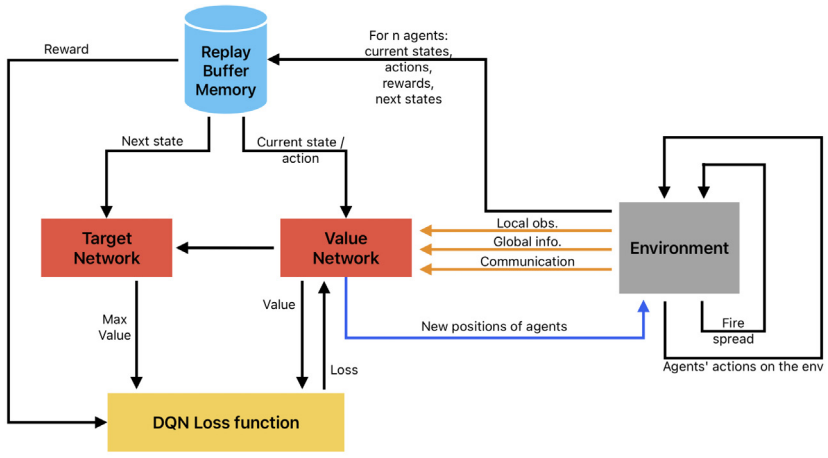
$$d_0(X) \sim \mathcal{U}(0, 1), \qquad \forall X \in G \tag{1}$$

2: Define a 2D-Gaussian filter w_{σ_d} defined as

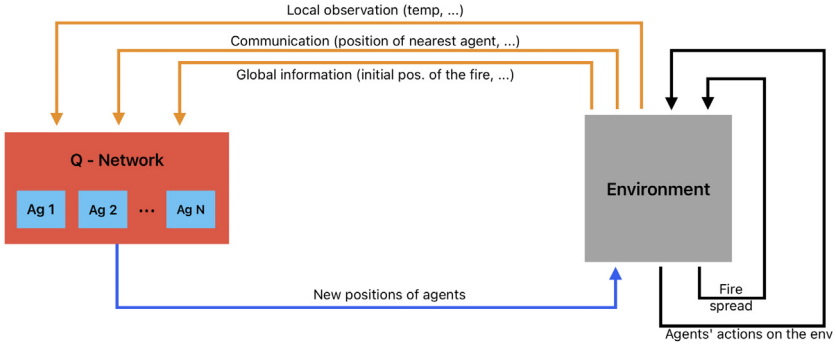
$$w_{\sigma_d}(m, n) = \frac{1}{2\pi\sigma_d^2} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma_d^2}\right)} \tag{2}$$

3: Apply w_{σ_d} to the initial density matrix d_0 by performing a convolution as

$$d_{ij} = \frac{1}{2\pi\sigma_d^2} \sum_{m=-K}^{K} \sum_{n=-K}^{K} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma_d^2}\right)} d_0(i-m, j-n) \tag{3}$$



(a) Training



(b) Testing

Fig. 1. Architecture of the proposed MADQN for SAR missions.

We separate the 2D convolution $d = d_0 * w_{\sigma_d}$ into two one-dimensional convolutions to improve the computation time. The two convolutions are given by

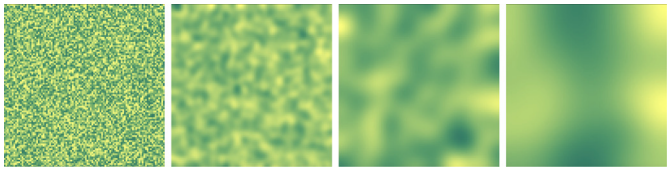
$$d_{ij} = \frac{1}{2\pi\sigma_d^2} \sum_{m=-K}^{K} e^{-\frac{m^2}{2\sigma_d^2}} \sum_{n=-K}^{K} e^{-\frac{n^2}{2\sigma_d^2}} d_0(i-m, j-n) \tag{4}$$

$$d_{ij} = \left(d_0 * w_{\sigma_d}^{1D(i)}\right) * w_{\sigma_d}^{1D(j)} \tag{5}$$

Fig. 2 shows how changes in σ_d produce different density distributions. Notice that small σ_d produces grids with random and discontinuous distributions. This is not useful since forests are discontinuous by nature, but not at this level of granularity. On the other hand, as σ_d increases, a density distribution is obtained that matches what we expect to observe in a forest environment. In this paper, we propose to use $\sigma_d = 10$ to obtain a mid vegetation distribution between the results observed in Figs. 2(c) and (d).

2.1.2. Fire propagation model

The main process of the fire propagation model is similar to that proposed in [31,36]. The model consists of the following steps: (1) it starts with a fire in one cell of the grid with a probability α of transmitting fire to each of its neighbouring cells at each step. Each burning cell has a probability β of "extinguishing," i.e., ceasing to burn and becoming completely burned out. We proposed a similar fire propagation model that includes the proposed vegetation density model which adds a new level of complexity to the environment design.



(a) $\sigma_d = 0$ (b) $\sigma_d = 2$ (c) $\sigma_d = 6$ (d) $\sigma_d = 17$

Fig. 2. Variable Vegetation Density distribution d_{ij}.

To this end, we define a grid of states on G which corresponds to the states of each cell. Each cell can be in one of three different states: either the cell has not yet been reached by the fire and is considered healthy, or it is reached by the fire and is burning, or it is burned and the fire is extinguished due to the loss of vegetation, i.e., dead. Hence, we denote s_X^k as the state of a cell at position $X \in G$ at step $k \in [0, \infty)$ given by

$$s_X^k \in \{\mathcal{H}, \mathcal{F}, \mathcal{D}\} = \{\text{Healthy},\ \text{Fire},\ \text{Dead (burnt)}\}. \tag{6}$$

We consider two types of cells: vegetation and houses. Here, houses are not considered sources of fire propagation, but they can be burnt. We denote Γ as the set of cells containing the positions of all houses. Let us define $I_0(X)$ as the set of immediate neighbours (see Fig. 3) of a cell $X = (i, j)$ as

$$I_0(X) = \{(i-1, j), (i, j-1), (i+1, j), (i, j+1)\}. \tag{7}$$

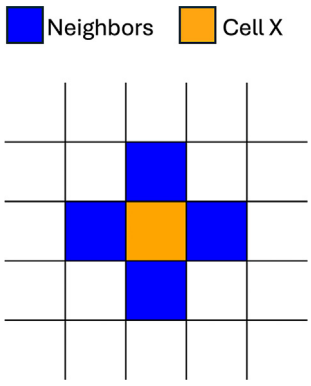


Fig. 3. Immediate neighbours $I_0(X)$ of X.

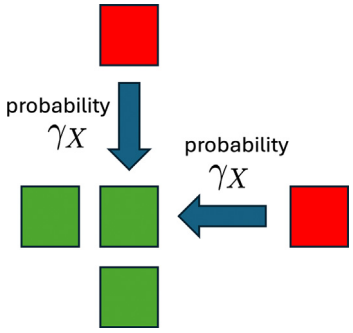


Fig. 4. Visualization of how fire propagates to neighbouring cells. Each burning cell next to a given cell X has probability γ_X to transmit the fire to X.

The proposed fire propagation model can be regarded as a local Markov process (MP) for each cell. Let's define $p^k_{X,s\to s'} := P(s^{k+1}_X \in s' \mid s^k_X \in s)$ as the probability of being at cell X and transitioning from state s to state s' at step k. This local MP can be classified in the following cases

- If a cell is healthy, i.e., $s^k_X \in \mathcal{H}$, then the probability that it catches fire depends on the number of its neighbours on fire. This number, f^k_X, depends on the cell X and the current step k and can be expressed as follows,

$$f^k_X = \sum_{Y\in I_0(X)} \mathbb{1}_{\mathcal{F}}(s^k_Y) \times \mathbb{1}_{G\setminus\Gamma}(Y). \tag{8}$$

A probability of γ_X is given at each step to describe the probability of a healthy cell X, that is next to a burning cell Y, to get fire from Y. This probability depends only on the density of X and the transmission probability α as

$$\gamma_X = \alpha d_X. \tag{9}$$

This process can be observed in Fig. 4. Here, cells with higher vegetation density have a higher probability to get a fire from the neighbouring cells.

Then the transition probabilities from state $\mathcal{H}$ to either state $\mathcal{F}$ or state $\mathcal{H}$ is given by

$$\begin{aligned} p^k_{X,\mathcal{H}\to\mathcal{H}} &= (1-\gamma_X)^{f^k_X} \\ p^k_{X,\mathcal{H}\to\mathcal{F}} &= 1-(1-\gamma_X)^{f^k_X}. \end{aligned} \tag{10}$$

- If a cell X is burning at step k, i.e., $s^k_X \in \mathcal{F}$, then we have two cases: either the cell stops burning at step $k+1$ or it continues burning. We define the burning time of a cell X as

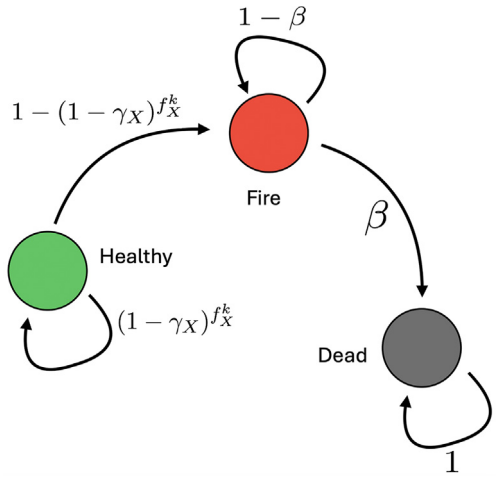


Fig. 5. Local Markov process of the wildfire evolution for a cell X at step k.

$$T^{\mathcal{F}}_X = \text{Card}(\{k \in \mathbb{N} \mid s^k_X \in \mathcal{F}\}). \tag{11}$$

We model the random variable $T^{\mathcal{F}}_X$ as a geometric distribution such as a burning cell X will tend to have a short burning time with minor frequency of long burning times. We use the probability β as parameter of this distribution, i.e., $T^{\mathcal{F}}_X \sim \text{Geom}(\beta)$. From the properties of geometric distribution we have that $\beta = \frac{1}{\mathbb{E}(T^{\mathcal{F}}_X)}$, this implies that the state transition probabilities can be written as

$$\begin{aligned} p^k_{X,\mathcal{F}\to\mathcal{D}} &= p_{\mathcal{F}\to\mathcal{D}} = \beta \\ p_{\mathcal{F}\to\mathcal{F}} &= 1-\beta. \end{aligned} \tag{12}$$

This means that the probability that a burning cell extinguishes at a given step k does not depend on k or X but it always depends on β.

- If a cell X is burnt or dead, i.e., $s^k_X \in \mathcal{D}$, then the cell X can only remain dead. Therefore, we have only on state transition probability given by

$$p_{\mathcal{D}\to\mathcal{D}} = 1. \tag{13}$$

Therefore, the local state transition process can be summarized in Fig. 5 and satisfies the following state transition matrix $\boldsymbol{P}^k_X$ of cell X at time step k, which is given by

$$\boldsymbol{P}^k_X = \begin{bmatrix} (1-\gamma_X)^{f^k_X} & 1-(1-\gamma_X)^{f^k_X} & 0 \\ 0 & 1-\beta & \beta \\ 0 & 0 & 1 \end{bmatrix}. \tag{14}$$

Fig. 6 shows the mean final proportion of burnt cells for 10 different values of β in the range of $[0.1, 1.0]$. This graph is used to select the best values for α and β. The results show that more cells are burnt when α is large and β is small. This is the behaviour that we are interested in since we want to generate wildfires that can burn a large proportion of cells to demonstrate the capabilities of the proposed MADRL for SAR missions. To this end, we set $\alpha = 0.25$ and $\beta = 0.1$ which means an average final proportion of burnt cells of 90 %, and a thin fire front. This configuration can be regarded as the worst-case scenario, i.e., the fire propagates fast, such that fires that propagate slowly can be managed relatively easier by the proposed MADRL.

Fig. 7 shows a fire propagation example in a variable vegetated environment using the proposed α and β values. The results show that the proposed wildfire simulation has thin and discontinuous fire front with a relative complex shape compared with traditional radial propagation

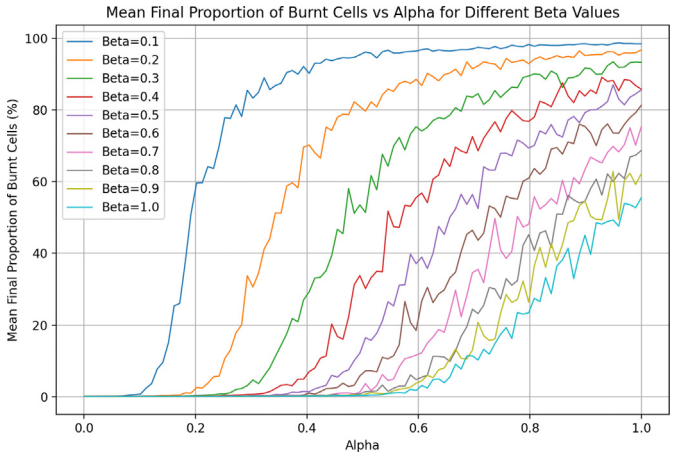
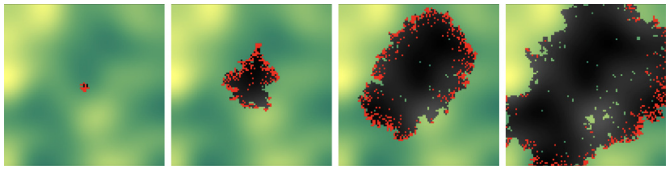


Fig. 6. Mean final proportion of burnt cells as a function of α for 10 different β values.



(a) $k = 5$ (b) $k = 50$ (c) $k = 250$ (d) $k = 500$

Fig. 7. Wildfire simulation.

shapes. Notice that the fire struggles to propagate in areas with low vegetation density, which is consistent with the real behaviour of wildfires.

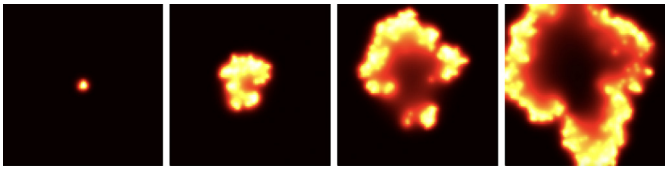
Remark 1. The proposed environment model design aims to emulate the stochastic spread of wildfires considering variable vegetation density. Future work aims to match and adapt the proposed fire propagation model using real-world fire data to enhance its realism and generalisation across different vegetation types.

2.1.3. Temperature modelling

Traditional fire detection approaches are based on cameras embedded in the drones to observe the environment and localized fire. However, these cameras are susceptible to high number of false positives due to lightning condition issues or disturbances (e.g., smoke). To overcome this issue, we propose using temperature as observation method instead of cameras. Temperature measurements are obtained from thermal cameras installed on UAVs [32,37]. Here, the main challenge of using this approach lies in the modelling of heat associated with the wildfire with enough realism and computational efficiency.

To solve this issue, several approaches use heat equations across diverse scenarios [38], which have been shown to be computationally intensive. Therefore, we propose a new method that emulates heat diffusion and heat cooling in a 2-dimensional space. The key idea is to use a recursive relation between the future temperature grid T^{k+1} at step $k+1$ and the current temperature grid T_k at step k as follows,

1. **Initialisation:** each burning cell at step k is set at temperature $T_{\max}$, while other cells retain their temperature: $T_X \leftarrow T_{\max} \times \mathbb{1}_{\mathcal{F}}(s_X) + T_X \times (1 - \mathbb{1}_{\mathcal{F}}(s_X))$, where $\mathbb{1}_{\mathcal{F}}(s^k) = \left(\mathbb{1}_{\mathcal{F}}\left(s^k_{(i,j)}\right)\right)_{(i,j)\in G}$ denotes the matrix where all coefficients corresponding to burning cells are equal to 1, otherwise are set to 0.



(a) $k = 5$ (b) $k = 50$ (c) $k = 250$ (d) $k = 500$

Fig. 8. Wildfire temperature grid.

2. **Temperature diffusion modelling**: the temperature matrix T_X is convoluted with a Gaussian kernel of parameter (diffusion factor) σ_T, i.e., $T \leftarrow T * w_{\sigma_d}$ with $T = (T_X)_{X=(i,j)\in G}$.
3. **Cooling after burning modelling**: the matrix T_X is multiplied by a cooling factor $c \in [0, 1]$: $T \leftarrow cT$.

This process can be reduced to iterating the following relation,

$$T^{k+1} = c\left(T_{max}\mathbb{1}_{\mathcal{F}}(s^{k+1}) + \mathbb{1}_{\mathcal{H}\cup\mathcal{D}}(s^{k+1}) \odot T^k\right) * w_{\sigma_T}. \tag{15}$$

In this paper, we set the following values $T_{\max} = 100$, $\sigma_T = 1$ and $c = 0.965$. Here, a low standard deviation Gaussian filter enables spreading the heat locally. This fits the proposed requirements, that is, the temperature anomaly will be almost zero outside of the burning / burnt cells, but will increase rapidly in front of the fire front. On the other hand, a cooling value c close to 1 provides a progressive cooling behind the fire front. Fig. 8 shows a temperature grid example per step k. The results show a realistic evolution of the temperature grid that can be used as input for the MADRL training for SAR missions, which is discussed in the next section.

2.2. Multi-agent deep Q-network

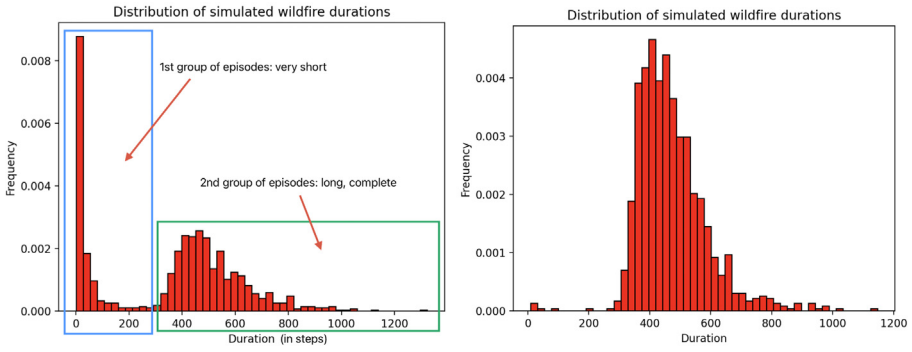
In this paper, we use a multi-agent deep Q-network (MADQN) under a centralized training - decentralized execution (CTDE) framework. The key idea is that the agents share global information about the environment and their experiences. In addition, in the testing part the agents operate in a decentralized mechanism such that they make decisions using their own copy of the trained model. This framework has been used in both value-based and policy-based (e.g., proximal policy optimization (PPO)) algorithms [39,40].

The proposed MADQN implementation uses a single network for choosing actions and a single target network for all the agents. We use M to denote the number of UAV agents in the swarm. This network uses a batch of previous experiences sampled from a replay buffer as shown in Fig. 1. During the testing phase, each agent receives a copy of the neural network with parameter θ and each agent uses the same policy π_θ for the action selection. Here, the agents are communicating with each other and receive global information about the environment state s. In this paper, the neural networks of both the Q-network and target network will have 2 hidden layers of respectively 512 and 256 neurons using ReLU as activation functions.

2.2.1. Pre-training modifications

Due to the dynamic nature of the forest fires generated in the proposed model, it can be observed that some fires are not useful for training episodes of the MADQN algorithm. Indeed, some episodes might be short because the fire does not spread to any neighbouring cells according to the proposed transmission probability model and vegetation. To visualize this, a histogram of the duration of forest fires from 1000 simulations is shown in Fig. 9(a).

We can observe that there are short forest fires with less than 200 steps of duration. These episodes are problematic because the fire does not have the chance to spread to the edges of the grid and has even extinguished shortly after igniting. These fires need to be removed to avoid biased behaviour in the RL-training. We distinguish two groups:



(a) Before modification (b) After modification

Fig. 9. Forest fire durations.

(1) short forest fires with $k < 200$ steps, and (2) long forest fires with $k > 200$ steps. We additionally distinguish two subgroups of group (1) defined as follows,

- **Subgroup (1.1)**: This subgroup consists of forest fires that last a short time because the fire is initially present on a single cell in the center of the grid, such that it was unable to spread to its neighbouring cells. This outcome is possible according to the proposed model and its transition probabilities. Here, the duration is less than $k < 5$ steps.
- **Subgroup (1.2)**: This subgroup contains forest fires that could not spread to very large areas because of their initial fire zone. Here, the middle of the grid has a low vegetation density such that the probability of spreading is too low. The duration time of these fires is greater than $k \geq 5$ steps but less than 200 steps.

To eliminate the wildfires from subgroup (1.1), we propose to increase the number of cells that are initially on fire from one to nine (a 3×3 square located in the middle of the grid). This way, the initially burning cells will have a much higher probability of spreading the fire to their neighbours in the next time step. To eliminate the wildfires from subgroup (1.2), we simply propose regenerating the vegetation density grid, if it has an average density lower than 0.5 within a 3×3 square around the initial fire position. This approach eliminates the short forest fires and results in the distribution shown in Fig. 9(b).

2.2.2. Search and rescue (SAR) mission

The main problem to be solved by the MADQN is a Search and Rescue (SAR) mission where the agents/drones not only need to explore an unknown forest environment with a progressing fire, but also need to locate houses threatened by the advancing forest fire. This means that we need to include houses in the proposed forest environment. To this end, we label N_h random positions in the grid as houses. We set $N_h = 40$ until the end of this work. An example of the forest environment that includes houses is shown in Fig. 10.

We define $I_2(X) = \{X + (i,j) \mid (i,j) \in \{-2,0,2\}^2\}$, the set of positions that are less than two units of distance away from X, including corners. I_2 can be visualized in Fig. 11. Throughout this part, the set $\overline{I}_2^k = \bigcup_{X \in \mathcal{M}^k} I_2(X)$ represents all the cells of G that are observed by at least one agent. We recall that Γ is the set of the positions of G occupied by houses, and $\mathcal{M}^k$ is the set of the positions occupied by the agents at step k. Then, let Γ_v^k be the set of houses that have been visited at step k while still being healthy

$$\Gamma_v^k = \bigcup_{k'=0}^{k} \left(\Gamma \cap H^{k'} \cap \overline{I}_2^{k'} \right), \tag{16}$$

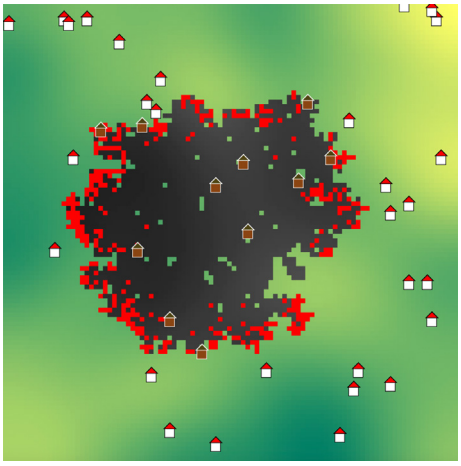


Fig. 10. Example of an environment where houses have been randomly added, with $N_h = 40$. Burnt houses are shown in brown, healthy houses (the ones to be visited by the trained agents) are still in white and red.

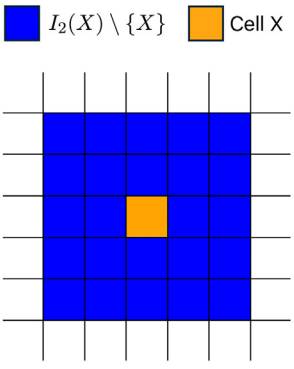


Fig. 11. Visualization of the $I_2(X)$ set, blue cells represent neighbouring cells included in I_2, while I_2 contains blue cells and X itself.

with $H^{k'} = \{X \in G \mid s_X^{k'} \in \mathcal{H}\}$. We define $\kappa_{2,\mathcal{E}}$ as the score function for the proposed problem, defined as the proportion of houses that have been visited while not being burnt, after an infinite number of steps given by

$$\kappa_{2,\mathcal{E}} = \lim_{k \to \infty} \frac{\text{Card}(\Gamma_v^k)}{\text{Card}(\Gamma)} \tag{17}$$

2.2.3. Agents modelling

The agents initially start at random positions located on the edges of the grid, namely on the four borders of the environment [41]. They can also move freely in the grid, according to the action space that will be defined in the next section. They can observe the environment around them, within their I_2 space. The agent modelling is quite simple, that is, when a given house h is in the neighbouring cells of an agent (in the sense of I_2), it becomes included in Γ_v^k. The corresponding probability can be inferred from (16) as

$$P(h \in \Gamma_v^k \mid h \notin \Gamma_v^{k-1}) = \mathbb{1}(s_h^k \in \mathcal{H}) \cdot \mathbb{1}(h \in \overline{I}_2^k) \tag{18}$$

For realism purposes, agents have to deal with one constraint, i.e., when they identify a house, they have to stop moving and stay next to the localized house for 20 steps. During this time, according to the problem definition, they can localize people, perform their evacuation task and transmit the localisation of the recognized people to the fire fighters. All these are not visible nor implemented in our environment and we merely stop the UAVs' movements during these 20 steps.

Remark 2. The proposed hovering time is heuristically selected to emulate the search operation and communication latency between the UAVs and the fire fighters, whilst considering the urgency of moving forward as fast as possible to search new houses and potential victims. This time needs to be optimised or adapted as given in Remark 3.

Remark 3. The proposed time constraint needs to be optimised or adapted to the findings of the UAV. Here, the hovering time could be larger or smaller depending on how many people were detected/evacuated and the time it takes to transmit the information to firefighters. Here, human-in-the-loop algorithms could be incorporated to optimise the hovering time as well as federated architectures to optimise the information transmission. These are open challenges that are not addressed in this paper.

2.2.4. State and action spaces

For an agent m, at step k, the state space consists in the following features

- A *presence* map: One of the main challenges in search/exploration tasks in almost continuous environments (small cell size) is to provide agents with data informing agents which zones have already been explored enough and which have not, so that they can move towards areas that lack exploration. Therefore, agents need to have as a state feature data about the neighbouring cells informing which, if the agent moves on them at the current step, have the highest probability to lead to an area lacking exploration and which have the lowest. This reasoning allows to define the variable *presence*, defined on the whole grid, that indicates whether a cell is in a much visited area or not.

  To define this matrix, we simply keep track of all the cells on which the agents have been at least once since the beginning of the episode. The presence matrix Ps^k gets one values at coordinates belonging a certain history of agents' positions that we will specify later. We then convolute the obtained matrix with a Gaussian filter of parameter σ_p. This way, the spreading effect obtained through the Gaussian filter defines areas where the "presence" is more significant. The hyperparameter σ_p allows to control the width of these areas. Its value must be chosen carefully, that is, with a σ_p that is too small, the areas will be too "precise" and centered on the visited positions, which will not provide to many information to the agents. Conversely, a σ_p that is too large defines areas of high presence that are too wide compared to reality, in which houses might have been missed by the UAVs. Then, the history of positions that we convolute has to be specified. If we select the complete history, i.e., all the current and past positions of the agents until step k, the agents will always be at the center of a zone in which the presence is high, because their current cell is in the history on which we apply the Gaussian filter. Consequently, it is required to apply our Gaussian kernel on a history of positions until step $k-l$ instead of step k, with l the *lag*, the value of which needs to be optimised.

  By experimentation that we will discuss in the next section, we found that the optimal values for σ and l are $(\sigma, l) = (4, 6)$. Lastly, we recall that Ps^k is a matrix that contains values informing the agents where they should move or not to improve their area coverage. Again, by experimentation we found that the agents do not have to move 'behind' the fire front. Indeed, their task is mainly to discover healthy houses, not to assist potential victims that have already been reached by the fire. Thus, we add a second term B^k in the definition of our matrix, containing the cells that are burning while being in the agents' cumulative observation space $\bar{I}_2^k$. This quantity is calculated using an estimate of the set containing the burning cells, because UAVs do not have the ability to establish the real state of a cell, but they can estimate it using temperature. The following formula specifies how to calculate matrix Ps at each step $k > l$

$$\mathrm{Ps}^k = \left(\mathbb{1}\left((i,j) \in A^{k,l} \cup B^k\right)\right)_{(i,j)\in G} * w_{\sigma_p} \tag{19}$$

with,

$$\begin{aligned} A^{k,l} &= \bigcup_{k'=0}^{k-l} \bigcup_{m\in\mathcal{M}^{k'}} m \\ B^k = \hat{F}^k \cap \overline{I}_2^k &= \{X \in G \mid T_X^k > \tfrac{T_{\max}}{2}\} \cap \overline{I}_2^k \end{aligned} \tag{20}$$

Conversely if $k \leq l$, then $\mathrm{Ps}^k = \left(\mathbb{1}\left((i,j) \in B^k\right)\right)_{(i,j)\in G} * w_{\sigma_p}$. The input provided to the agent is a local 3×3 map obtained using the previous matrix given by

$$p_m^k = (\ln \mathrm{Ps}^k(i,j) - \ln \mathrm{Ps}^k(X_m^k))_{(i,j)\in I_1(X_m^k)} \tag{21}$$

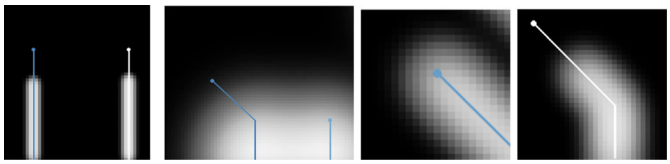
Fig. 12 shows the influence of varying σ_p and l on the Ps grid.

- The initial position of the fire (X_F^0) relative to the position of the agent X_m^k given by $u_m^k = X_F^0 - X_m^k$;
- The current speed of the UAV, that we define as the distance between the agent's current position and its position 5 steps ago: $v_m^k = ||X_m^k - X_m^{k-5}||_2/5$. As the proposed task requires that the agents to move as fast as possible, this feature will be useful;
- If m_c^k is the closest agent to m, then we use as input the position of m_c^k relative to the one of m given by $u_{c,m}^k = X_{m_c}^k - X_m^k$.

Therefore, the feature vector is given by

$$s_m^k = [p_m^k \;\; u_m^k \;\; v_m^k \;\; u_{c,m}^k]^t. \tag{22}$$

The action space consists of 8 possible moves: up, down, left, right, upper left, upper right, lower left, lower right.



(a) $\sigma_p = 1$ (b) $\sigma_p = 10.$ (c) $l = 0$ (d) $l = 50$

Fig. 12. Examples of the Ps *presence* grid around agents.

2.2.5. Reward function

The reward function is designed and tuned manually until the best performance is obtained in the proposed SAR operation. Let N_h be the number of houses. At step k, for agent m, we split the reward into four components,

- A reward $r^k_{\text{visit,m}}$ obtained when the agent visits a house in $I_2(X^k_m)$. In this case, the agent receives a reward that is high but decreases exponentially if the house is far away from the fire start. This way, agents are enticed to search for houses around the fire start, which corresponds to the area where the houses are the most threatened by the fire. The agents can then search for houses near the edges of the grid but get a smaller reward. If the agents discover a house that is already burnt, they get a low positive reward.
- A reward $r^k_{\text{pres,m}}$ obtained if the agent moved to a cell that has a lower *presence* than its previous one ($+\frac{1}{10M}$ in this case), otherwise it receives a negative reward $-\frac{1}{10M}$, where M is the number of UAVs.
- A reward $r^k_{\text{speed,m}}$ that increases with the agent's speed. As a result, the agents are encouraged to move faster and explore larger areas.
- A reward $r^k_{\text{close,m}}$ that increases with the distance to the agent that is the closest to m. With such a reward, the agents may learn not to stay too close to each other, a behaviour that can limit the agents' efficiency and avoid multiple agents in the same zone.

The final reward function is given by a linear combination of each sub-reward with unitary weights, i.e.,

$$r^k_m = r^k_{\text{visit},m} + r^k_{\text{pres},m} + r^k_{\text{speed},m} + r^k_{\text{close},m}. \tag{23}$$

The corresponding algorithm, with detailed reward definitions, is given in Algorithm 2.

Remark 4. Each sub-reward component plays a pivotal role in the success and generalisation of the proposed MADQN, such that removing one or more sub-reward components will compromise the overall performance.

Algorithm 2 Reward calculation algorithm.

Initialize rewards $r^k_{\text{visit},m} = r^k_{\text{pres},m} = r^k_{\text{speed},m} = r^k_{\text{close},m} = 0$
for $X \in I_2(X^k_m)$ **do**
  if $X \in \Gamma$ and $X \notin \Gamma^k_v$ **then**
    if $s^k_X \in \mathcal{H}$ **then**
      Calculate distance to center $\text{dist}^k_{c,m}$
      $r^k_{\text{visit},m} = \frac{100}{N_h} \exp\left(-\frac{\text{dist}^k_{c,m}}{N/2}\right)$
    else if $s^k_X \in \mathcal{D} \cup \mathcal{F}$ **then**
      $r^k_{\text{visit},m} = \frac{10}{N_h}$
    end if
  end if
end for
if $\text{Ps}^{k+1}_X - \text{Ps}^k_X < 0$ **then**
  $r^k_{\text{pres},m} = \frac{1}{10M}$
else
  $r^k_{\text{pres},m} = -\frac{1}{10M}$
end if
$r^k_{\text{speed},m} = \frac{v^k_m}{40M}$
Determine closest agent m^k_c
Calculate distance to m^k_c: $\text{dist}^k_m = \|X^k_{m^k_c} - X^k_m\|_2$
$r^k_{\text{close},m} = (1 - \frac{1}{\text{dist}^k_m})/(100M)$
Total reward at step k: $r^k_m = r^k_{\text{visit},m} + r^k_{\text{pres},m} + r^k_{\text{speed},m} + r^k_{\text{close},m}$

Table 1
Final hyperparameter values.

Hyperparameter	Value
learning_rate lr	10^{-4}
discount factor γ	0.98
ε-probability	0.12
batch_size	32

2.2.6. Hyperparameters

We perform hyperparameter tuning of the main MADQN hyperparameters: (lr, γ, ε, batch_size). Table 1 summarizes the best hyperparameter values.

2.3. Heuristic-based algorithm

We design a heuristic-based model for comparison purposes with the MADQN architecture used in this paper. In the literature, plenty of algorithms exist to explore environments without reinforcement learning [42], and this research area has been widely covered, even for swarms of agents [43–45]. These algorithms have been proven to be efficient in grid exploration and discrete environment exploration. However, we identify two characteristics of most of these algorithms that make them incompatible with the requirements of the proposed task:

- These algorithms most often aim to be efficient by minimizing the number of nodes or edges visited more than once, with no regard to exploration time. However, in this case, visiting a cell more than once is not important, as the objective is to explore as much area as possible and as quickly as possible.
- The proposed environment is dynamic due to the fire propagation modelling. As a result, the graph to be explored changes with time, as it is of no use exploring already burnt areas. Indeed, applying an algorithm that was designed to explore a static grid and migrate it into a grid where some regions do not have the same importance compared with others (e.g., in burned zones) would be less efficient.

As a consequence, we hypothesize that an exploration algorithm based on heuristics has to be designed for this specific task. Therefore, we design an efficient heuristic algorithm that considers the constraints imposed by both the task and the environment design. Then, the algorithm will use the *presence* data to determine towards which direction it will move at each step. The principle of the proposed heuristic algorithm is summarized in Algorithm 3.

Algorithm 3 Heuristic-based UAV algorithm.

Initialize UAV positions $(X_m)_{m\in\mathbb{N}_M}$
while $\text{Card}(\{s^k_X \in \mathcal{F}\}) \neq 0$ **do**
  for $m \in [\![1, M]\!]$ **do**
    Calculate local relative log *presence* map p^k_m
    if $\min_{dX' \in \mathcal{A}} p^k_m(dX') < 0.5$ **then**
      Calculate next move, to non-burning neighbor of minimal presence:
      $dX = \underset{dX' \in \mathcal{A} \mid T^k(X+dX') < T_{\max}/2}{\arg\min} p^k_m(dX')$
    else
      Calculate the minimal presence's position: $X^k_{\text{minPs}} = \underset{X \in G}{\arg\min}\, \text{Ps}^k(X)$
      Calculate next move to this minimum: $dX = \underset{dX' \in \mathcal{A}}{\arg\min} \|X^k_{\text{minPs}} - (X^k_m + dX')\|_2$
    end if
    Move UAV on $X^k_m + dX$
  end for
end while



Fig. 13. Reward and score curves for 1000 epochs.

3. Results and discussion

In this section, we train the proposed model. The trainings are performed with 5 agents ($M = 5$) and the results correspond to the $M = 5$ case. We also perform ablation studies to identify how the model is affected by changing specific hyperparameters, e.g., the number of agents.

3.1. Reward curve

The reward curve is shown in Fig. 13. We additionally plot the score curve (score obtained throughout the episodes of the training) in this graph. All the reward components are also exhibited. These curves are obtained during the training using a moving average with a window size of 50.

We notice that the reward increases until epoch 200 and then stabilises until the end of the training. All the reward components increase, as well as the score which improves in each episode. Here, the *close* reward component remains with low values which means that the agents do not learn to avoid staying too close to one another. However, the experiments show that removing this component slightly decreases the final score, hence we decide to keep it. Nonetheless, the reward curve demonstrates that the agents learn to explore the environment and find houses, by minimizing their *presence* value and maintaining high speeds.

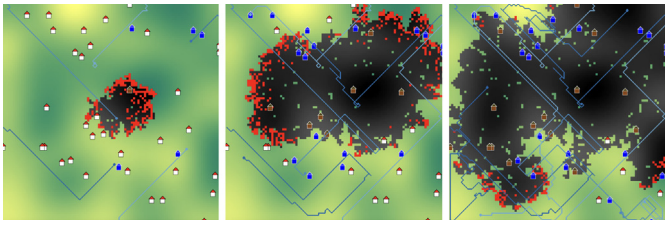
Fig. 14 shows an example of how the trained agents search for houses and help people evacuate from non-burnt houses in an environment where a wildfire spreads.

In addition, we show in Fig. 15 a Ps grid at the end of a simulation.

We observe in both Figs. 14 and 15 some interesting behaviours that need to be discussed

- The agents, or at least some of them, seem to initially rush to the middle of the grid, certainly because of the *visit* reward which decreases with the distance to the middle of the grid.
- Then, the agents quickly try to move as far as possible from the fire, since they do not need to stay in burnt areas where they no longer have houses to visit.
- Their trajectories are mainly straight, which is a synonym for high speed in the proposed simulations and can be attributed to the r_{speed} reward.
- Most of the time, the agents try to remain in unexplored areas, where Ps is low. This behaviour is achieved because the agents try to cover as much area as possible.

All these elements show that the training was efficient. We nonetheless observe some unexpected behaviours, for instance, some agents cross the fire front and move within burnt areas. This remains a minority behaviour.



(a) $k = 100$ (b) $k = 250$ (c) $k = 500$

Fig. 14. Example of a simulation of the SAR mission using MADQN.

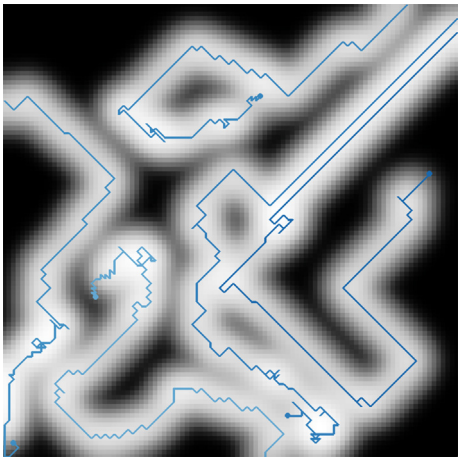
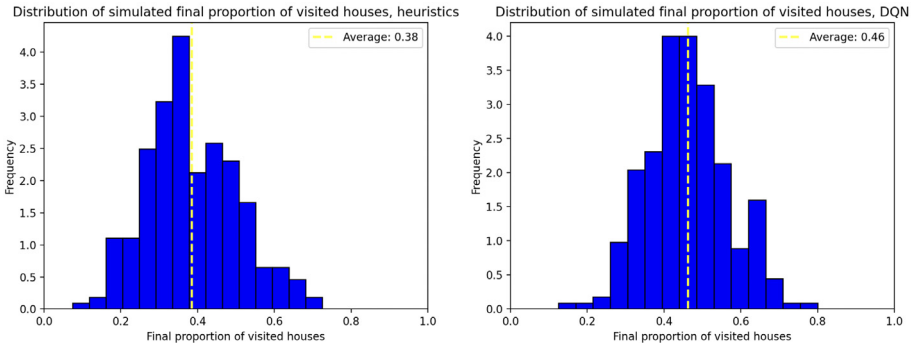


Fig. 15. Ps grid for $k = 500$.

3.2. Comparison studies

We compare the proposed MADQN algorithm with the heuristic-based algorithm to show the benefits and challenges of the approach. A histogram of the distribution of scores obtained from 500 simulations of the SAR mission on wildfires is plotted. This histogram is displayed for both the heuristic-based algorithm and the MADQN model in Fig. 16.

The distributions show that the MADQN model scores slightly better than the heuristic-based one. Here, the score difference is about 8 %. As a reminder, this score means that, using MADQN, the agents get to visit on average 46 % of the healthy houses before the wildfire ends. The heuristic algorithm manages to visit only 38 %.



(a) Heuristic algorithm (b) MADQN algorithm

Fig. 16. Distribution of the scores obtained on 500 simulations, and corresponding average score.

3.3. Ablation studies

We perform hyperparameter tuning for the σ_p and l parameters, namely the size of the explored zones in the *presence* grid, and the last l values that we do not take into account, whilst obtaining the history of positions for the calculation of Ps (the *lag*). This optimization will allow us to determine if these hyperparameters are useful, i.e., if the proposed approach based on the Ps matrix is relevant, and what values enable maximizing $\hat{\mathbb{E}}(\kappa_{2,\mathcal{E}})$. We consider the following procedure,

- We first train the proposed MADQN model using parameter values chosen by common sense and by examining the graphical output of the simulations;
- We calculated with the obtained model the score values after modifying the σ_p and l parameters in the environment, which means that the model is used on an environment that does not have the same parameters as the one it was trained with. This is a flaw of our approach, however, it enables saving an enormous amount of training time and consequently gaining precision when calculating the score. This means that with a pre-trained model, we can calculate its score on more simulations, in this case 50 per combination of parameters, which is still infinitely faster than training a model per combination;
- We modify the environment's parameters σ_p and l using the ones that maximize the score and retrain the agents on this new environment.

We detail the ranges of values tested as well as the optimal values obtained in Table 2.

The plot obtained using all the possible combinations of these parameters is given in Fig. 17, which shows the possible combinations of the parameters σ_p and l. We observe that low values of σ_p (1 and 2) yield low scores, which demonstrates the usefulness of defining large enough zones of past *presence*, so that the agents know which areas have already been visited. Conversely, too high values of σ_p decrease the score, but in low magnitude compared to the low ones. In addition, we study the influence of l. According to Fig. 17, a *lag* of zero ($l = 0$) significantly decreases the performance for $\sigma_p \geq 3$. This observation demonstrates the usefulness of the *lag* variable. However, if $l \neq 0$, l does not significantly influence the score for the σ_p values that lead to high scores. We note that $(\sigma_p, l) = (4, 6)$ maximizes the score.

After analysing the influence of σ_p and l in the proposed model, we examine the impact of another important feature introduced in Ps given by B^k in (20). This term is used to encourage the agents not to move towards the fire front, as burning cells are given a high *presence* value.

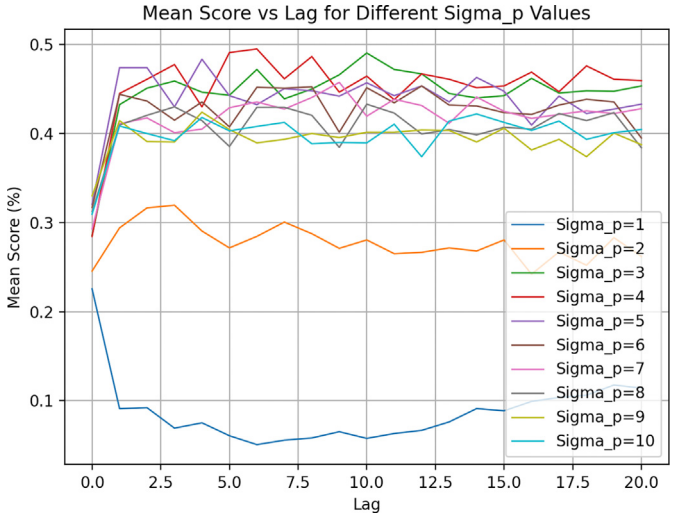


Fig. 17. Scores obtained with a pre-trained model as a function of the lag l, for different σ_p values. Except for $l = 0$ which yields low scores (for $\sigma_p \geq 3$), the variable l does not seem to have a major influence on the score.

To analyse this property of the model, we plot in Fig. 18 the score against the number of agents for three models: (1) the trained MADQN model with B^k, (2) a MADQN model without B^k, and (3) the heuristic-based algorithm denoted as dummy.

The curves show that the score grows as the number of UAVs increases and that the heuristic (dummy) algorithm scores less despite the number of agents increasing in the environment. This plot also indicates that the B^k term has a small influence or no influence on performance for some values of M, while it increases the score for other values. In a nutshell, B^k never yields a drop in performance while increasing it in the best case. This allows us to conclude that B_K has a positive influence on the score of the MADQN model.

Table 2
Range of values for σ_p and l.

Parameter	Range tested	Best value
σ_p	$[\![1, 10]\!]$	4
l	$[\![0, 20]\!]$	6

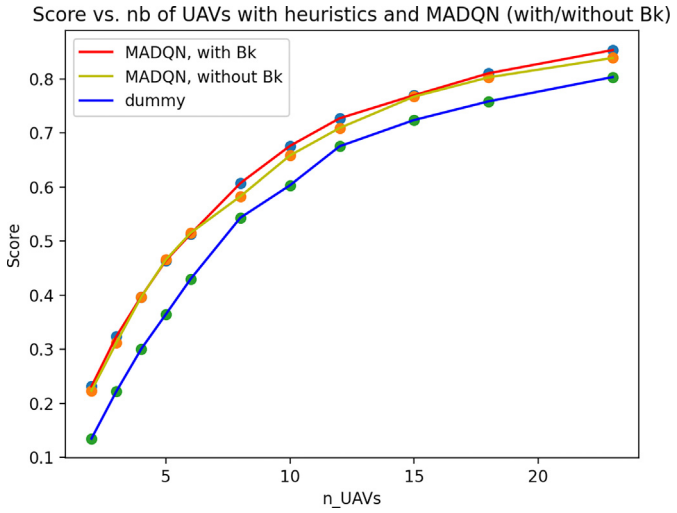


Fig. 18. Score as a function of the number of UAVs M, for three different settings heuristic(dummy) algorithm, MADQN-based model with the B^k term, MADQN-based model without the B^k term.

4. Conclusion

In this paper a search and rescue application in wildfire environments using UAVs is proposed. Here, a novel forest environment that considers variable density vegetation is introduced. This helps to model more accurately the evolution of a wildfire. In addition, we consider a temperature map that provides a better insight into zones that need assistance compared with cameras, which are prone to high false positives. Then, a multi-agent deep Q-network is designed to search for houses that might shelter people who need to be evacuated and are threatened by the fire. In addition, a heuristic-based algorithm is developed for comparison purposes with the approach. We demonstrated that agents can successfully carry out this kind of exploration and SAR mission, being able to identify nearly 50 % of the threatened houses and outperform the heuristic model.

Future work will consider the incorporation of additional elements in the environment and agent design such as wind, humidity, battery life, and delays/noise in the sensor measurements. We are interested in incorporating heuristics into the RL design to improve the overall SAR mission, as well as integrating robust RL models to enhance resilience and robustness in the swarm operation.

CRediT authorship contribution statement

Maxime Collignon: Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Adolfo Perrusquía:** Writing – original draft, Supervision, Project administration, Methodology, Conceptualization. **Antonios Tsourdos:** Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **Weisi Guo:** Writing – original draft, Visualization, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] A.A. Deo, D. Ganer, G. Nair, Tropical cyclone activity in global warming scenario, Nat. Hazards 59 (2011) 771–786, https://doi.org/10.1007/S11069-011-9794-8.
[2] S.C. Chapman, N.W. Watkins, D.A. Stainforth, Warming trends in summer heatwaves, Geophys. Res. Lett. 46 (3) (2019) 1634–1640.
[3] J. Xu, D. Panagopoulos, A. Perrusquía, W. Guo, A. Tsourdos, Generalising rescue operations in disaster scenarios using drones: a lifelong reinforcement learning approach, Drones 9 (6) (2025) 409.
[4] Y. Hirabayashi, R. Mahendran, S. Koirala, L. Konoshima, D. Yamazaki, S. Watanabe, H. Kim, S. Kanae, Global flood risk under climate change, Nat. Clim. Change 3 (2013) 816–821, https://doi.org/10.1038/NCLIMATE1911.
[5] C. Gruffeille, A. Perrusquía, A. Tsourdos, W. Guo, Disaster area coverage optimisation using reinforcement learning, in: 2024 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2024, pp. 61–67.
[6] Y. Liu, J.A. Stanturf, S.L. Goodrick, Trends in global wildfire potential in a changing climate, For. Ecol. Manag. 259 (2010) 685–697, https://doi.org/10.1016/J.FORECO.2009.09.002.
[7] A. Kumar, A. Perrusquía, S. Al-Rubaye, W. Guo, Wildfire and smoke early detection for drone applications: A light-weight deep learning approach, Eng. Appl. Artif. Intell. 136 (2024) 108977.
[8] S. F. B. Tett, P.A. Stott, M.R. Allen, W.J. Ingram, J. F. B. Mitchell, Causes of twentieth-century temperature change near the earth's surface, Nature 399 (6736) (1999) 569–572.
[9] M. Krawchuk, M. Moritz, M. Parisien, J. Van Dorn, K. Hayhoe, Global pyrogeography: the current and future distribution of wildfire, PLoS ONE 4 (2009) https://doi.org/10.1371/journal.pone.0005102.
[10] T. Paveglio, H. Brenkert-Smith, T. Hall, A. Smith, Understanding social impact from wildfires: advancing means for assessment, Int. J. Wildland Fire 24 (2015) 212–224, https://doi.org/10.1071/WF14091.
[11] P. Palaiologou, A.A. Ager, M. Nielsen-Pincus, C.R. Evers, M.A. Day, Social vulnerability to large wildfires in the Western USA, Landsc. Urban Plan. 189 (2019) 99–116.
[12] L.A. Palinkas, The California Wildfires, Springer International Publishing, Cham, 2020, pp. 53–67.
[13] J. Bayham, J. Yoder, P. Champ, D. Calkin, The economics of wildfire in the united states, Annu. Rev. Resour. Econ. (2022) https://doi.org/10.1146/annurev-resource-111920-014804.
[14] D. Wang, D. Guan, S. Zhu, M.M. Kinnon, G. Geng, Q. Zhang, H. Zheng, T. Lei, S. Shao, P. Gong, S.J. Davis, Economic footprint of California wildfires in 2018, Nat. Sustain. 4 (3) (2021) 252–260.
[15] O.V. Sanderfoot, S. Bassing, J. Brusa, R.L. Emmet, S. Gillman, K. Swift, B. Gardner, A review of the effects of wildfire smoke on the health and behavior of wildlife, Environ. Res. Lett. 16 (2021) https://doi.org/10.1088/1748-9326/ac30f6.
[16] W. Tekalign, Impacts of wildfire and prescribed fire on wildlife and habitats: a review, J. Nat. Sci. Res. 6 (2016) 15–27, https://api.semanticscholar.org/CorpusID:55057559.
[17] V. Chitale, M.D. Behera, How will forest fires impact the distribution of endemic plants in the himalayan biodiversity hotspot?, Biodivers. Conserv. 28 (8) (2019) 2259–2273.
[18] M. Jhariya, A. Raj, Effects of wildfires on flora, fauna and physico-chemical properties of soil-an overview, J. Appl. Nat. Sci. 6 (2014) 887–897, https://doi.org/10.31018/JANS.V6I2.550.
[19] M. El Debeiki, S. Al-Rubaye, A. Perrusquía, C. Conrad, J.A. Flores-Campos, An advanced path planning and uav relay system: Enhancing connectivity in rural environments, Future Internet 16 (3) (2024) 89.
[20] E. Bildik, A. Tsourdos, A. Perrusquía, G. Inalhan, Decoys deployment for missile interception: A multi-agent reinforcement learning approach, Aerospace 11 (8) (2024) 684.
[21] C. A.B. Baker, S. Ramchurn, W.T. Teacy, N.R. Jennings, Planning search and rescue missions for uav teams, in: ECAI 2016, IOS Press, 2016, pp. 1777–1782.
[22] B. Ai, M. Jia, H. Xu, J. Xu, Z. Wen, B. Li, D. Zhang, Coverage path planning for maritime search and rescue using reinforcement learning, Ocean Eng. 241 (2021) 110098.
[23] A. Rahman, A. Bhattacharya, T. Ramachandran, S. Mukherjee, H. Sharma, T. Fujimoto, S. Chatterjee, Adversar: Adversarial search and rescue via multi-agent reinforcement learning, in: 2022 IEEE International Symposium on Technologies for Homeland Security (HST), 2022, pp. 1–7, https://doi.org/10.1109/HST56032.2022.10025434.
[24] X. Cao, M. Li, Y. Tao, P. Lu, Hma-sar: Multi-agent search and rescue for unknown located dynamic targets in completely unknown environments, IEEE Robot. Autom. Lett. 9 (6) (2024) 5567–5574, https://doi.org/10.1109/LRA.2024.3396097.
[25] C. Lei, S. Wu, Y. Yang, J. Xue, Q. Zhang, Maritime search and rescue leveraging heterogeneous units: A multi-agent reinforcement learning approach, in: 2023 IEEE/CIC International Conference on Communications in China (ICCC), 2023, pp. 1–6, https://doi.org/10.1109/ICCC57788.2023.10233604.
[26] V. Sadhu, C. Sun, A. Karimian, R. Tron, D. Pompili, Aerial-deepsearch: Distributed multi-agent deep reinforcement learning for search missions, 2020 IEEE 17th Int. Conf. on Mob. Ad Hoc and Sens. Syst. (MASS) (2020) 165–173, https://doi.org/10.1109/MASS50613.2020.00030.
[27] B. Fraser, A. Perrusquía, D. Panagiotakopoulos, W. Guo, A deep mixture of experts network for drone trajectory intent classification and prediction using non-cooperative radar data, in: 2023 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2023, pp. 1–6.
[28] A. Perrusquía, W. Guo, Trajectory inference of unknown linear systems based on partial states measurements, IEEE Trans. Syst. Man Cybern. Syst. (2024).

[29] Z. Yang, L. Nguyen, J. Zhu, Z. Pan, J. Li, F. Jin, Coordinating disaster emergency response with heuristic reinforcement learning, in: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2020, pp. 565–572, https://doi.org/10.1109/ASONAM49781.2020.9381416.
[30] T. Kacker, A. Perrusquia, W. Guo, Multi-spectral fusion using generative adversarial networks for uav detection of wild fires, in: 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), IEEE, 2023, pp. 182–187.
[31] R.N. Haksar, M. Schwager, Distributed deep reinforcement learning for fighting forest fires with a network of aerial robots, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1067–1074, https://doi.org/10.1109/IROS.2018.8593539.
[32] D.M. Doolin, N. Sitar, Wireless sensors for wildfire monitoring, in: M. Tomizuka (Ed), Smart Structures and Materials 2005: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, vol. 5765, International Society for Optics and Photonics, SPIE, 2005, pp. 477–484.
[33] Z. Wang, W. Gong, Y. feng Jiao, Q. lan Wu, W. yan Li, Fully convolutional network for recognition of small buildings in aerial images, 2018 Fifth International Workshop on Earth Observation and Remote Sensing Applications (EORSA) (2018) 1–5, https://doi.org/10.1109/EORSA.2018.8598558.
[34] V. Slyusar, M. Protsenko, A. Chernukha, V. Melkin, O. Biloborodov, M. Samoilenko, O. Kravchenko, H. Kalynychenko, A. Rohovyi, M. Soloshchuk, Improving the model of object detection on aerial photographs and video in unmanned aerial systems, East.-Eur. J. Enterp. Technol. (2022) https://doi.org/10.15587/1729-4061.2022.252876.
[35] G.A. Dorrer, S.V. Yarovoy, Description of wildfires spreading and extinguishing with the aid of agent-based models, IOP Conf. Ser.: Mater. Sci. Eng. 822 (1) (2020) 012010, https://doi.org/10.1088/1757-899X/822/1/012010.
[36] A. Somanath, S. Karaman, K. Youcef-Toumi, Controlling stochastic growth processes on lattices: Wildfire management with robotic fire extinguishers, in: 53rd IEEE Conference on Decision and Control, 2014, pp. 1432–1437, https://doi.org/10.1109/CDC.2014.7039602.
[37] J. Mugabe, M. Wisniewski, A. Perrusquía, W. Guo, Enhancing situational awareness of helicopter pilots in unmanned aerial vehicle-congested environments using an airborne visual artificial intelligence approach, Sensors 24 (23) (2024) 7762.
[38] A. Barbosa, C.A. Moura, J.P. de Negreiros, J. Aguiar, Numerical treatment of a data completion problem in heat conduction modelling, J. Univers. Comput. Sci. 26 (2020) 1177–1188, https://doi.org/10.3897/jucs.2020.061.
[39] R. Azzam, I. Boiko, Y. Zweiri, Swarm cooperative navigation using centralized training and decentralized execution, Drones 7 (3) (2023).
[40] T. Rashid, M. Samvelyan, C.S. De Witt, G. Farquhar, J. Foerster, S. Whiteson, Monotonic value function factorisation for deep multi-agent reinforcement learning, J. Mach. Learn. Res. 21 (178) (2020) 1–51.
[41] D. Panagopoulos, A. Perrusquia, W. Guo, Selective exploration and information gathering in search and rescue using hierarchical learning guided by natural language input, in: 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2024, pp. 1175–1180.
[42] K. Tamanakijprasart, A. Perrusquia, S. Mondal, A. Tsourdos, Autonomous path selection of unmanned aerial vehicle in dynamic environment using reinforcement learning, in: AIAA SCITECH 2025 Forum, 2025, pp. 1998.
[43] M. Sardar, D. Das, S. Mukhopadhyaya, Grid exploration by a swarm of autonomous robots with minimum repetitions, Theor. Comput. Sci. 933 (2022) 67–87.
[44] P. Panaite, A. Pelc, Exploring unknown undirected graphs, J. Algorithms 33 (2) (1999) 281–295.
[45] A. Dessmark, A. Pelc, Optimal graph exploration without good maps, Theor. Comput. Sci. 326 (1) (2004) 343–362.

Author biography



Maxime Collignon received the M.Sc. in Applied Artificial Intelligence at Cranfield University in 2024. His main research interests focused on reinforcement learning, deep learning, geopolitics, and machine learning techniques applied in complex cyberphysical systems.



Adolfo Perrusquía received the M.Sc. and Ph.D. degrees, both in Automatic Control from the Automatic Control Department at the CINVESTAV-IPN in 2016 and 2020, respectively. He is currently a lecturer at Cranfield University, and a RAEng alumni. He is a member of the IEEE Computational Intelligence Society and Associate Editor of the IEEE TNNLS. He is member of the Early Career Advisory Board of the Control Engineering Practice. His main research of interest focuses on reinforcement learning, inverse reinforcement learning, data-driven control nonlinear control, machine learning and system identification.



Antonios Tsourdos obtained an MEng in electronic, control and systems engineering from the University of Sheffield (1995), an MSc in systems engineering from Cardiff University (1996), and a PhD in nonlinear robust missile autopilot design and analysis from Cranfield University (1999). He is a Professor of Control Engineering at Cranfield University and was appointed Head of the Centre for Cyber-Physical Systems in 2013. Prof. Tsourdos was a member of Team Stellar, the winning team for the UK MoD Grand Challenge (2008) and the IET Innovation Award (Category Team, 2009). He was an Associate Editor of IEEE Transactions on Aerospace and Electronic Systems, Proceedings of the IMechE Part G: Journal of Aerospace Engineering, Aerospace Science & Technology, and Internati onal Journal of Systems Science.



Weisi Guo received the M.Eng. degree in engineering, and the M.A. and Ph.D. degrees in computer science from the University of Cambridge, Cambridge, U.K., in 2005, 2011, and 2011, respectively. He is a Chair Professor of Human-Machine Intelligence at Cranfield University, Cranfield, U.K. He has published over 180 papers and is the principal investigator on several molecular communication research grants. Prof. Guo's research has won him several international awards. He was a Turing Fellow at the Alan Turing Institute.



https://dspace.org/

Search and rescue operations in wildfires using unmanned aerial vehicles: a multi-agent deep reinforcement learning approach

Collignon, Maxime; Perrusquía, Adolfo; Tsourdos, Antonios; Guo, Weisi

2025-11-07
Elsevier
https://doi.org/10.1016/j.neucom.2025.131211
Collignon M, Perrusquía A, Tsourdos A, Guo W. (2025) Search and rescue operations in wildfires using unmanned aerial vehicles: a multi-agent deep reinforcement learning approach. Neurocomputing, Volume 653, November 2025, Article number 131211

Downloaded from a DSpace repository. https://dspace.org/