Swarm Intelligence (2023) 17:89–115
https://doi.org/10.1007/s11721-022-00218-9

Wildfire detection in large-scale environments using force-based control for swarms of UAVs

Georgios Tzoumas[1,2] · Lenka Pitonakova[3] · Lucio Salinas[1,2] · Charles Scales[3] · Thomas Richardson[1,2,4] · Sabine Hauert[1,2]

Received: 4 January 2022 / Accepted: 15 September 2022 / Published online: 1 November 2022

Abstract

Wildfires affect countries worldwide as global warming increases the probability of their appearance. Monitoring vast areas of forests can be challenging due to the lack of resources and information. Additionally, early detection of wildfires can be beneficial for their mitigation. To this end, we explore in simulation the use of swarms of uncrewed aerial vehicles (UAVs) with long autonomy that can cover large areas the size of California to detect early stage wildfires. Four decentralised control algorithms are tested: (1) random walking, (2) dispersion, (3) pheromone avoidance and (4) dynamic space partition. The first three adaptations are known from literature, whereas the last one is newly developed. The algorithms are tested with swarms of different sizes to test the spatial coverage of the system in 24 h of simulation time. Best results are achieved using a version of the dynamic space partition algorithm (DSP) which can detect 82% of the fires using only 20 UAVs. When the swarm consists of 40 or more aircraft 100% coverage can also be achieved. Further tests of DSP show robustness when agents fail and when new fires are generated in the area.

Keywords Swarms · UAVs · Monitoring · Physicomimetics · Wildfires · Dynamic space partition

1 Introduction

Wildfires affect human activity in social, economic, and environmental (Tedim et al., 2018) aspects. Over the last 10 years, the world has witnessed catastrophic wildfires that destroyed vast areas of forests and properties. Wildfires killed billions of animals and caused the loss of human lives (Gill et al., 2013; Molina-Terrén et al., 2019; Legge et al., 2021). As the effects of climate change have started to create more favourable conditions for wildfires, novel methods will need to be developed to assist in their early detection (Bowman et al., 2020). Detecting wildfires at an early stage is crucial for their successful

✉ Georgios Tzoumas
georgios.tzoumas@bristol.ac.uk

✉ Sabine Hauert
sabine.hauert@bristol.ac.uk

Extended author information available on the last page of the article

suppression (Barmpoutis et al., 2020). However, patrolling for wildfires and detecting them early can be a difficult task for firefighters due to the size of the areas that need to be monitored. Some methods of detection of wildfires involve satellite imaging and aerial patrolling with the use of manned aircraft or fire outposts (Pradhan et al., 2007). These methods have advantages such as spatial resolution and early mitigation of identified fires but they also have disadvantages. Manned operations are costly, often dangerous and cannot operate continuously or during night time. Satellite imagery does not have adequate temporal resolution and it is also expensive (Alkhatib, 2014). This creates a window of opportunity for uncrewed aerial vehicles (UAVs) to be used as they can operate for long periods of time and they can be more cost-effective.

UAVs are often used in firefighting operations. They are controlled by trained firefighters who use them to receive information about wildfires during many stages of their missions (Roldán-Gómez et al., 2021; Barmpoutis et al., 2020; Ollero and Merino, 2006). In literature, a quad-copter was also developed to actively extinguish fire spots (Aydin et al., 2019). Firefighters around the world are in need of automated tools but as robotics engineers are not always engaged with the procedures of firefighting operations, the developed systems are not often suitable to face real-life scenarios. Also, it is not easy to define the precise requirements of a firefighting system as there are many variables that affect the methods that firefighters use to identify and to extinguish a wildfire (Gazzard et al., 2016). Furthermore, large areas need to be patrolled which can not be accomplished by a single UAV as it cannot provide coverage in an adequate time frame. Therefore, swarms of UAVs that can communicate with one another seem to be a powerful tool to provide coverage over larger areas autonomously. Swarms are very useful when it comes to operations in harsh environments. As wildfires can be very unpredictable, it is possible that robotic mechanisms fail. Using decentralised algorithms on swarms of UAVs can instead lead to more robust systems (Ghamry et al., 2017).

In our work, we propose a swarm system consisting of high payload UAVs to monitor large areas. Four algorithms are tested, three that are known from literature and a newly developed algorithm called dynamic space partition (DSP). Similar to the classical centralised approach that partitions a search space based on a number of robots (Leonard and Feder, 2000), we aimed to create an algorithm that is able to operate alike, but is distributed and able to react to changes in numbers of robots. This algorithm is inspired by lattices of solid elements as seen in physicomimetics by Spears et al.,. (Spears and Spears, 2012). The novelty of the DSP algorithm lies in the ability to partition a space in a distributed manner autonomously. The distributed nature allows swarms to cope with dynamic environments and swarms of different sizes. The loss of an agent and its associated DSP point will trigger a reconfiguration of neighbouring DSP points without any additional steps. The paper is organised as follows: Section 2 describes related work. Section 3 describes the simulation environment and provides information about the development of our algorithm. Section 4 presents results for every algorithm, and a comparison between their performance is discussed. In Sect. 5, a discussion regarding our results is presented. In Sect. 6, our conclusions are shown, and finally in 7, we discuss our plans for future work.

2 Related work

2.1 Firefighters and UAVs

Our previous work focused on understanding what a swarm of high payload UAVs should do from the user's perspective. Out of our discussions with firefighters from around the world (USA, UK, Australia, Greece and Portugal), we found that they are interested in a system that would patrol an area and provide feedback if a fire is identified to a swarm commander or a swarm operator. Ideally, this system would be able to interact and communicate effectively with them. Additionally, firefighters mentioned that the UAVs that are used are usually off-the-shelf UAVs that have limited payload and battery capacity. This limits their operational capability to a few minutes, requiring firefighters to change batteries and re-deploy the aircraft. In this application though, long operational hours are needed as firefighters can face wildfires for many hours or even days (Pausas and Keeley, 2021).

Larger aircraft have potential as they can fly for longer periods of time and they can withstand high wind speeds that are often seen in wildfire scenarios. These requirements can be satisfied by medium altitude long endurance UAVs such as the ULTRA platform. This UAV is developed by Windracers Ltd., and it can travel for 1000 km carrying 100 kg of payload. Additionally, it is able to withstand wind speeds up to 56 km/h–70 km/h, thus making it a useful aircraft for firefighting applications (Oakey et al., 2021). UAVs can be used to monitor large areas in various applications from environmental monitoring, urban search and rescue to anti-poaching operations (Carpin et al., 2013; Koh and Wich, 2012; Penny et al., 2019). Researchers have also been investigating the use of teams of UAVs in these operations (Atten et al., 2016; Basilico and Carpin, 2015). Additionally, swarms of robots are used as exploration tools to perform area coverage and to monitor effectively an area (Ghamry et al., 2017; Stolfi et al., 2020). We can conclude that UAVs are becoming a reality in the aforementioned operations.

2.2 Monitoring fire fronts with UAVs

Searching for wildfires can be treated as a problem of searching and tracking. Fires can be considered as targets that are located in an unknown environment which needs to be explored and monitored. Self-organising agents that can monitor the propagation of a fire front have been studied to serve as an autonomous monitoring tool (Sherstjuk et al., 2018; Kumar et al., 2011). The work of Yang et al.,. has shown that UAVs using particle swarm optimisation algorithm, can locate and monitor the development of a fire front (Yang et al., 2021). The work of Inoccente et al.,. has shown a self-organising system of UAVs that can detect and extinguish wildfires. Their algorithm is a version of particle swarm optimisation with a randomness factor incorporated in the trajectory generation algorithm. Their simulations are focused on a scenario where a targeted area of 100 × 100 m is explored (Innocente and Grasso 2019). However, restricting the environment to such a small area is a strong assumption as firefighters are called to monitor and effectively suppress larger fires. The work of Atten et al. shows the use of a swarm of UAVs using pheromones to track and monitor potential areas of interest. They use a discretised world where agents deposit repulsive pheromones in areas that have already been explored and attractive pheromones when a target has been seen to attract other UAVs (Atten et al., 2016).

In literature, monitoring fire fronts has been investigated using various techniques. The work of Seraj and Gombolay demonstrates a coordinated control system for UAVs to monitor a propagating fire front while taking into account the locations of the firefighters to inform them about the propagation of the fire front (Seraj et al., 2019). They used an adaptive extended Kalman filter (AEKF) to create goal positions for their UAVs depending on the location of fire fronts and firefighters. The algorithm was demonstrated in reality in a mock scenario (Seraj and Gombolay, 2020). Monitoring wildfires has also been investigated in the creation of path planning trajectories for UAVs. The work of Ruiz et al.,. presents a monitoring control system using a variable neighbourhood search (VNS). Using this method it is possible to update the planned trajectory when potential improvements or once new information is provided such as updates to the locations of the fire fronts (Bailon-Ruiz et al., 2018). A distributed control framework for a team of UAVs is also shown by Pham et al. monitoring a fire front and maintaining spatial distance from each other to maximise the coverage of the fire front. The UAVs monitor the fire front with sensory equipment and this is fed back to their control system to adjust their positions in relation to each other and to the fire front propagation (Pham et al., 2017).

Another example of swarm intelligence which can be used in fire identification and task allocation has been shown in the work of Schwarzrock et al. They developed versions of the swarm gap algorithm, a task allocation sorting algorithm, to allocate tasks to each individual agent based on their resources (Schwarzrock et al., 2018). The work of Leonard et al. demonstrates the use of swarm monitoring and fire detection. An A* algorithm is used to define the shortest possible path from the UAVs to a desired location. The algorithm allows the definition of paths that the UAVs will need to follow in order to explore a given area (Leonard et al., 2012).

We see in previous work that multi-UAV teams and swarms of UAVs can be used to monitor propagating fire fronts. Their control systems work based on feedback that they get from the environment. This can be the locations of firefighters, the fire front or other UAVs in the environment. In the following section, other methods are explored.

2.3 Deep learning in wildfire monitoring

Deep learning and reinforcement learning methods have been used to monitor wildfires. Haksar and Schwager have used a heuristic approach where UAVs change their position based on the location of burnt areas. They compare this work with a multi agent deep Q network (MAQDN). They show that the MAQDN was able to scale better than the heuristic approach using 10 agents and 16 fire positions (Haksar and Schwager, 2018). The work of Viseras et al. has shown the use of multiple single trained Q-learning agents (MSTA) and value decomposition networks (VDN). They use a stochastic fire model which can also be wind driven. Their control framework shows that it is possible to use these methods to control a group of 3 and a group of 9 UAVs to monitor a fire front (Viseras et al., 2021).

This work can be used in smaller environments where a developing fire front is monitored. This can be very useful and can provide support to firefighters and use modern methods to optimise how UAVs can autonomously provide information to firefighters. They are limited though in creating solutions for post-event firefighting incidents. This means that they tackle a problem after a fire incident has occurred. Furthermore, it may be difficult to use these techniques with larger swarms as only 1–10 UAVs were used in these scenarios (Viseras et al., 2021; Haksar and Schwager, 2018). In previous work with firefighters, it was seen that they are very interested in patrolling large areas with UAVs to capture potential fires at an early stage and thus mitigate them more effectively. Thus, different area coverage and monitoring algorithms are investigated in the following section.

2.4 Area partitioning

Voronoi tessellation has been used as a strategy to separate an area and to allocate robots to these area subsets (Cortes et al., 2004; Cortés et al., 2005). The work of Adepegba et al. presents a control framework for multi-agent systems to cover an area. They use a Voronoi tessellation method to partition the search space in sub-areas. Additionally, they use a reinforcement learning actor-critic technique to optimise the control inputs of acceleration to move the robots in their desired positions (Adepegba et al., 2016). Voronoi tessellation for robotic organisation in an area has also been seen in the work of Alexandrov et al. where they experimented with various world sizes, start positions and number of robots in their system. Additionally, they have proven robustness of the system in various swarm sizes and in convex shaped environments (Alexandrov et al., 2018). This work shows how the use of Voronoi tessellations can be a useful tool for separating an area in different sub areas and to allocate robots for area coverage. They do not present how their robotic team monitors the area subsets. Furthermore, the authors explain that their algorithm increases its run time as the size of the robotic team increases. Additionally, it is not shown how these systems are able to cope with potential changes in environments. This can be used as an initial step to partition an area but requires re-calculation of the partition areas whenever a change is seen in the environment after their robots have been deployed.

The work of Spears et al. shows how robots can maintain formations similar to structures that are seen in elements in nature. They are able to show how these formations can be maintained as the robots move through obstacles to search an area. This is achieved with forces that are generated between their robots. The formations can be rectangular, tetrahedral or triangular (Spears and Spears, 2012; Spears et al., 2004). Inspired by these approaches we create the dynamic space partition algorithm where it is possible to control a swarm of UAVs to partition the environment, and explore the partitions for potential fires. Additionally, we show how failing agents do not compromise the functionality of the system.

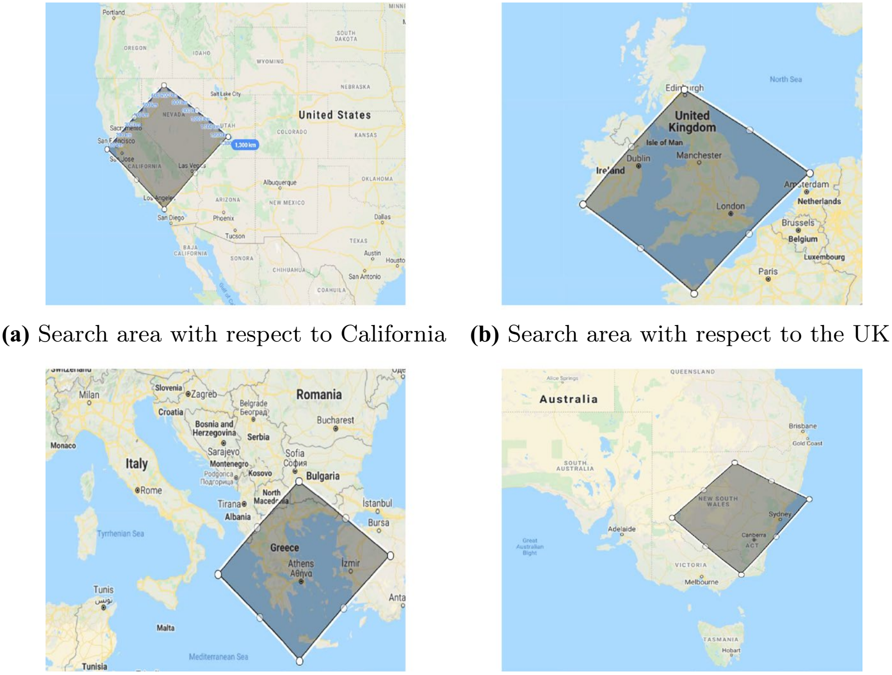
3 Methodology

3.1 The case of California as a simulation scenario

The proposed scenario was inspired from the wildfires that take place in California. The dry climate of the region in combination with the strong winds that are developed from ocean currents provide favourable conditions for the initiation of wildfires. The state of California has an area of 423,970 km^2 with 133,546 km^2 of forest land (Schoenherr, 2017). This is almost a third of the total area of the state. This vast forest area requires frequent monitoring to take place in a day. Therefore, the simulation scenario that is considered is to explore an area as large as the whole state of California with fire areas that are located at random positions in the map. These fire areas, or areas of interest, are required to be identified by the swarm of UAVs (Fig. 1).

The scenario chosen also generalises to other fire-prone areas. Southern Mediterranean countries such as Greece, Spain and Portugal suffer every year from wildfires. Australia, Brazil and Siberia have experienced mega-fires as well. Environmental conditions of long droughts and high temperatures are common in every fire season around the world (Gill

Fig. 1 Windracers ULTRA platform on a runway (Steffen, 2020)



(a) Search area with respect to California **(b)** Search area with respect to the UK

(c) Search area with respect to Greece **(d)** Search area with respect to Australia

Fig. 2 Area of 651 km × 651 km with respect to different locations of the world

et al., 2013; Bowman et al., 2020). Figure 2 shows that an area similar to our scenario here would also be useful in these contexts.

3.2 Simulation environment

We built a custom simulator using Python 3.8. The program creates a 2D environment which allows a user to create a world, a swarm of ULTRAs and wildfires. The world is assumed to be a square of 651.15 km × 651.15 km corresponding to the total area of California. The motion of the UAVs is simulated using a simple kinematics model that captures the dynamics of the existing ULTRA UAV platform. A base where all the UAVs are deployed from is assumed to be located at the central location of the area. The simulations run for 24 h to represent a day in the operation of the swarm system. The size of the swarm is then altered and its performance is assessed based on the number of fires that the swarm could identify. A summary of all the simulations parameters is given in Table 1.

The environment that the agents are called to explore is static. We also investigate the performance of our system after new fires have appeared. Thus, fires appear at random times and locations. The fires are considered mapped once the UAVs move within their fire sensor range.

The performance metric used asses how many of the existent fires were identified by our swarm:

$$\text{performance} = \frac{\text{fires identified by the swarm}}{\text{fires existent in the world}} \tag{1}$$

3.3 UAV flight model

The UAV that was chosen to be modelled is a fixed wing UAV called the ULTRA platform. This UAV is developed by Windracers ltd. and it can travel for 1000 km carrying 100 kg of payload. It has a cruise speed of 40 m/s and a wingspan of 10 m (Oakey et al., 2021).

A kinematic model was developed that controlled the motion of the UAVs with the alteration of the heading of the agents. Planes are assumed to operate at the same altitude (Hauert et al., 2011; Innocente and Grasso, 2019; Hao et al., 2021). The kinematic model of the i$^{\text{th}}$ UAV is described using the following equations:

$$\begin{cases} \dot{x}_i = v \times \cos\theta_i \\ \dot{y}_i = v \times \sin\theta_i \\ \dot{\theta}_i = \omega_i \end{cases} \tag{2}$$

The velocity v is the cruise speed of the aircraft. The position is denoted as (x_i, y_i) and θ_i is the orientation of the i-th aircraft. Constant cruise speed was applied to every agent and to change their orientation a PID controller was implemented to ensure smooth alterations of the heading. The maximum angular speed was defined as:

$$\omega_{\max} = \frac{v}{r_{\min}} \tag{3}$$

where v is the cruise speed of the aircraft and $r_{\min}$ is the minimum turn radius of the aircraft. As we aim to catch fires at an early stage, a small sensory range of 6 km was chosen. This assumption takes into account the potential to use multispectral or visual sensors to

Table 1 Scenario parameters

World size	Simulation duration	Number of agents	Number of fires	Time step
651.15 × 651.15 km	24 h	10–50	20	0.5 s

Table 2 UAV parameters

Cruise speed	Minimum turn radius	Fire sensor range	Obstacle sensor range	Size
40 m/s	150 m	6 km	1–5 km	10 m

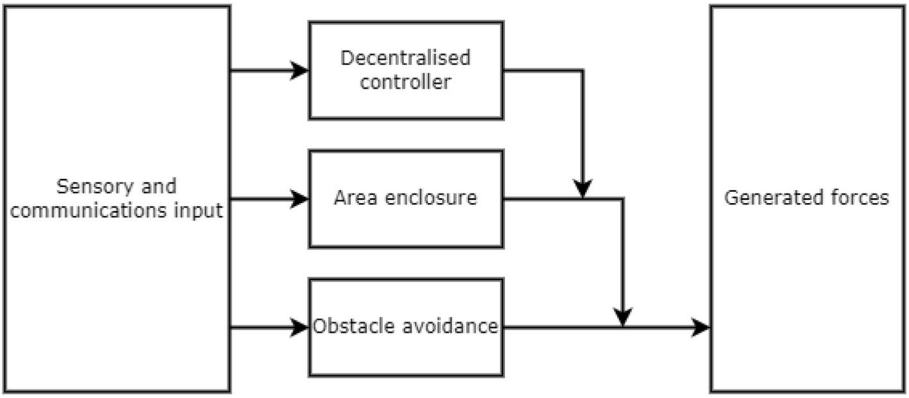


Fig. 3 Subsumption architecture of the behaviour of an agent. As a first priority agents must avoid obstacles to ensure their safety, then they will need to remain within the desired boundaries and lastly the decentralised controller takes over to explore the given area

identify a target. Depending on the size of the fire this sensory equipment can identify fires that are located tens of kilometres away (Sherstjuk et al., 2018).

An obstacle sensor range of 1–5 km was used assuming that aircraft use sensors to detect other agents, world barriers and environmental queues. Additionally, the UAVs are assumed to be equipped with 4G modules that allows them to communicate with one another at large distances to pass information. The parameters that were used to define the behaviour of the UAVs are given in Table 2.

3.4 Algorithmic development

Four swarm controllers were developed to perform fire identification. The algorithms are distributed, meaning every aircraft computes its next action-based in information received from other aircraft and sensing from the local environment. The first three algorithms have been previously seen in the literature, whereas the fourth one is a newly developed algorithm to tackle the scenario at hand (Fig. 3):

- Uniform random walking (RW)
- Random walking with dispersion (RWDP)
- Pheromone avoidance (PHA)
- Dynamic space partition (DSP)

3.4.1 Force-based control

To control the behaviour of the platforms a physicomimetics approach was taken as described by Spears (Spears et al., 2004; Spears and Spears, 2012). We apply forces to the UAVs to control their motion. This avoids complex trajectory computations and allows smooth manoeuvres of the aircraft. By applying external forces on the platform different behaviours can be achieved. Applying forces that are random based on a distribution can result in a random walking behaviour. Repulsion from other agents, borders or environmental cues (pheromones) can be performed by applying repulsive forces to the agents. Forces are applied on the aircraft and they are analysed at the x and y directions. A resultant desired angle as a result of these forces is calculated using:

$$\text{desired angle} = \arctan \frac{F_y}{F_x} \tag{4}$$

The desired angle is taken from the effect of the external forces and the current heading is corrected to reach the desired value using a PID controller. That is how manoeuvres are achieved ensuring realistic flight behaviour. The magnitude of the applied forces do not affect the turns but only the new direction that the aircraft need to take.

3.4.2 Baseline behaviours

Every search algorithm has two basic behaviours that are implemented. These are: obstacle avoidance and area enclosure. Agents are repulsed from other agents that are within their obstacle sensor range. If two agents are in close proximity, they generate a repulsive force which grows larger as their relative distance becomes smaller. The resultant force is combined with the current velocity vector of the aircraft resulting in a new global heading for the agent. This can be seen in Fig. 4 and it is described in Algorithm 1. The same logic is performed to keep agents enclosed in an area. When an agent approaches a boundary a repulsive force is applied from the point of contact of the obstacle sensor range with the boundary. This directs agents away from boundaries. This can be seen in Algorithm 2 (Figs. 4, 5).

Algorithm 1 Obstacle avoidance controller

```
for Obstacles that are within obstacle sensor range do
    Calculate repulsive force from obstacle
    Apply repulsive force to current velocity vector
end for
```

Algorithm 2 Area enclosure controller

```
if Agent senses barrier then
    Calculate repulsive force from the nearest barrier point
    Apply repulsive force to current velocity vector
end if
```

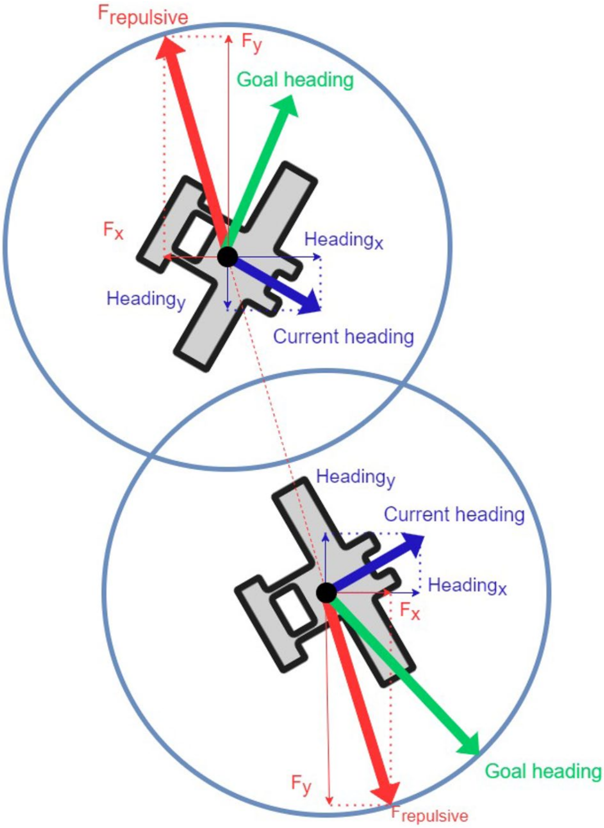


Fig. 4 Repulsion force between agents. A demonstration of the forces that are applied on the UAVs when pheromones, world boundaries or other agents are seen. The current heading of the aircraft are shown in blue. The repulsive forces that are generated due to the sensed object are shown in red. The resulting goal heading is shown in green (Color figure online)

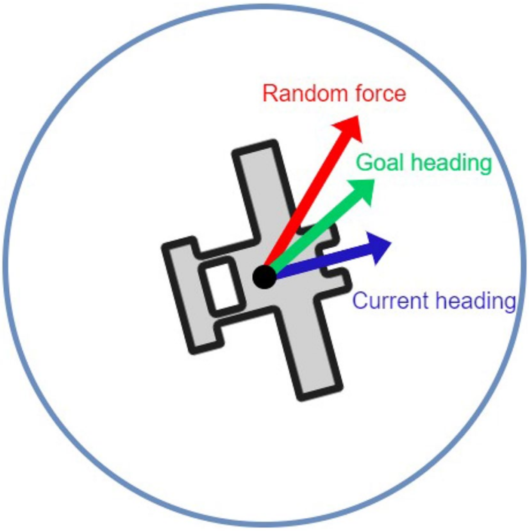


Fig. 5 Random walking behaviour. The current heading of the aircraft is shown in blue. The controller creates a random force that is shown in red. The resulting heading is shown in green (Color figure online)

3.4.3 Uniform random walking

Random walking is a common technique when it comes to area exploration (Yang, 2014). It is used as the baseline for a simple decentralised algorithm. There can be various methods to perform random walking using different distributions for the forces that are generated in the agent's controller. We used a uniform distribution using the *random* python package generating values between − 1 and 1 as magnitudes of forces on the x and y direction. We then add these values to the applied forces on the x and y direction for each agent to generate a new desired heading vector. That vector is then combined with the current heading of the aircraft. A new force in the x direction is generated using Eq. 5:

$$\text{Random force}_x = a \tag{5}$$

where

$$a \in [\text{ -1, 1}]$$

The same process takes place for the y direction. A new random force is generated at random intervals between 0 and 10 s. During the tests of this algorithm, the obstacle sensor range was set to 1 km. This was performed as this algorithm would be used as our baseline algorithm. This is described in Algorithm 3.

Algorithm 3 Uniform random walk controller

```
if Random force generator timer is reached then
    Apply force in random direction
    Regenerate random timer
end if
```

3.4.4 Random walking with dispersion

Using only random walk without a large obstacle sensor range is not expected to allow the swarm to disperse adequately. Indeed, the agents only avoided other agents if they were within the range of their obstacle sensory equipment (1 km). Using a larger value for the obstacle sensor range instead creates dispersing behaviours at a larger range. This allows the UAVs to disperse faster thus improving their performance in finding more fires. For the test runs an obstacle sensor range of 5 km was used, as we assume that agents would be able to detect one another using different sensory equipment. This is shown in Algorithm 4.

Algorithm 4 Random walking with dispersion

```
if Agent senses other agent within obstacle sensor range then
    Calculate repulsive force from other agent
    Apply repulsive force to current velocity vector
end if
if Random force generator timer is reached then
    Apply force in random direction
    Regenerate random timer
end if
```

3.4.5 Pheromone avoidance

Many animals deposit information in their environment with the release of pheromones. They do this to mark their territories or to communicate with other animals (Hunt et al., 2019; Atten et al., 2016). Inspired by this mechanism, the pheromone avoidance algorithm creates a trail of previous locations of UAVs. These previous locations create a trajectory of historic positions that act as repulsive beacons. The information that is sent to different agents to perform pheromone avoidance are lists of pheromones of other agents. Each agent notifies other agents in the swarm when it deposits a pheromone. Thus, every agent has a record of the pheromones deposited that it can then react to. UAVs avoid the pheromones of other agents if a pheromone is located within their obstacle sensor range. For the tests of this algorithm the obstacle sensor range was set at 5 km. At the same time, the swarm is required to keep monitoring already visited areas in case a fire appears there. Therefore, the evaporation of deposited pheromones plays an important role. As time goes by pheromones lose their strength and eventually cease to exist. This allows UAVs to revisit areas that have been explored after the pheromone trails of other agents have been evaporated. Additionally, it is possible to have congested areas where UAVs have deposited a plethora of information and thus trapped other agents. These areas will be cleared out of pheromones as they evaporate. In this algorithm, the agents are designed to perform a random walk, as it was seen in Algorithm 3, while depositing pheromones at their previous visited locations. The controller requires timers to be initialised, so that they will perform specific actions when the timers are reached. A pheromone deposition timer is initialised so that the agents will deposit a pheromone at a specific rate. Additionally, an evaporation time is needed so that the pheromones can be removed after the timer is reached. A description of this controller can be seen in Algorithm 5.

Algorithm 5 Pheromone avoidance controller

```
Perform random walk as described in algorithm 3
Deposit pheromone at a given deposition rate
Send latest pheromone location to other agents
Receive pheromone locations from other agents
Save pheromone locations on pheromone list
Remove pheromone at a given evaporation rate
for Pheromone in pheromone list do
    if Pheromone is within obstacle sensor range then
        Calculate repulsive force from pheromone
        Apply repulsive force to current velocity vector
    end if
end for
```

3.4.6 Dynamic space partition

In this algorithm, each agent creates its own virtual DSP point which determines the sub area it is meant to explore. DSP points react to other DSP points through attraction and repulsive forces defined as *gravitational DSP forces* whose magnitude is defined in Eq. 6. This allows the virtual DSP points to spread out over the area. Agents are attracted to their DSP point, and upon reaching it they start exploring the area around it. This approach aims to create a robust method of separating an existent space in roughly equal sized areas, without the requirement for central calculations. When the DSP points are moved further away from other points an attractive force is generated to ensure that an equilibrium is achieved result in a distribution of DSP points over an area. This distance R is defined as the partition distance shown in Eq. 9. When the distance is larger than this value the forces change from being repulsive to attractive. The forces create hexagonal lattice structures, based on a specified gravitational constant. Densest packing of a 2-D space using a hexagonal lattice was chosen as a method for distributing dynamic space partition points in an area. The highest-density lattice packing of circles is the hexagonal packing arrangement, in which the centres of the DSP points are arranged in a hexagonal lattice (Goldberg, 1971; Spears and Spears, 2012). Other structures can also be created by changing the force equation to achieve different equilibrium positions. The magnitude of the *gravitational DSP force* is calculated by a gravitational force equation (Figs. 6, 7):

$$\|\text{gravitational DSP force}\| = \frac{G_{\text{constant}}}{d^p} \tag{6}$$

where d is the distance between two DSP points. The entity p is defined as a selected raised power by Spears in their book of physicomimetics (Spears and Spears, 2012) and its effect is shown in Fig. 8. To calculate these forces, the gravitational constant $G_{constant}$ is required. This is given by this equation as seen in the book of physicomimetics by Spears et al. (2012):

$$G_{\text{constant}} = F_{\text{max}} \times R^p \times (2 - 1.5^{1-p})^{\frac{p}{1-p}} \tag{7}$$

where F_{max} is calculated based on the maximum velocity of the DSP and its mass. The DSP is a virtual entity thus it can take any value for both its velocity and mass as it is not restrained by physical constraints. The DSP are therefore considered as particles that can be controlled with forces that are applied to them. We used for V_{max} 45 m/s which is larger than the cruise speed of the UAVs and a mass of 1 kg. dt is the time step of our simulation which is 0.5 s as it is specified in Table 1.

$$F_{\text{max}} = \frac{\text{mass} \times v_{\text{max}}}{\mathrm{d}t} \tag{8}$$

The gravitational constant requires a specific dispersion distance R to be defined. This is calculated using:

$$R = 2 \times \sqrt{\frac{\text{search area for each agent}}{\pi}} \tag{9}$$

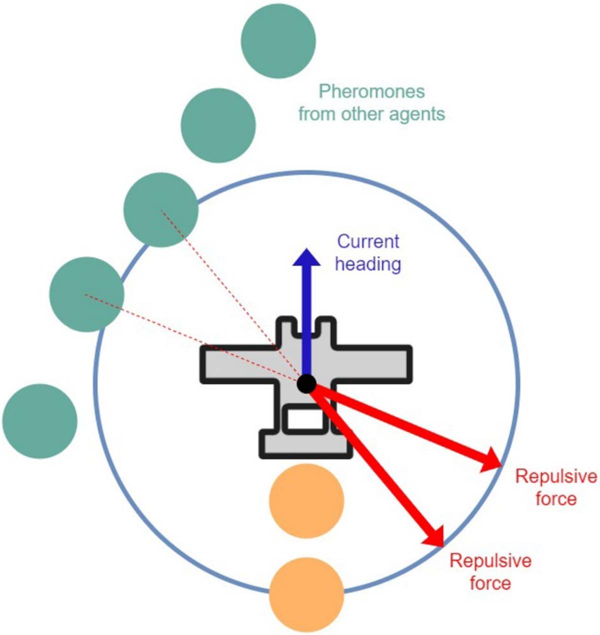
We calculate this based on the total area that needs to be explored and the number of agents in the swarm. Initially, the area of the world is calculated by multiplying the area longitude and latitude in meters. Given the density of the desired packaging, we have:

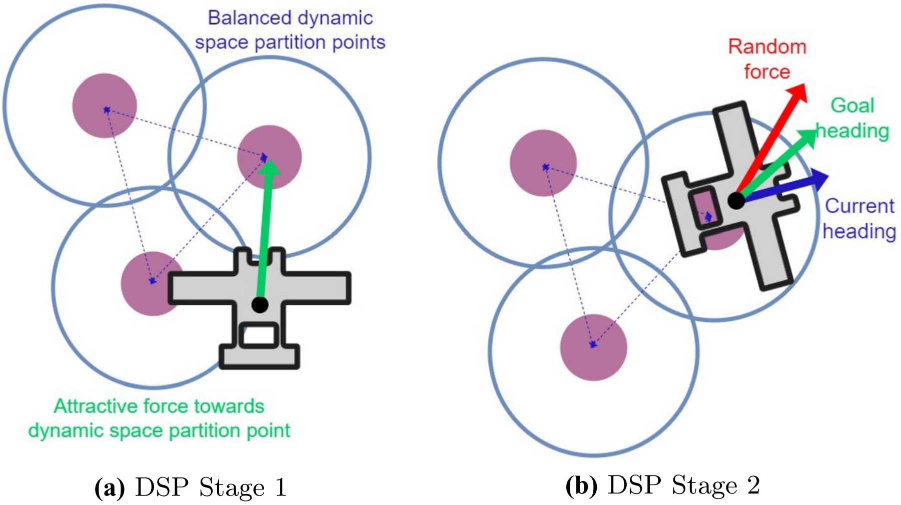
$$\text{density} = \frac{\pi \times \sqrt{3}}{6} \tag{10}$$

Multiplying this with the total area that is needed to be searched we get the maximum area that can be explored from the densest distribution:

$$\text{maximum area coverage} = \text{density} \times \text{area} \tag{11}$$

Fig. 6 Repulsion between an agent and other agents' pheromones shown in an olive colour. The beige-coloured pheromones are the agent's own pheromones. The agent ignores his own pheromones and is only repulsed from pheromones of other agents





(a) DSP Stage 1 **(b)** DSP Stage 2

Fig. 7 In the first stage **a** the virtual DSP points are repulsed from other agent's DSP points following a balance of attractive and repulsive gravitational forces. In the second stage **b** agents move to their DSP point and then perform a random walk to explore the area around it

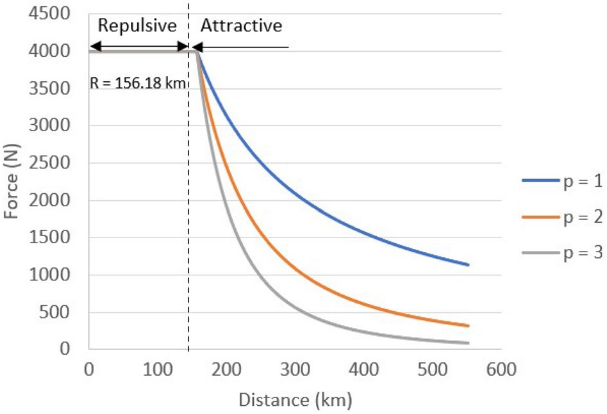


Fig. 8 Effect of p on the repulsive/attractive forces that are generated among the DSP points where R denotes the dispersion distance at 156.18 km. This is calculated with 20 agents to cover an area of 650 km by 650 km. The forces are capped in the maximum force which here is calculated as 4000 N

Thus, the required search area by each agent can be specified by:

$$\text{search area for each agent} = \frac{\text{maximum area coverage}}{\text{number of agents}} \tag{12}$$

In Fig. 8, we can find the effect of p in the applied forces on the DSP depending on the distance of two DSP points. We chose a value of $p = 2$ for our system as it allowed DSP points to reach equilibrium faster.

The direction of this force is calculated based on the location of each individual DSP point that is received by the agent. This is calculated using the following equation:

$$\text{direction} = \arctan \frac{\text{Other DSP point}_y - \text{Own DSP point}_y}{\text{Other DSP point}_x - \text{Own DSP point}_x} \tag{13}$$

The force is the analysed on the x and y direction and applied on the DSP. The ith DSP belonging to the ith UAV is described by the following kinematics equations:

$$\begin{cases} \dot{v_{ix}} = F_x \\ \dot{v_{iy}} = F_y \\ \dot{x}_i = v_{ix} \\ \dot{y}_i = v_{iy} \end{cases} \tag{14}$$

Each agent updates their respective DSP point at every time step until the system stabilises. The agents need to follow the positions of their individual DSP point as they change their location. When the locations of the DSP points are stabilised and the agents reach their respective DSP point they perform a random walk around them for a specified amount of time to explore the areas. The amount of time that each agent performs a random walk is calculated based on the largest distance that each UAV needs to cover. This is the diagonal from the central base to the edge of the desired area to explore. The time is calculated by dividing the size of the diagonal with the cruise speed of the UAV:

$$\text{time to random walk} = \frac{\sqrt{(\text{world longitude})^2 + (\text{world latitude})^2}}{2 * \text{cruise speed}} \tag{15}$$

Once their random walk operation is completed, they return to their individual DSP point. We do this so that agents can check again for potential fires in their respective area. This is important to ensure that there is not only exploration performed from our system but also monitoring, as areas that have been explored might have a new fire developed. Thus, re-visiting areas that have been explored is a feature that can be useful in firefighting. This process is described in Algorithm 6.

Algorithm 6 Dynamic space partition controller

```
for Every DSP point in DSP points received do
    Calculate gravitational force from other DSP points
    Apply gravitational force to own DSP point velocity vector
end for
if DSP point is not reached by agent then
    Calculate attractive force to own DSP point
    Apply attractive force to current velocity vector
end if
if DSP point is reached by agent then
    Perform random walk for calculated random walk time
end if
```

With the design of this algorithm we aim to develop a framework which will allow for potential changes in the environment and in the swarm. If an agent for some reason fails, or returns to base, or losses communications, the distributed DSPs will be able to re-adjust themselves to cope with this alteration. This is taking place without any centralised calculation and onboard each UAV. Each DSP re-adjusts themselves based on the current information that is available to the UAVs.

4 Results

Here, we present our results of the four different algorithms that were developed and tested. Random walk (RW), random walk with dispersion (RWDP), pheromone avoidance (PHA) and dynamic space partition (DSP). Additionally, a comparison of the performance of all algorithms is presented. Lastly, robustness tests are presented for the best performing algorithm.

4.1 Uniform random walking

The uniform random walk (RW) results show that it is an algorithm that can identify required fires, but not all, when varying the swarm size from 10 to 50. A small increase in performance can be seen as the number of agents increases which is rational as more agents can cover more space and thus identify more fires. A plateau of performance is seen as the number of UAVs increased from 30 up to 40. The best performance is achieved when 50 UAVs are used. In that case the system identified 48% of the fires. This algorithm is considered a baseline to compare the performance of the other three algorithms. An example of a test run of the algorithm is presented in Fig. 9.

4.2 Random walking with dispersion

Random walking with dispersion (RWDP) performs better for all of the different swarm sizes compared to RW. With 30 UAVs in a swarm, it is possible to identify 83% of the fires. Additionally once 50 agents are used it is possible to nearly identify all of the potential fires. This change in performance shows how important it is to disperse agents from one another at a longer range to avoid overlap in coverage.

4.3 Pheromone avoidance

The pheromone avoidance (PHA) algorithm is initially tested with a pheromone evaporation time of 2 h. This means that the pheromones would only be present for 2 h after they were generated. The algorithm needed to be tested further to define which evaporation time would be more useful as the 2 h evaporation rate is potentially removing information from the map without being processed by the agents or it could be creating a congestion of information and therefore blocking some agents from exploring the desired area. Thus, a parameter sweep of the evaporation rate of the pheromones is performed using a swarm of 30 UAVs. Results in Fig. 10 show that with 30 agents using an evaporation rate of 1 h, the system delivers the highest performance results. Additionally, an example test run of PHA can be seen in Fig. 14.

The PHA outperformed both the RW and the RWDP algorithms. As the number of agents increases, an increase in performance can be seen. With 20 agents the system is able to identify 80% of the fires. Additionally with 40 agents almost every fire is identified. As the swarm size increases to 50 agents, 100% of the fires are also identified. Thus, sharing information of historic locations can help in exploring an area more effectively.

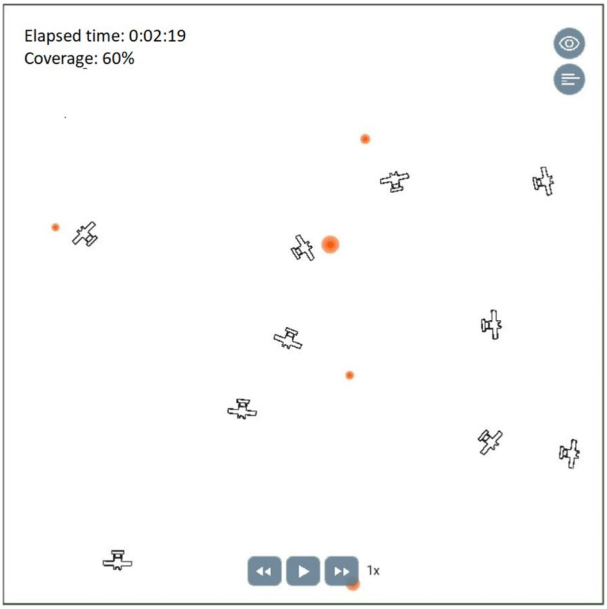


Fig. 9 Random walking algorithm in operation. The aeroplanes represent uncrewed aerial vehicles, the red circles represent fires that need to be identified. The world boundaries are represented with the black square. On the top left the simulation time and performance of the swarm are shown. The size of the world is 651.15 km × 651.15 km. The size of the aircraft and the fires is augmented for illustration purposes

4.4 Dynamic space partition

The dynamic space partition algorithm (DSP) outperforms all previous algorithms. With 20 agents the algorithm identifies 82% of the fires in the world. Additionally, with 30 agents an average of 96% of fires are identified. As the size of the swarm increases to 40 and 50 UAVs, all of the fires are identified. The results of the experimental runs of the algorithms are shown below. Additionally, an example test run of the DSP algorithm where the dynamic partition points are in equilibrium can be seen in Fig. 16.

4.5 Algorithmic comparison

After these algorithms were developed and tested they were compared in terms of performance which can be seen in Fig. 11. RW was the lowest performing algorithm without even achieving a 50% coverage in most test runs. RWDP outperformed pure RW but it was not able to achieve better performance than PHA and DSP. When the swarm size reached 30, the RWDP started performing similarly to PHA. Once 40 agents were used the two algorithms achieved the same results with an average of 96% of fires mapped. The best performing algorithm was the DSP. It outperformed all other algorithms for all of the swarm sizes. Using a swarm of 20 UAVs, it was possible to cover adequately the state of California with the DSP algorithm, by identifying 82% of the fires in the world. A swarm of 20 ULTRA platforms can be reasonable for real-world deployments over large environments (Table 3).

4.6 DSP in more challenging scenarios

To prove the effectiveness of the DSP algorithm, we tested its performance in more challenging simulation environments. A swarm of twenty agents is used to test the performance of the algorithm. This is decided as it is a cost-effective solution which can provide fire identification coverage of 82% as it can be seen in Fig. 11. One of the challenging scenarios is to test the robustness of the algorithm when a number of agents have failed. This is performed to check for graceful degradation of the swarm performance. To test this, agents

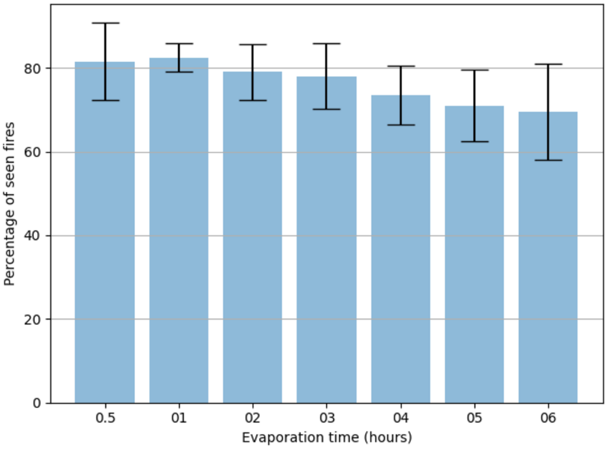


Fig. 10 Performance of a swarm of 30 agents with different pheromone evaporation rates. The error bar shows the standard deviation over 50 experimental runs. In this test, a parameter sweep is performed to identify which evaporation rate of pheromones performs best. The *x*-axis shows the values of the evaporation rate of pheromones in hours. The *y*-axis shows the percentage of fires that the system has identified. Results indicate that 1 h of evaporation time is the optimum value

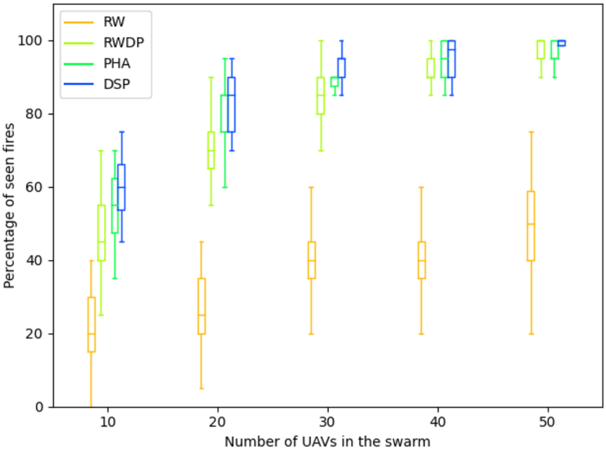


Fig. 11 Performance comparison between: Random walk (RW), random walk with dispersion (RWDP), pheromone avoidance (PHA) and dynamic space partition (DSP). Box plots show the mean of all 50 test runs and whisker plots show the higher and lower recorded values in the performance of the controllers. The swarm sizes varied from 10 to 50 in increments of 10 as it is shown on the x-axis. The percentage of identified fires is shown on the y-axis. It can be seen that DSP outperforms all other algorithms for all swarm sizes

started to fail after half of the simulation time passed. The agents did not fail simultaneously, instead agents failed at regular intervals of time. This is defined based on the number of agents that would fail in each scenario. When one agent failed, the failure occurred after half of the simulation time passed. When two agents failed, the first agent failed at 50% of the completed simulated time and the second one at 75% of the total simulation time and so on. The number of agents that failed varied from 1 to 10 agents. This corresponds to 50% of the size of the swarm. Each experiment ran for fifty times to test the performance of the system. The system has shown robustness as its performance remained unchanged when only one agent failed. As the number of failed agents increases the performance of the swarm is affected but the system is still functional. Even when 50% of the agents eventually fail the performance remains at a mean of 77%. The results of this experiment can be seen in Fig. 12. This is due to the nature of DSP which re-adjusts the DSP points in a distributed manner. If one of the agents fail then its dynamic space partition point is removed from the world. As a result the other points move to reach a new equilibrium state and to cover the area that the failed agent needed to explore. This dynamic nature of the algorithm allows the swarm system to cope with agent failures.

The other scenario was to assess if the system is able to cope with dynamic environments by generating new fires in the search area. To test this, ten fires are initially generated in the world and halfway through the total simulation time ten more fires are generated at a random location. This test is performed to examine if the swarm can identify newly generated fires at locations that have already been explored. This creates a scenario that

Table 3 Summary of results showing what percentage of fires have been identified by the swarm

Swarm size (number of UAVs)	Random walk (RW)	Random walk with dispersion (RWDP)	Pheromone avoidance (PHA)	Dynamic space partition (DSP)
10	21	43	57	58
20	28	70	81	82
30	39	83	87	91
40	41	90	93	96
50	47	95	96	98

is closer to reality as it is possible that a fire can be generated at any given moment in the world.

Results show that in this case the performance is reduced as the system is able to see 71% of the fires when 10 new fires are generated. As shown in Fig. 13, the system has a reduced performance compared to its performance in a static environment. The new target generation test shows that if new fires are generated the DSP is able to identify many of the new fires.

5 Discussion

The novelty of the DSP algorithm lies in the ability to automatically partition space in a distributed manner. The distributed nature allows swarms to cope with changing environments and swarms sizes. The loss of an agent and its associated DSP point will lead to a reorganisation of neighbouring DSP points without any additional steps (Fig. 14).

There are certain weaknesses that are identified with this methodology. The aircraft need to communicate with each other the locations of their DSP points over large distances. This requires a form of long-range communication (e.g. 4G). If we reduce the range of communications of the swarm we see that there is a decrease in their performance as shown in Fig. 15. As the communications range increase, we see that the performance is improved.

To use the DSP with decentralised communications, we would need to equip the aircraft with mesh radio modules. Typically, these modules do not exceed a communication range of 50km. Thus, to accommodate this, we would need to have a swarm of more aircraft.

Further testing in the time of life of pheromones can be performed to vary the size of our swarm and find the optimal number of UAVs and optimal evaporation time of the agents'

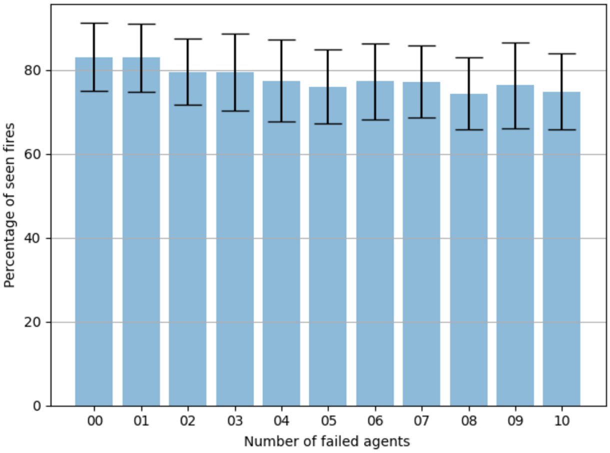


Fig. 12 Graceful degradation of the performance using DSP when individual robots fail. The error bar shows the standard deviation over 50 experimental runs. The x-axis shows the number of agents that failed. The y-axis shows the percentage of fires that the system has identified. The swarm size that was tested was 20 agents. After half of the simulation time, robots start to fail. The results show that the system is able to cope with agent failure. With the failure of one agent the performance remains the same. As more agents started to fail a decline in performance is seen but without a failure of the whole system. Even when 50% of the swarm malfunctioned 77% of the fires were seen

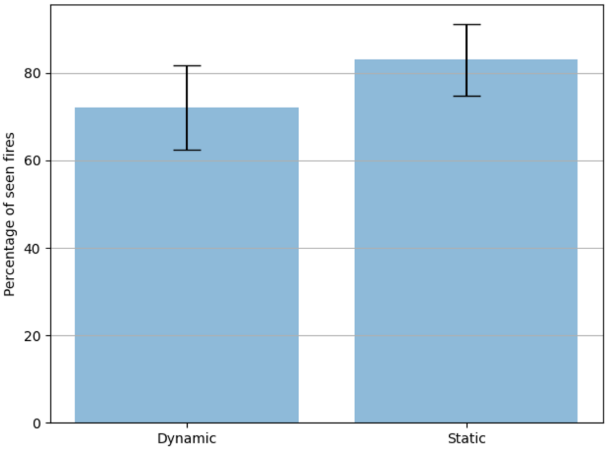


Fig. 13 Performance of DSP in a dynamic and static scenario. The error bar shows the standard deviation over 50 experimental runs. The swarm size that was tested was 20 agents. In the dynamic scenario, when half of the simulation time has passed 10 new fires are generated. The system shows a slightly reduced performance as 71% of the fires are seen when 10 new fires are generated as opposed to 82% in the static scenario

pheromones. We see in our current results that as the size of the swarm increases, the performance increases as well. Thus, there is a clear connection between the ability of more aircraft to identify more fires while using the pheromone avoidance algorithm. Interestingly, as the pheromone life increases with a swarm of 30 agents, we do not see an increase in performance. Instead, we see a decrease as when the time changes from 2 h of life to 6 h of life, the fires detected decreases from 84% to 62%. This has to do with the large amount of information that is deposited in the world. Agents are therefore trapped between areas with pheromones and as a result they are not able to explore more areas. If more than 30 UAVs were used in this scenario, then there would be even more pheromones in the world potentially trapping the agents in areas that they have already explored. If less agents are

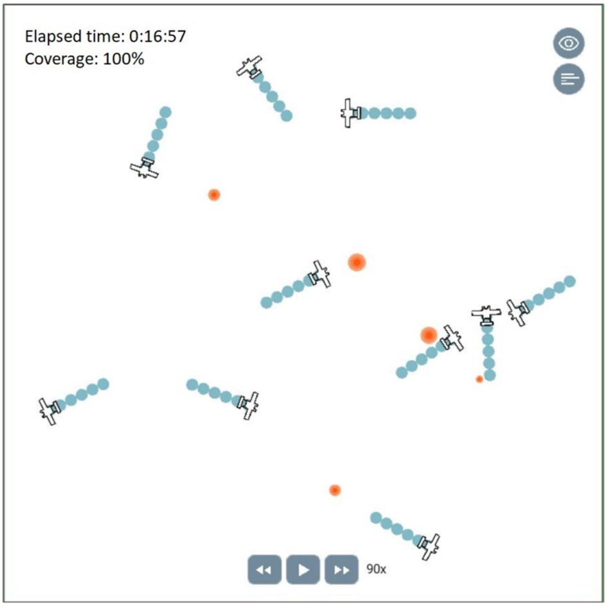


Fig. 14 Pheromone avoidance in operation showing agents depositing pheromones. Pheromones are represented with blue dots that are deposited behind the aircraft. The size of the world is 651.15 km × 651.15 km. The size of the aircraft and the fires is augmented for illustration purposes

used, then they would benefit from larger pheromone life than 2 h but up to a certain point where information would not restrict them from exploring the rest of the world.

6 Conclusion

Monitoring large areas to identify potential fires at their early stage can assist in their successful mitigation. To do so, a given area must be explored to identify potential fires. Our focus was on large-scale areas the size of California, which would require the use of UAVs with long endurance such as the ULTRA platform. Four main algorithms were developed and tested: Random walk (RW), random walk with dispersion (RWDP), pheromone avoidance (PHA) and dynamic space partition (DSP). The algorithms were compared based on their ability to identify fires while varying the size of the swarm from 10 to 50 agents. Different algorithms were also combined to identify potential behaviours that can increase the performance of the system. Results have shown that DSP was the best performing algorithm. The system was able to identify 82% of the fires with 20 UAVs. This is a promising result showing that a relatively small swarm of 20 UAVs could operate over very large-scale areas for 24-h monitoring of wild fires. Additionally, the DSP algorithm proved to be robust when a number of agents failed while in operation. Removing 50% of the agents resulted in identifying 77% of the fires. We also tested our algorithm with reduced communication capabilities showing improved performance as the range increases.

7 Future work

7.1 Simulation improvements

More work will need to be undertaken to develop a fully operational system that will help firefighters. A full mission would require a system to identify potential fires and then coordinate the UAVs to monitor and mitigate the propagation of the fire fronts. Thus, some changes will need to take place in simulation. Firstly, the environment must become dynamic so that simulations reflect the propagation of the fire front based on environmental

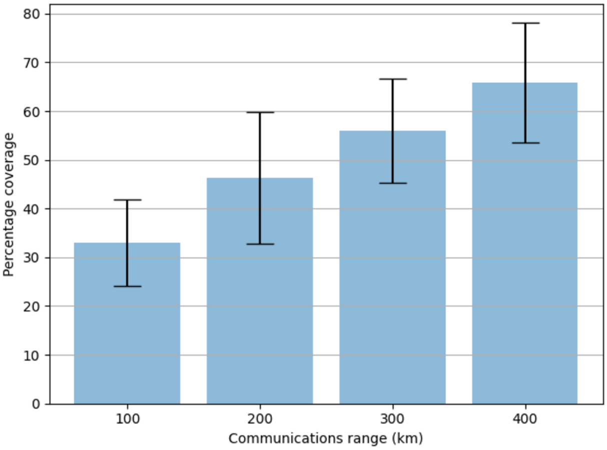


Fig. 15 Performance of DSP algorithm with 20 aircraft. It is possible to see that with reduced communication ranges the performance of the algorithm drops. As the communication ranges increase, the performance improves as well

aspects such as wind speed or topography as it is seen in the work of Innocente et al. The behaviour of the agents will need to change as well once a fire has been spotted. The agents could call nearby agents to verify that at their location there is actually a fire and to remain engaged to monitor the propagation of the fire front. In some pieces of work such as Seraj et al, this is seen where fixed winged aircraft perform an exploration task and quad-copters monitor the developing fire-front once the fire-fronts are identified. This will have an impact on the exploration of the desired areas and will affect the performance of the system which can then lead to the exploration of other metrics, such as the time that agents were engaged in mapping an identified fire or how many fires were engaged by the swarm. This can be performed by measuring the amount of time each agent is engaged on top of a fire area compared to the total time of operation. Additionally, the effectiveness of the swarm in terms of exploration can be given from the average time detecting an adequate number of fires. ULTRA UAVs can carry 100kg of payload so it is possible to develop an early extinguishing mechanism. Thus, the effect of that extinguishing material on a fire can also be studied.

7.2 Comparison to other methods

It would also be useful to perform some benchmark testing with other methodologies that are seen in literature. As we will move towards fire front monitoring it will be interesting to compare it with deep learning and reinforcements learning methods that were seen in literature and other distributed control frameworks. Great interest is in the comparison with the methods of Viseras et al. (2021) and Haksar et al. (2018). Additionally, the development of decentralised fuzzy controllers in fire identification and monitoring can be investigated (Yan et al., 2021). Furthermore, different exploration behaviours will need to be developed for the second stage of DSP. This can be achieved with search patterns such as spiral exploration or lawn mower patterns. Furthermore, other tests can be performed in the scenario with failed agents to assess the effect of random failures on the performance of the system. This should also be compared with other systems as it is seen in the work of Ramachandran et al. where the robustness of their system was tested as communications failed. This is achieved with reconfiguration of the positions of the robots to allow information sharing between them.

7.3 Communications and agents failure

Another interesting aspect is the investigation of our system's stability when agents lose communication. In the work of Seraj et al. a centralised coordinated control structure with uncertain network structure was developed where loss of communications and environmental noises were compensated (Seraj et al., 2022). Additionally, in the work of Chang et al. we see a formation controller which is also able to adjust to potential communication losses. They investigated this using a Lyapunov stability analysis where desired robot formations were maintained regardless of communication and actuator faults (Chang et al., 2018). Lastly, the work of Ramachadran et al. has shown how a monitoring task can continue given that some of the robots have malfunctioned (Ramachandran et al., 2022). Although certain elements of the aforementioned work are not applicable to our study such as the effect of communications loss on different formations, we can use similar approaches in assessing the effect of malfunctioning agents or malfunctioning communications in the

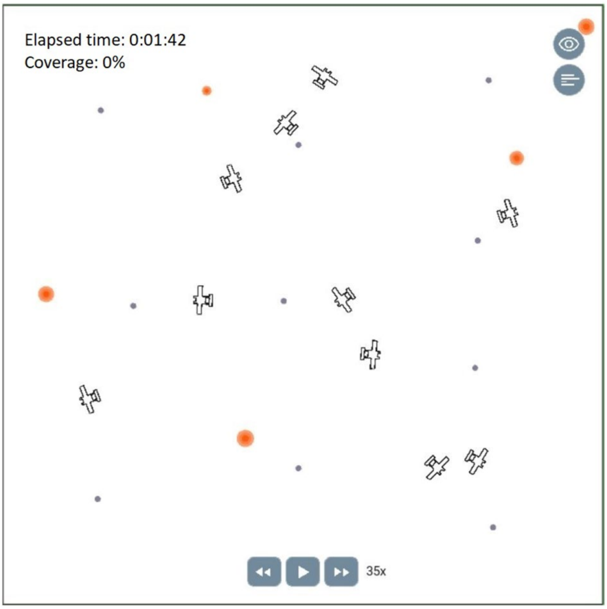


Fig. 16 Dynamic space partition algorithm in operation. The image shows the dynamic space partition points in blue in equilibrium. The aircraft is trying to reach their points or perform a random walk to explore the area. The size of the world is 651.15 km × 651.15 km. The size of the aircraft and the fires is augmented for illustration purposes

task of area exploration. Specifically, we aim to investigate the effect of communication losses in the task of area exploration to identify fires and the effect that this has when agents aim to attack a wildfire front (Fig. 16).

Acknowledgements The authors would like to thank everyone involved in the work that was presented in this paper.

Author contributions GT designed, coded and developed the framework and algorithms presented in the study. LP and LS have provided assistance with the framework and the algorithm development SH supervised the project and made substantial contributions the study and algorithms. She also contributed to this manuscript. TR provided feedback and ideas in the shape of the paper. CS provided information regarding the ULTRA platform.

Funding GT was supported by Windracaers ltd. This work was supported by Innovate UK Future Flight Challenge.

Data availability The datasets generated for this study are available on request to the corresponding author.

Declarations

Conflict of interest The study was done impartially, showing how a swarm of UAVs can explore an area with the usage of a novel algorithm named dynamic space partition.

References

Adepegba, A. A., Miah, S., & Spinello, D. (2016). Multi-agent area coverage control using reinforcement learning. In: *Florida artificial intelligence research society conference* (pp. 368–373).

Alexandrov, V., Kirik, K., & Kobrin, A. (2018). Multi-robot voronoi tessellation based area partitioning algorithm study. *Paladyn, Journal of Behavioral Robotics, 9*(1), 214–220. https://doi.org/10.1515/pjbr-2018-0014.

Alkhatib, A. A. (2014). A review on forest fire detection techniques. *International Journal of Distributed Sensor Networks, 10*(3), 597–368. https://doi.org/10.1155/2014/597368.

Atten, C., Channouf, L., & Danoy, G. et al (2016). Uav fleet mobility model with multiple pheromones for tracking moving observation targets. In *European conference on the applications of evolutionary computation* (pp. 332–347). Springer. https://doi.org/10.1007/978-3-319-31204-0_22

Aydin, B., Selvi, E., Tao, J., et al. (2019). Use of fire-extinguishing balls for a conceptual system of drone-assisted wildfire fighting. *Drones, 3*(1), 17. https://doi.org/10.3390/drones3010017.

Bailon-Ruiz, R., Lacroix, S., & Bit-Monnot, A. (2018). Planning to monitor wildfires with a fleet of uavs. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4729–4734). https://doi.org/10.1109/IROS.2018.8593859

Barmpoutis, P., Papaioannou, P., Dimitropoulos, K., et al. (2020). A review on early forest fire detection systems using optical remote sensing. *Sensors, 20*(22), 6442. https://doi.org/10.3390/s20226442.

Basilico, N., & Carpin, S. (2015). Deploying teams of heterogeneous uavs in cooperative two-level surveillance missions. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 610–615). IEEE. https://doi.org/10.1109/IROS.2015.7353435

Bowman, D., Williamson, G., Yebra, M., et al. (2020). Wildfires: Australia needs national monitoring agency. *Nature, 584*(7820), 188–191. https://doi.org/10.1038/d41586-020-02306-4.

Carpin, S., Burch, D., Basilico, N., et al. (2013). Variable resolution search with quadrotors: Theory and practice. *Journal of Field Robotics, 30*(5), 685–701. https://doi.org/10.1002/rob.21468.

Chang, Y. H., Wu, C. I., & Lin, H. W. (2018). Adaptive distributed fault-tolerant formation control for multi-robot systems under partial loss of actuator effectiveness. *International Journal of Control, Automation and Systems, 16*(5), 2114–2124. https://doi.org/10.1007/s12555-016-0587-4.

Cortés, J., Martínez, S., & Bullo, F. (2005). Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations, 11*(4), 691–719. https://doi.org/10.1051/cocv:2005024.

Cortes, J., Martinez, S., Karatas, T., et al. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation, 20*(2), 243–255. https://doi.org/10.1109/TRA.2004.824698.

Gazzard, R., McMorrow, J., & Aylen, J. (2016). Wildfire policy and management in england: An evolving response from fire and rescue services, forestry and cross-sector groups. *Philosophical Transactions of the Royal Society B: Biological Sciences, 371*(1696), 20150341. https://doi.org/10.1098/rstb.2015.0341.

Ghamry, K. A., Kamel, M. A., & Zhang, Y. (2017). Multiple uavs in forest fire fighting mission using particle swarm optimization. In *2017 International conference on unmanned aircraft systems (ICUAS)* (pp. 1404–1409). IEEE. https://doi.org/10.1109/ICUAS.2017.7991527.

Gill, A. M., Stephens, S. L., & Cary, G. J. (2013). The worldwide “wildfire’’ problem. *Ecological Applications, 23*(2), 438–454. https://doi.org/10.1890/10-2213.1.

Goldberg, M. (1971). On the densest packing of equal spheres in a cube. *Mathematics Magazine, 44*(4), 199–208. https://doi.org/10.1080/0025570X.1971.11976147.

Haksar, R. N., & Schwager, M. (2018). 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)* (pp. 1067–1074). IEEE Press. https://doi.org/10.1109/IROS.2018.8593539

Hao, C., Xiangke, W., Lincheng, S., et al. (2021). Formation flight of fixed-wing UAV swarms: A group-based hierarchical approach. *Chinese Journal of Aeronautics, 34*(2), 504–515. https://doi.org/10.1016/j.cja.2020.03.006.

Hauert, S., Leven, S., & Varga, M., et al. (2011). Reynolds flocking in reality with fixed-wing robots: Communication range versus maximum turning rate. In *2011 IEEE/RSJ international conference on intelligent robots and systems* (pp. 5015–5020). IEEE. https://doi.org/10.1109/IROS.2011.6095129

Hunt, E. R., Jones, S., & Hauert, S. (2019). Testing the limits of pheromone stigmergy in high-density robot swarms. *Royal Society Open Science, 6*(11), 190–225. https://doi.org/10.1098/rsos.190225.

Innocente, M., & Grasso, P. (2019). Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems. *Journal of Computational Science, 34,* 80–101. https://doi.org/10.1016/j.jocs.2019.04.009.

Koh, L. P., & Wich, S. A. (2012). Dawn of drone ecology: Low-cost autonomous aerial vehicles for conservation. *Tropical Conservation Science, 5*(2), 121–132. https://doi.org/10.1177/194008291200500202.

Kumar, M., Cohen, K., & HomChaudhuri, B. (2011). Cooperative control of multiple uninhabited aerial vehicles for monitoring and fighting wildfires. *Journal of Aerospace Computing, Information, and Communication, 8*(1), 1–16. https://doi.org/10.2514/1.48403.

Legge, S., Woinarski, J. C., Scheele, B. C., et al. (2021). Rapid assessment of the biodiversity impacts of the 2019–2020 Australian megafires to guide urgent management intervention and recovery and lessons for other regions. *Diversity and Distributions*. https://doi.org/10.1111/ddi.13428.

Leonard, J., Savvaris, A., & Tsourdos, A. (2012). Towards a fully autonomous swarm of unmanned aerial vehicles. In *Proceedings of 2012 UKACC international conference on control* (pp. 286–291). IEEE. https://doi.org/10.1109/CONTROL.2012.6334644

Leonard, J. J., & Feder, H. J. S. (2000). A computationally efficient method for large-scale concurrent mapping and localization. In J. M. Hollerbach & D. E. Koditschek (Eds.), *Robotics research* (pp. 169–176). Springer.

Molina-Terrén, D. M., Xanthopoulos, G., Diakakis, M., et al. (2019). Analysis of forest fire fatalities in southern Europe: Spain, Portugal, Greece and Sardinia (Italy). *International Journal of Wildland Fire, 28*(2), 85–98. https://doi.org/10.1071/WF18004.

Oakey, A., Waters, T., Zhu, W., et al. (2021). Quantifying the effects of vibration on medicines in transit caused by fixed-wing and multi-copter drones. *Drones, 5*(1), 22. https://doi.org/10.3390/drones5010022.

Ollero, A., & Merino, L. (2006). Unmanned aerial vehicles as tools for forest-fire fighting. *Forest Ecology and Management, 234*(1), S263.

Pausas, J. G., & Keeley, J. E. (2021). Wildfires and global change. *Frontiers in Ecology and the Environment, 19*(7), 387–395. https://doi.org/10.1002/fee.2359.

Penny, S. G., White, R. L., Scott, D. M., et al. (2019). Using drones and sirens to elicit avoidance behaviour in white rhinoceros as an anti-poaching tactic. *Proceedings of the Royal Society B, 286*(1907), 20191–135. https://doi.org/10.1098/rspb.2019.1135.

Pham, H. X., La, H. M., Feil-Seifer, D. et al. (2017). A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 6648–6653). https://doi.org/10.1109/IROS.2017.8206579

Pradhan, B., Suliman, M. D. H. B., & Awang, M. A. B. (2007). Forest fire susceptibility and risk mapping using remote sensing and geographical information systems (GIS). *Disaster Prevention and Management: An International Journal, 8,* 9. https://doi.org/10.1108/09653560710758297.

Ramachandran, R. K., Pierpaoli, P., Egerstedt, M., et al. (2022). Resilient monitoring in heterogeneous multi-robot systems through network reconfiguration. *IEEE Transactions on Robotics, 38*(1), 126–138. https://doi.org/10.1109/TRO.2021.3128313.

Roldán-Gómez, J. J., González-Gironda, E., & Barrientos, A. (2021). A survey on robotic technologies for forest firefighting: Applying drone swarms to improve firefighters' efficiency and safety. *Applied Sciences, 11*(1), 363. https://doi.org/10.3390/app11010363.

Schoenherr, A. A. (2017). *A natural history of California*. Univ of California Press.

Schwarzrock, J., Zacarias, I., Bazzan, A. L., et al. (2018). Solving task allocation problem in multi unmanned aerial vehicles systems using swarm intelligence. *Engineering Applications of Artificial Intelligence, 72,* 10–20. https://doi.org/10.1016/j.engappai.2018.03.008.

Seraj, E., & Gombolay, M. (2020). Coordinated control of uavs for human-centered active sensing of wildfires. In *2020 American Control Conference (ACC)* (pp. 1845–1852). https://doi.org/10.23919/ACC45564.2020.9147613

Seraj, E., Silva, A., & Gombolay, M. C. (2019). Safe coordination of human–robot firefighting teams. CoRR abs/1903.06847. arXiv:1903.06847

Seraj, E., Chen, L., & Gombolay, M. C. (2022). A hierarchical coordination framework for joint perception-action tasks in composite robot teams. *IEEE Transactions on Robotics, 38*(1), 139–158. https://doi.org/10.1109/TRO.2021.3096069.

Sherstjuk, V., Zharikova, M., & Sokol, I. (2018). Forest fire-fighting monitoring system based on uav team and remote sensing. In *2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)* (pp. 663–668). IEEE. https://doi.org/10.1109/ELNANO.2018.8477527

Spears, W., & Spears, D. (2012). Physicomimetics: Physics-Based Swarm Intelligence. https://doi.org/10.1007/978-3-642-22804-9

Spears, W. M., Spears, D. F., & Heil, R. et al. (2004). An overview of physicomimetics. In *International workshop on swarm robotics* (pp. 84–97). Springer. https://doi.org/10.1007/978-3-540-30552-1_8

Steffen, A. D. (2020). Drones are delivering medical supplies to the isle of wight. https://www.intelligentliving.co/drones-medical-supplies-isle-of-wight/

Stolfi, D. H., Brust, M. R., & Danoy, G. et al. (2020). A cooperative coevolutionary approach to maximise surveillance coverage of uav swarms. In *2020 IEEE 17th annual consumer communications & networking conference (CCNC)* (pp. 1–6). IEEE. https://doi.org/10.1109/CCNC46108.2020.9045643

Tedim, F., Leone, V., Amraoui, M., et al. (2018). Defining extreme wildfire events: Difficulties, challenges, and impacts. *Fire, 1*(1), 9. https://doi.org/10.3390/fire1010009.

Viseras, A., Meißner, M., & Marchal, J. (2021). Wildfire front monitoring with multiple uavs using deep q-learning. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3055651.

Yang, X. S. (2014). Swarm intelligence based algorithms: A critical analysis. *Evolutionary Intelligence, 7*(1), 17–28. https://doi.org/10.1007/s12065-013-0102-2.

Yang, J., Qian, J., & Gao, H. (2021). Forest wildfire monitoring and communication uav system based on particle swarm optimization. *Journal of Physics: Conference Series.*https://doi.org/10.1016/j.jocs.2019.04.009.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Georgios Tzoumas[1,2] · **Lenka Pitonakova**[3] · **Lucio Salinas**[1,2] · **Charles Scales**[3] · **Thomas Richardson**[1,2,4] · **Sabine Hauert**[1,2]

Lenka Pitonakova
lpitonakova@windracers.org

Lucio Salinas
lucio.salinas@bristol.ac.uk

Charles Scales
cscales@windracers.org

Thomas Richardson
thomas.richardson@bristol.ac.uk

[1] Bristol Robotics Laboratory, Bristol, UK

[2] Engineering Mathematics, University of Bristol, Bristol, UK

[3] Windracers ltd, Southampton, UK

[4] Aerospace Engineering, University of Bristol, Bristol, UK