



Real-time autonomous path planning for dynamic wildfire monitoring with uneven importance

S M Towhidul Islam¹ · Xiaolin Hu¹

Accepted: 31 May 2024 / Published online: 28 June 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Unmanned Aircraft Systems (UASs) offer many benefits in wildfire monitoring when compared to traditional wildfire monitoring technologies. Wildfire spread is a highly heterogeneous process with non-uniform spreading speed and fireline intensity in both space and time. When planning the path of a UAS for monitoring a dynamically growing wildfire, it is important to consider the uneven spread of the fire because different parts of the fire boundary would need different levels of monitoring attention. This paper presents a real-time path planning algorithm for a UAS to autonomously monitor the perimeter of a spreading wildfire with uneven importance. The proposed path planning algorithm allows the UAS to focus on the most active regions of a wildfire while still covering the entire fire perimeter. The design and implementation of this algorithm are described, and an in-depth analysis of the performance of the algorithm is provided. Experiment results based on simulated wildfires demonstrate the effectiveness and robustness of the proposed algorithm for monitoring dynamic wildfires.

Keywords UAS path planning · Wildfire monitoring · On-board computation · Simulation

1 Introduction

Due to the increasing frequency of wildfires in recent years [1], wildfire monitoring has become a vital task and a subject of greater interest for researchers and practitioners. To support wildfire management, different strategies and technologies for wildfire monitoring and data collection have been used by them, including the use of satellite systems, manned aircraft systems, ground sensors, etc. Despite having some unique benefits, each of these technologies has its own limitations in terms of application. These limitations include—but are not limited to lower effectiveness, lower safety, higher cost, and limited adaptability to dynamically spreading wildfires. For example, satellite images typically have a lower spatial and temporal resolution [2]; helicopter missions are relatively unsafe and costly; real-time deployment of ground sensors is difficult and

time-consuming, and it is impractical to make such sensor systems adaptive to the size and spread of a fire.

A more effective and cost-efficient option for wildfire monitoring is using Unmanned Aircraft Systems (UASs). Due to the recent developments of highly capable UASs, it is considered a very suitable option for collecting important information from wildfires. Modern UASs can fly over a wide range of altitudes at desired speed for hours. They can be equipped with different types of sensors and cameras to collect valuable information about a wildfire. Furthermore, UASs have the potential to fly autonomously to monitor a fire with minimal external supervision and communication. Altogether, UASs hold great potential for collecting real-time wildfire data and supporting wildfire management.

A variety of factors influence the wildfire spreading process in a dynamic way. For example, the spatiotemporal speed and intensity of the fire spread can be impacted by different fuel loading, non-uniform terrain, and dynamic weather conditions. As a result, different parts of the fire spread at different speeds. Generally, the head of a fire has more frequent changes in fire state than the tail of the fire. Moreover, such active region(s) of a spreading wildfire might change over time due to the dynamic weather condition and non-uniform terrain and fuels across the wildland. From the wildfire monitoring perspective, it is important

✉ S M Towhidul Islam
sislam9@gsu.edu

Xiaolin Hu
xhu@gsu.edu

¹ Department of Computer Science, Georgia State University, Atlanta, GA, USA

for the more active regions to be visited more frequently in order to capture a more accurate state of the fire spread.

To effectively monitor wildfires with heterogeneous spreading behavior, previously we developed an importance based multi-UAS path planning algorithm that coordinates multiple UASs to monitor a simulated wildfire [3]. That work assumes that all the related wildfire and UAS data are available on a central computer (e.g., a ground station). The central computer makes the path planning decisions and only the results are communicated to each UAS for execution. Thus, even if that algorithm is restricted to a single UAS, it will still depend on a centralized computer to support importance-based path planning. However, wildfires happen in extremely challenging environments, e.g., in mountain areas that have limited or unstable wireless communications. To achieve robust and autonomous path planning, it is necessary for the UASs to have real-time on-board path planning capabilities so that they can monitor a wildfire without depending on a centralized computer. Furthermore, the algorithm presented in our previous work [3] has not been specifically designed considering a single UAS, it mainly focused on multi-UAS coordination. If this algorithm is used in a single UAS scenario, the UAS will cover the entire fire perimeter in a back-and-forth approach without focusing on the most active fire regions. Thus, a more tailored algorithm is required to achieve the importance-based wildfire monitoring goals in a single UAS scenario. Motivated by the above needs, this paper presents a novel on-board path planning algorithm for wildfire monitoring using a single UAS. The proposed algorithm uses real-time data collected by the UAS and still supports importance-based monitoring by paying more attention to the more active fire regions.

Real-time on-board path planning in a decentralized way for wildfire monitoring faces several unique challenges compared to a centralized approach. First and foremost, the UAS lacks full knowledge about the wildfire. To achieve effective path planning, a UAS needs to know the fire perimeter and rate of spread in real-time. For large-scale wildfires, at any moment a UAS can only monitor a small portion of the fire perimeter that is within the field of view of the UAS. This means the UAS needs to have a mechanism to construct the full fire perimeter in real-time and estimate the rate of spread of the different segments of the perimeter. Second, due to real-time requirements and the limited computing resources of UAS, computation efficiency is important for real-time on-board path planning. This means the proposed algorithm should take computation cost into consideration and achieve a balance between accuracy and computation cost.

Based on the design principles described above, the developed algorithm presented in this paper dynamically constructs the perimeter of a wildfire based on real-time collected data and then uses it to perform fire monitoring with uneven importance. The path planning focuses on

determining the UAS's best flying direction (clockwise or counterclockwise) along the fire perimeter to effectively monitor a spreading wildfire. This paper significantly extends our previous work [4] that presented an initial design of the algorithm and preliminary results. It makes the following new contributions compared to the previous work: i) an improved algorithm is provided that introduces a new heterogeneity-adjusting factor for modulating the tendency of staying at the active fire regions. In other words, the heterogeneity-adjusting factor can be used to control the frequency of back-and-forth visits over the most active fire region.; ii) a new analysis (both quantitative and qualitative) of the performance and robustness of the algorithm is added. These analyses help to reveal the characteristics of the algorithm and to better understand how it works; iii) comprehensive experiments are carried out to demonstrate the features and effectiveness of the algorithm. This includes more challenging experiment scenarios such as non-uniform fuel loadings and different wind conditions.

The remainder of the paper is organized as: Section II describes the related works. Section III describes the conceptual design of the UAS path planning considering uneven importance. Section IV describes the implementation details of the real-time autonomous path planning algorithm. Section V presents analyses of the algorithm from several important perspectives. Section VI shows the experiment results for different simulated fire scenarios. Finally, Section VII concludes the paper.

2 Related work

UAS-based wildfire monitoring has received more and more attention from researchers over the years as the technical capabilities of UASs increased. One of the major interests among researchers is how to monitor a wildfire using UASs efficiently and autonomously. Wildfire monitoring can be regarded as a path planning problem where the area needed to be covered is the fire boundary and the mobile agents to be used are the UASs.

The work of [5] presented a range of path planning strategies for unmanned aerial vehicles to cover different shapes of areas of interest, such as rectangular, concave, and convex polygons. Different flight patterns were described, including geometric flight patterns, such as back-and-forth and spiral, and more complex grid-based solutions to cover areas with different shapes. However, the problem of monitoring a wildfire is different in the extent that the shape of the fire is different from a regular geometric shape. Moreover, the shape also changes over time as the fire progresses. Another UAS based path planning strategy was presented in [6] which uses an augmented planning space to generate vehicle paths concentrated around the area of interest, and its application has been demonstrated in precision farming. In

case of multiple UASs, the area is partitioned into multiple sub-areas and then the path planning technique is applied to the smaller regions. An algorithm to compute UAS waypoints in nonconcave regions was presented in [7] where the UAS follows a spiral coverage pattern – starting from the boundary of the area towards the inner regions. This algorithm leads to a uniform coverage pattern where different regions of the area receive even monitoring attention. In case of a dynamically spreading wildfire, different parts of the fire demand different levels of monitoring attention. Thus, the UAS based wildfire monitoring task is a different problem from many of the existing area coverage problems.

Several previous works demonstrated the use of UASs in the context of wildfire monitoring. A deep learning-based forest fire monitoring system was presented in [8]. That system uses images acquired from unmanned aerial vehicles through the connected optical sensor. A convolutional neural network is pre-trained with past forest fire images and the unmanned aerial vehicle sends query images to this network. The presented system is dependent on a centralized system for recognizing a wildfire. In [9], the authors developed a collaborative framework for multi-agent reinforcement learning to train Unmanned Aerial Vehicles (UAVs) for monitoring wildfires. The work of [10] presents a multi-modal UAV dataset with dual-feed videos (RGB and thermal) capturing a prescribed fire and introduces a deep learning method for detecting fire and smoke pixels. In [11], the authors study the performance and reliability of the UAV-IoT networks in wildfire detection and conclude that such networks offer more efficiency and reliability compared to the state-of-the-art satellite imaging-based solutions. The work of [12] proposes a real-time wildfire state estimation system to improve fire spread prediction and perimeter propagation tracking.

Though some previous works addressed the problem of fire perimeter monitoring, however, very few of them investigated the problem considering the uneven importance of the fire perimeter. The work of [13] provides an extensive overview of different continuous object tracking methodologies and discusses their strengths and weaknesses. According to this survey, the accuracy of boundary detection and localization remains an open research challenge. The work of [14] proposes a high-level controller for UAVs considering dynamic resource allocation. The allocation is based on a cost function that incorporates fire location, wind speed and direction. In [15], the authors present a UAV path planning approach for wildfire monitoring considering realistic models of UAVs, terrain, and fire propagation. The work of [16] proposes and evaluates a decentralized approach for UAV-based autonomous fire boundary tracking. For the patrolling problem with multiple UAVs, four different strategies have been discussed in [17] considering time, uncertainty, and communication. In this work, the authors presented an

improved version of their previously developed centralized algorithm.

The work of [3] presented a multi-UAS path planning algorithm for wildfire monitoring, which is defined as a fire perimeter coverage problem. The goal is to have a balanced coverage of the fire perimeter using multiple UASs so that the UASs can collect the most useful information about the fire and construct a fire shape as accurately as possible. The key concept is to treat different parts of the fire to have different levels of importance. UASs are assigned different regions of the perimeter to monitor those regions in a back-and-forth approach. Over time, adjacent UASs perform boundary negotiation between them to maintain a balanced responsibility. This work assumes that necessary path planning information will be provided to the UASs by a central ground station. A distributed leader–follower coalition formation model was presented in [18], where a set of drones are grouped into multiple sub-groups to cover a designated field. When a fire incident is reported, a mission is initiated by the UASs. A set of follower UASs relies on a group leader UAS for the monitoring task. This work supports on-board decision making; however, it does not support importance-based identification and monitoring of the most actively spreading region. A more sophisticated approach is required for the identification and monitoring of the most active fire region.

Given the most recent advancements of using UASs in wildfire monitoring, there is a high need for a real-time path planning algorithm that focuses more closely on the most active region of a fire while considering real-time and on-board computation. To the best of our knowledge, none of the previous works considered such a real-time, decentralized, and importance-based approach in UAS-based wildfire monitoring.

3 UAS path planning considering uneven importance

For a spreading wildfire, the location of the burning fire perimeter is one of the most useful information to fire managers. In this work, the proposed path planning strategy focuses on the burning perimeter of the fire, which is consistent with other works in the literature. We assume that the UAS has the capability of boundary following and always flies on top of the fire perimeter. In other words, the UAS works as a mobile agent covering the boundary of the fire. It follows the fire boundary by flying clockwise (CW) or counterclockwise (CCW) on top of the perimeter. Thus, the UAS's path planning task essentially becomes deciding when the UAS needs to change its flying direction (CW or CCW) along the fire perimeter. As the most straightforward approach, a UAS can keep circling the fire perimeter without changing its flying direction. This allows the UAS to monitor

the perimeter of the fire in a cyclic fashion. Nevertheless, it does not allow the UAS to pay more attention to the more active fire perimeter segments because it treats all segments with even importance.

In real-life scenarios, a wildfire spreads at different speeds across different parts of the fire perimeter due to non-uniform fuel, terrain, and weather condition. Therefore, a segment of the fire that spreads more actively demands more monitoring attention compared to a segment that spreads slowly. This work focuses on this need and performs path planning based on the uneven importance of the burning fire perimeter segments.

To demonstrate such a path planning need, let's consider a sample fire spreading scenario presented in Fig. 1. The red region represents the fire region, the white dot in the middle represents the ignition point (from where the fire started), the yellow line represents the UAS trajectory since the UAS has been deployed, and the white arrows represent fire spread at different directions of the fire. It is clear that the fire spread is uneven across different regions of the fire and more monitoring attention is required on the fast-spreading region. Since fire spreads in a non-uniform way, it makes sense for the UAS to turn back and forth to monitor the fire segment that spreads very fast, while still covering the entire fire perimeter from time to time. The proposed path planning algorithm aims to decide when the UAS needs to change its flying direction along the fire perimeter. It focuses on the high-level path planning instead of the low-level UAS control.

We represent the fire area as a 2D space and discretize it into a cell space, whose size can be set based on the granularity of path planning. Accordingly, in the path planning algorithm, we represent the fire perimeter as a discretized boundary comprising individual cells corresponding to the locations of the fire perimeter. Based on this discretized

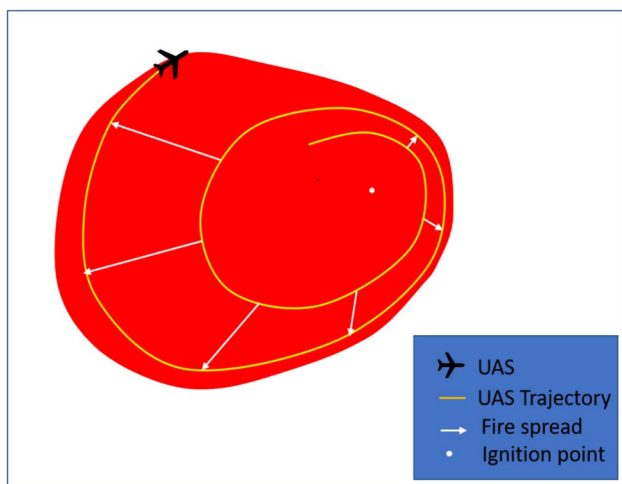


Fig. 1 Sample fire spread scenario

approach, we assume the UAS always flies from the center of a cell to the center of a neighboring cell that is on the fire perimeter along its flying direction (CW or CCW). After reaching the corresponding neighboring cell, it finds a new neighboring cell on the fire perimeter as the next destination to fly to and so on. When the UAS reaches a new cell, the proposed algorithm helps to decide if it should continue flying in the current flying direction (CW or CCW), or, it should turn back to cover more important segments of the fire perimeter. To make this decision, the UAS calculates the importance of visiting each perimeter cell, called cell importance, and utilizes the cell importance values. We note that this approach of space discretization is useful for providing a level of granularity for the UAS's decision making (e.g., how often to carry out a path planning decision) as well as the importance computation (e.g., a continuous fire perimeter is divided into discretized cells for computing the importance). This design choice introduces some approximation but still works with a real-world environment. Below we describe in detail how the cell importance values are calculated and how the path planning is performed based on those cell importance values.

3.1 Cell Importance

The basic idea of importance-based path planning is to treat different segments of a fire perimeter to have different importance levels that represent different levels of monitoring attention. Each cell that makes up the fire perimeter is assigned a dynamic value that represents the importance of visiting that cell for data collection. This importance can be thought of as the value of the data to be collected from a cell, which is related to the information uncertainty of that cell. If a specific segment of the fire perimeter spreads very fast and/or has not been visited for a long time, then there is more information uncertainty for that segment of the fire perimeter, and thus the segment has more importance.

Based on the above idea, the algorithm utilizes several key measurements that are necessary for quantifying the importance of the different parts of the fire perimeter. For the on-board path planning, the UAS needs to calculate those measurements in real-time based on the information collected along its flying trajectory. As the UAS flies, it saves the visiting time for each cell that it has visited. The cells that have been visited by the UAS are referred to as the trajectory cells in this paper. For each trajectory cell, the UAS assigns an angle value to it to represent the direction of the cell from the ignition point of the fire. Due to limited computing resources, we only consider a discretized set of angle values. The granularity of the discretization of the angles is a tradeoff between computation precision and on-board computation efficiency. In this work, we consider 360 discrete angles to keep track of the visited cells as it provides

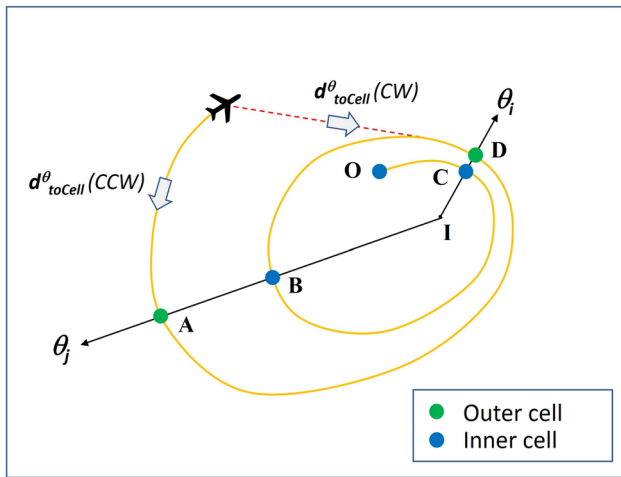


Fig. 2 Angle based trajectory cell tracking

adequate precision and efficiency. Figure 2 illustrates four trajectory cells A, B, C, D and their corresponding angles θ_j and θ_i . Among them, cell A and B have the same angle θ_j , and cell C and D have the same angle θ_i . The angle lines are based on the cell I, which is the original ignition point of the fire. Cell O represents the location where the UAS is first deployed. In this example, the UAS flies clockwise around the fire without changing direction. The real fire shape is not shown in the figure.

The importance of a cell is directly related to how fast the fire spreads in the direction of the cell and how much time has elapsed since the cell was last visited. Based on this idea, we consider three measurements for computing the importance of a cell at an angle θ : 1) the rate of spread (ROS^θ); 2) the elapsed time since the cell was last visited ($t^\theta_{sinceVisit}$), and 3) the time it will take for the UAS to reach the cell along the fire perimeter from its current location (t^θ_{toCell}).

First, the ROS^θ is the measure of how fast the fire is spreading along the direction of angle θ . To calculate ROS^θ , we use the most recent trajectory cell along the angle θ and the previous trajectory cell along the same angle. The former is referred to as the *outer cell*, and the latter is referred to as the *inner cell*. Specifically, in our implementation for each angle θ , the UAS keeps track of the distance of the outer cell and inner cell from the ignition point (d^θ_{outer} and d^θ_{inner}) and the visit time of those cells (t^θ_{outer} and t^θ_{inner}). When the UAS reaches a new cell at the same angle θ , the previous outer cell at that angle becomes the inner cell. The outer cells and inner cells at two sample angles θ_i and θ_j are displayed in Fig. 2. With the information of the outer cell and inner cell the ROS^θ is calculated using (1).

Secondly, for each angle θ , the time since last visit $t^\theta_{sinceVisit}$ is measured by the difference between the current

time ($t_{current}$) and the visit time of the outer cell at the direction θ and calculated as per (2).

$$ROS^\theta = \frac{d^\theta_{outer} - d^\theta_{inner}}{t^\theta_{outer} - t^\theta_{inner}} \tag{1}$$

$$t^\theta_{sinceVisit} = t_{current} - t^\theta_{outer} \tag{2}$$

$$t^\theta_{toCell} = \frac{d^\theta_{toCell}}{UAS_SPEED} \tag{3}$$

$$imp^\theta = ROS^\theta * t^\theta_{sinceVisit} + ROS^\theta * t^\theta_{toCell} \tag{4}$$

The multiplication of the ROS^θ and $t^\theta_{sinceVisit}$ allow us to compute how far the fire has spread in the direction of a cell since the cell was last visited. This spreading distance is directly related to the importance of the cell at the current moment. Note that the ROS^θ may dynamically change due to the varying condition of the fire area. In this work, we ignore that dynamics and use the most recent ROS^θ to calculate the spreading distance.

While we can develop the path planning algorithm based on the current importance of cells, it makes more sense to consider the “potential importance” of cells, which is the importance when the UAS actually reaches the direction of the cell from its current location following the fire perimeter. This is because it takes time for the UAS to reach a cell and during this time period the fire will spread further in the direction of the cell. The longer it will take for the UAS to reach a cell, the larger the “potential importance” is. Based on this idea, when computing the importance of a cell we consider the third measurement t^θ_{toCell} , which is the time needed for the UAS to travel from its current location to the cell at the angle θ . Since the UAS always flies CW or CCW along the fire perimeter, the t^θ_{toCell} would be the “perimeter distance” to the cell, denoted as d^θ_{toCell} , divided by the flying speed of the UAS, as shown in (3).

To calculate t^θ_{toCell} , we need to know d^θ_{toCell} . Several things are worthy of mention for computing the d^θ_{toCell} . First, depending on if the UAS flies CW or CCW, the distance to a specific cell would be different. In the example of Fig. 2, if the UAS keeps moving CW, the d^θ_{toCell} to cell A would be the length covering all the perimeter cells between the UAS and cell A along the CW direction. If the UAS turns back and moves CCW, the d^θ_{toCell} to cell A would be the length covering the perimeter cells between the UAS and cell A along the CCW direction. Second, as the fire is always spreading, the UAS does not have full knowledge about the fire perimeter. This means to compute d^θ_{toCell} the UAS needs to construct the fire perimeter based on the information it has collected. The fire perimeter construction needs to take

into account the fact that the trajectory cells following the UAS might not be smoothly aligned into a closed loop at the current location of the UAS. This is because, as the fire is continuously spreading, there is some unvisited fire region ahead of the UAS. To fill this monitoring gap, it is important to construct the fire shape as accurately as possible based on the UAS trajectory. We use a fire shape construction procedure to estimate the fire perimeter of that unvisited region as shown by the dashed red line in Fig. 2 (see later for more details). The reconstructed fire perimeter represents the best knowledge about the real fire shape based on the UAS trajectory. Thus, we use this reconstructed perimeter as the real fire perimeter to calculate d_{toCell}^θ . Third, because the fire is always spreading, by the time the UAS reaches the direction of a target cell, the fire has already spread further. In other words, there is an error between the d_{toCell}^θ that we calculate based on the “current perimeter” and the actual travel distance for reaching the direction of the target cell (in fact, d_{toCell}^θ would always be smaller than the actual travel distance because the fire grows larger). In our current implementation, we ignore this error and use the currently estimated fire perimeter to compute the d_{toCell}^θ .

Based on the three measurements described above, we calculate the importance of a cell using (4). Here, the importance of a perimeter cell is determined by two factors: $ROS^\theta * t_{sinceVisit}^\theta$ and $ROS^\theta * t_{toCell}^\theta$. The first factor represents the new spread distance at the angle θ since the cell was last visited, and the second factor represents the new spread distance at the angle θ until the cell is projected to be visited again. Thus, the importance of a cell is essentially defined by the new fire spread distance in the direction of a cell between when the cell was last visited and when the cell is projected to be re-visited by the UAS. These two factors have specific types of influences on the path planning of a UAS. How these two factors impact the path planning of a UAS will be discussed in detail in the Analysis section of this paper (Section V).

3.2 Cell Importance based path lanning

Having described the importance of visiting a cell, the next step is to utilize cell importance values to carry out path planning. In our design, we say that the UAS should keep moving forward unless it is more beneficial to turn back, i.e., to change its flying direction. It makes sense for the UAS to turn back when the fire is spreading significantly faster on the back of the UAS. In such situations, the cell importance values in the back of the UAS are higher. A mechanism is required to identify if the total cell importance at the back of the UAS is higher compared to the front of the UAS.

To measure how much benefit the UAS can obtain by keep moving forward or by moving backward, we introduce a concept called planning window. We consider two identical

sized planning windows from the current position of the UAS—one window represents 50% of the fire perimeter at the back of the UAS called the back window and the other one represents 50% of the remaining fire perimeter in front of the UAS called the front window. Here, a 50%-50% split of the perimeter ensures that the front and back windows don't overlap. Furthermore, it also ensures that every cell of the fire perimeter is considered in the decision-making process.

The UAS then computes and compares the cumulative importance in the front and back window and makes its decision regarding its optimal flying direction. Between the two defined planning windows, the UAS dynamically chooses the one with more importance. For that purpose, we calculate the total importance of the back window (IMP_{BACK}) and compare it to the total importance in the front window (IMP_{FRONT}). If IMP_{BACK} is larger, it means that fire in the back window is more active than the front window, therefore, the UAS changes its flying direction along the perimeter to cover the more active region. Every time the UAS reaches a new cell, this procedure for checking the best flying direction is invoked. Thus, the real-time UAS path planning is performed dynamically to guide the UAS towards the best flying direction along the perimeter.

Assuming there are total $2N$ number of cells in the current perimeter, IMP_{BACK} and IMP_{FRONT} are calculated according to (5) and (6) respectively, where θ_i represents the angle of the i^{th} perimeter cell. Here, the index (i ; where $1 < i < 2N$) of a perimeter cell is relative to the position of the UAS on the fire perimeter. As the indexing convention, we consider the indices start from the cell that is currently occupied by the UAS; it increases along the opposite direction of UAS's flying direction on the perimeter; and ends at the cell that is just in front of the UAS.

$$IMPORTANCE_{BACK} = \sum_{i=1}^N imp^{\theta_i} \quad (5)$$

$$IMPORTANCE_{FRONT} = \sum_{i=N+1}^{2N} imp^{\theta_i} \quad (6)$$

The overall concept of decomposing the fire perimeter into the back and front planning window is illustrated in Fig. 3. Based on the fire spreading scenario presented in Fig. 1, the dashed red line represents the perimeter of the actual fire shape. The blue line represents the back window, the black line represents the front window. It is worth noting that the back window and front window are based on the fire perimeter constructed by the UAS using the on-board data in real-time. This constructed fire perimeter is different from the actual fire perimeter (the red line in Fig. 3), which is unknown to the UAS. The rest of the UAS trajectory marked

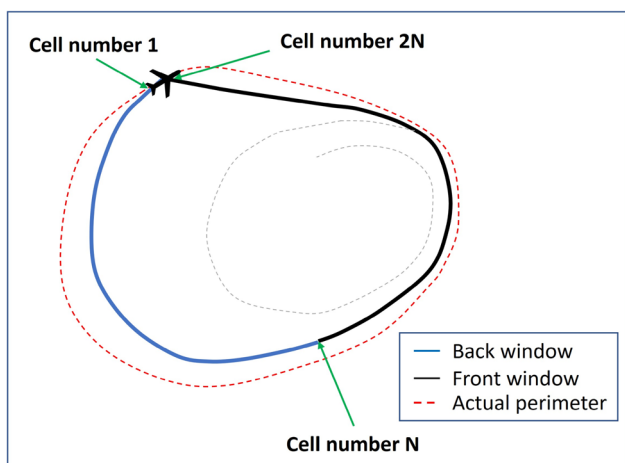


Fig. 3 Importance based path planning

by the grey dashed line is not a part of the reconstructed fire perimeter. Therefore, in this sample scenario, IMP_{BACK} is the sum of the importance of the cells falling under the blue line (cell number 1 to N), and IMP_{FRONT} is the sum of the importance of the cells falling under the black line (cell number N + 1 until 2N). By comparing these two values, the best flying direction is determined.

3.3 Modulating path planning sensitivity by adjusting heterogeneity of fire spread

The heterogeneous spreading behavior of the fire has a significant impact on the importance-based UAS path planning algorithm. In the proposed algorithm, the UAS changes its flying direction based on the total importance in the front and back planning windows, which is directly related to the heterogeneous rates of spread of the different cells on the fire perimeter. Therefore, heterogeneity of the fire spread is an important consideration in the proposed algorithm. If the fire spread is more heterogeneous, the UAS changes its flying direction more frequently compared to when the fire is less heterogeneous. Based on this characteristic, we introduce an additional parameter for the algorithm, named as the heterogeneity-adjusting factor (α), which provides the proposed algorithm with the capability of modulating the tendency of flying direction change.

In our fire spread model, the heterogeneity of fire spread is represented by the ROS values of the perimeter cells. The heterogeneity-adjusting factor is a non-negative real number (i.e., $\alpha \geq 0$) that works by adjusting the ROS values of the perimeter cells when computing the importance of the front and back planning windows. Equation (7) shows how the adjustment is carried out for a cell, where ROS^{θ} is the original ROS value of the cell (the value before adjustment),

ROS_{min} is the minimum ROS value among all the cells, ROS_{avg} is the average of the ROS values of all the cells, and $ROS^{\theta}_{adjusted}$ is the adjusted ROS value. When making the fire spread to be more heterogeneous than the original fire spread (i.e., $\alpha > 1$), the adjustment is applied to those perimeter cells that have larger ROS values than the ROS_{avg} . Other cells which have lower ROS values than the ROS_{avg} hold their original ROS values as shown in (7). Thus, the ROS values of the above-average cells are magnified to make the fire spread more heterogeneous. In contrast, when making the fire spread less heterogeneous (i.e., $0 \leq \alpha < 1$) the adjustment is applied to all the cells on the fire perimeter. Thus, based on the value of α and ROS_{avg} , the adjusted ROS values of the perimeter cells are calculated as per (7).

$$ROS^{\theta}_{adjusted} = \begin{cases} ROS^{\theta} & ROS^{\theta} \leq ROS_{avg} \text{ and } \alpha \geq 1 \\ ROS_{min} = \alpha(ROS^{\theta} - ROS_{min}) & \text{otherwise} \end{cases} \tag{7}$$

When $\alpha = 0$, the adjusted ROS values of every perimeter cell is constant ($ROS^{\theta}_{adjusted} = ROS_{min}$), thus there is no heterogeneity in the adjusted ROS values. As α increases, the adjusted heterogeneity of the fire spread also increases. When $\alpha = 1.0$, the adjusted heterogeneity is the same as the original heterogeneity. The adjusted fire spread is more heterogeneous than the actual spread when α is larger than 1. Figure 4 shows the heterogeneity adjustment for different α values for a sample fire spread scenario. We can see that there is no heterogeneity in the fire spread when $\alpha = 0$, while the fire spread is very heterogeneous when $\alpha = 5$. By choosing different α values, one can modulate the level of sensitivity of the path planning (i.e., the tendency of flying direction change of the UAS) to the heterogeneity of the fire spread.

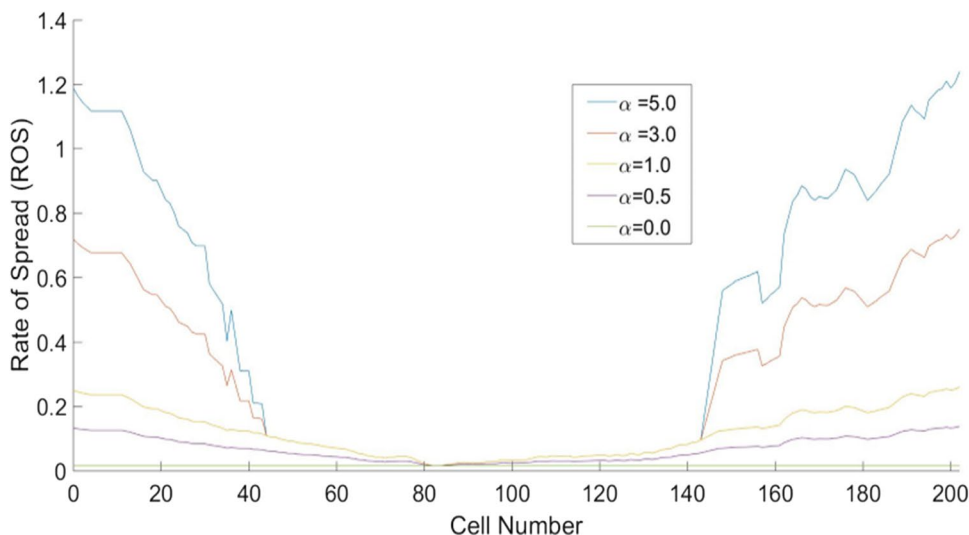
4 The real-time autonomous path planning algorithm

Based on the importance-based path planning concepts discussed in the previous section, this section describes the design and implementation details of the real-time autonomous path planning algorithm. The algorithm can be decomposed into three main components: i) Initialization of the on-board fire map ii) Fire shape construction iii) Determining the optimal flying direction. The overall steps involved in the algorithm are shown in Table 1.

4.1 Initialization of the on-board fire map

The UAS is deployed at the boundary of the fire after a certain time since the fire was started. The UAS needs the initial fire spread information such as which cells are burning at the time of UAS deployment. In our previous work, we assumed that

Fig. 4 Heterogeneity adjustment for fire spread



initial fire spread information will be provided at the time of UAS deployment. However, this may not be always practical to have an exact fire spread information beforehand. In this work, we implemented an automated on-board fire map initialization process which removes the dependency of external input for initialization. In this process, the UAS completes one initial round along the fire perimeter to gather the initial fire spread information such as the location of the fire perimeter. The UAS is initially deployed at a perimeter cell and the angle of that cell relative to the ignition point is denoted as θ_{init} . The UAS keeps flying along the fire perimeter until it returns back to θ_{init} and thus completes one round. Along the way, the UAS generates and updates an on-board fire map to keep track of the necessary information of the fire spread. The structure of the on-board fire map is shown in Table 2. The map is based on 360 discrete angles around the ignition point and it keeps track of the outer cells and inner cells along each angle. It is worthwhile to note that the map may not contain any outer and inner cells for some angles, e.g., when there are no cells located along those angles. In contrast, the map may contain multiple outer and inner cells along a same angle, e.g., when those cells lie along the same angle.

When the UAS is initially deployed, it has no knowledge about the inner cells of the initial fire perimeter. In our implementation, the ignition point is considered as the initial inner cell. Then, as the UAS moves along the perimeter, the

outer cells and inner cells are updated on the map accordingly. Note that here we assume the ignition point is known. When the ignition cell is unknown, we can use the center of the fire as the ignition cell. The center of the fire can be calculated after the UAS finishes the initial circling of the fire perimeter. The on-board fire map initialization procedure is shown in Table 3.

4.2 Fire shape construction

Every time the UAS arrives at a new cell, an on-board fire perimeter construction procedure is invoked to generate a more accurate fire perimeter. The on-board fire perimeter construction procedure utilizes the previous trajectory cells to form a full loop of the fire perimeter. To deal with the issue that the trajectory cells following the UAS might not be smoothly aligned into a closed loop at the current location of the UAS (as illustrated in Fig. 5(a)), a scan method is used to find the best cell among the previous trajectory cells that can generate a smooth closed loop by connecting with the UAS' current cell. The scan method works as below: starting from the inner cell of the UAS' current location, it keeps scanning

Table 1 On-board fire map

Angle	Outer cells	Inner cells
0	outer_cells ₀	Inner_cells ₀
1	outer_cells ₁	Inner_cells ₁
...
359	outer_cells ₃₅₉	Inner_cells ₃₅₉

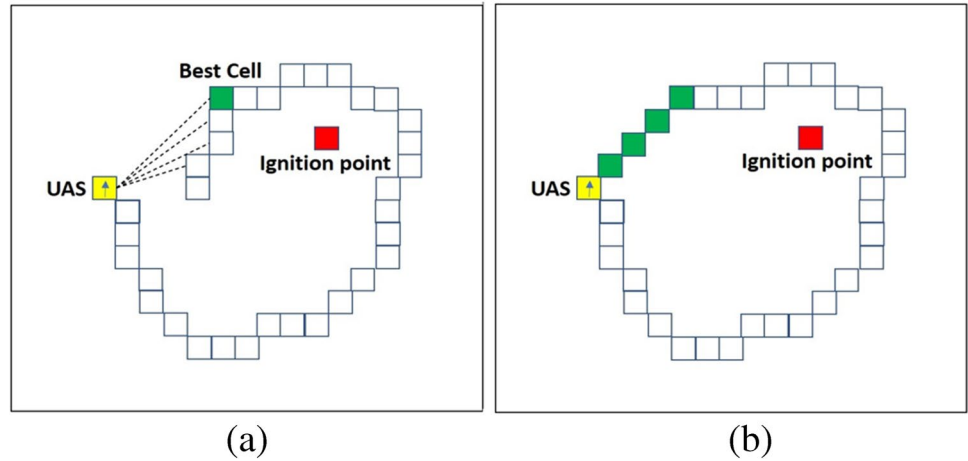
Table 2 Overview of the proposed algorithm

1. Initialize the on-board fire map when UAS is deployed
2. In each step (after UAS reaches a new cell), DO:
 - i. Reconstruct fire shape based on the on-board fire map
 - ii. Determine the importance of the backward and forward window
 - iii. IF $IMP_{BACK} > IMP_{FRONT}$
Change flying direction
ELSE
Keep flying in the current flying direction
 - iv. Select the next cell along flying direction and fly over there

Table 3 Procedure for on-board fire map initialization

OUTPUT: Initialized on-board fire map
 Determine the angle θ_{init} of the UAS deployment cell from the ignition point
REPEAT UNTIL the UAS comes back to θ_{init} :
 i. Determine the angle θ of the cell from the ignition point
 ii. Add the cell as the outer cell at the angle θ in on-board fire map
 iii. Add ignition point as the inner cell at the angle θ
 iv. Move to the next cell along UAS's current flying direction
END

Fig. 5 On-board fire perimeter construction **a** before construction **b** after construction



the trajectory cells along the moving direction of the UAS as long as the angle from UAS's current cell to the trajectory cell keeps increasing. This scan method is based on the assumption that fire spread tends to generate convex shapes.

Figure 5 illustrates how the on-board fire perimeter construction works. In Fig. 5(a), the white cells are previous trajectory cells; the yellow cell is the UAS's location, and the best cell found by the scan method is represented by the green cell. This is the best cell because the angle (shown by dashed lines) from UAS's current position keeps increasing until this cell. Beyond this cell, the angle decreases if scanned further along the trajectory cells. Finally, the trajectory cells between the UAS occupied cell and the best cell are replaced by the cells that fall under the connecting straight line, as highlighted by the green cells in Fig. 5(b). This way a smooth fire shape is generated using the on-board information collected by the UAS.

4.3 Determining optimal flying direction

After the fire perimeter construction, the final step is to calculate IMP_{BACK} and IMP_{FRONT} and decide the optimal flying direction for the UAS. The decision-making procedure is shown in Table 4. This procedure needs the on-board fire map, UAS's current flying direction as inputs. From the current position, the UAS first extracts a list of connected cells that represents the perimeter of the constructed fire shape. Next, it calculates the importance of each cell on

the constructed perimeter according to (4). Finally, the total importance of the front and back planning window is calculated from those individual cell importance values. The UAS changes its current flying direction if IMP_{BACK} is larger than IMP_{FRONT} .

5 Analysis of the proposed algorithm

In this section, we briefly analyze the proposed path planning algorithm from several important perspectives. These analyses are important to have a thorough understanding of how the proposed algorithm controls the path of a UAS. We

Table 4 Procedure for finding optimal flying direction

INPUT: On-board fire map, UAS's current flying direction, heterogeneity-adjusting factor
OUTPUT: Optimal flying direction (CW/CCW) along the fire perimeter
 Calculate the importance of the perimeter as:
FOR each cell "current_cell" in the on-board fire perimeter
DO:
 i. Calculate $ROS^{\theta}_{adjusted}$ of the current_cell
 ii. Calculate $t^{\theta}_{sinceVisit}$ of the current_cell
 iii. Calculate t^{θ}_{toCell} of the current_cell
 iv. Determine imp^{θ} of the current_cell
END FOR
 Calculate IMP_{BACK} and IMP_{FRONT}
 Change the flying direction if $IMP_{BACK} > IMP_{FRONT}$

divide the analyses into two parts – quantitative analyses and qualitative analyses. The quantitative analyses use a representative fire spread scenario to analyze the flying decisions made by the UAS at different locations of the fire perimeter. The qualitative analyses focus on factors that impact UAS’s flying direction in a more general context.

5.1 Quantitative analysis

We consider a representative fire spread scenario that has a minimum (r_{min}) and a maximum fire spread direction (r_{max}) as showed in Fig. 6(a), and the rate of spread (ROS) along the fire perimeter gradually increases from r_{min} to r_{max} as illustrated in Fig. 6(b). The difference of ROS between each pair of adjacent perimeter cell (δ) can be calculated as $\delta = (r_{max} - r_{min})/N$. Assuming r_{max} is a ($a > 1$) times larger than r_{min} (i.e., $r_{max} = ar_{min}$), then δ can be alternatively expressed as $\delta = (ar_{min} - r_{min})/N$. Thus, the rate of spread at any cell which is i cells away from the r_{min} direction can be expressed as $(r_{min} + i\delta)$. Based on the above description of the representative fire spread model, we consider four locations on the fire perimeter as shown in Fig. 6(a) and analytically find out whether the UAS should change its flying direction at those locations. We assume the speed of the UAS is s , the

size of each perimeter cell is c , the ROS of perimeter cell i is r_i , and the flying direction is CW.

To analyze the UAS’s decision regarding its flying direction change at those locations, we consider a metric (IMP_{DIFF}) which is calculated according to (8). If IMP_{DIFF} is positive at a location, the UAS should move forward at that location because IMP_{FRONT} is larger. Otherwise, the UAS should change its flying direction by turning back.

$$IMP_{DIFF} = IMP_{FRONT} - IMP_{BACK} \tag{8}$$

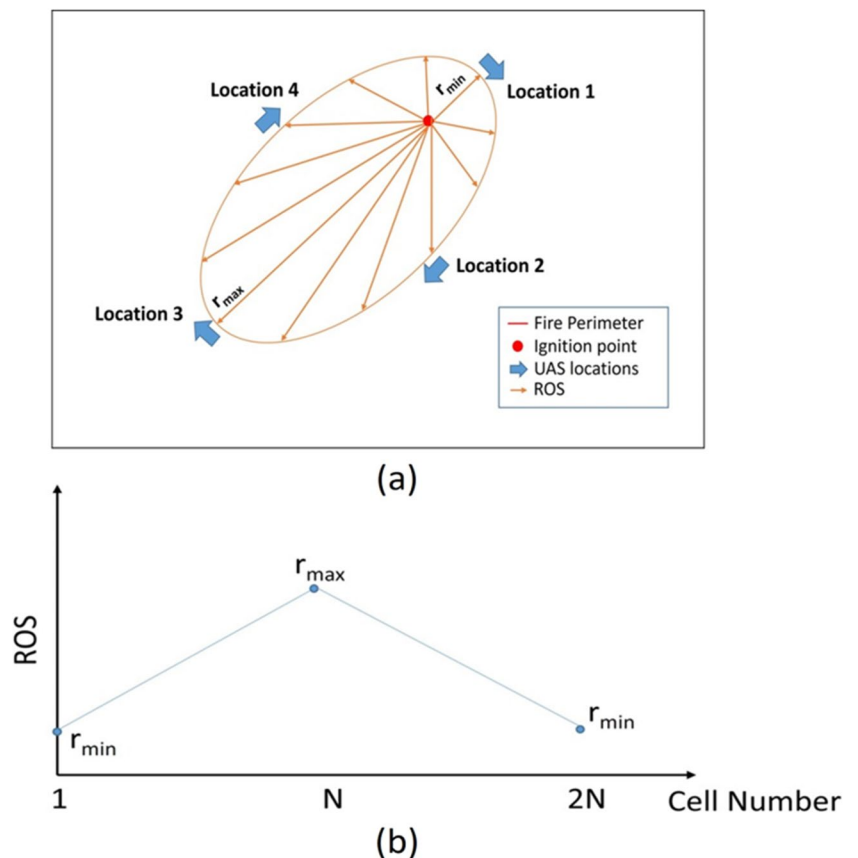
Below we show the IMP_{DIFF} at those four locations and discuss their impact on the UAS’s flying direction. The IMP_{DIFF} is calculated based on IMP_{FRONT} and IMP_{BACK} which have been shown in Appendix A for each of these four locations. To save space only the derived final equations have been shown there.

Location 1:

$$IMP_{DIFF} = \left[Nr_{min}(N - 1) + \frac{2\delta N^3}{3} + \frac{\delta N}{2} \left(N - \frac{1}{3} \right) \right] * \frac{c}{s}$$

Therefore, for location 1, if the window size (N) is larger than 1, IMP_{DIFF} is positive and the UAS should keep moving forward. Practically, N is always to be larger than 1, hence the UAS should always move forward at location 1.

Fig. 6 Representative fire spread scenario



Location 2:

$$\text{IMP}_{\text{DIFF}} = \frac{N_{\min}^{2r}}{2}(a-1) * \frac{c}{s}$$

Therefore, for location 2, if $a > 1$, i.e., $r_{\max} > r_{\min}$, IMP_{DIFF} is positive and the UAS should keep moving forward in the clockwise direction at this location.

Location 3:

$$\text{IMP}_{\text{DIFF}} = (2r_{\min} \sum_{i=1}^N i + 2\delta \sum_{i=1}^N i^2) * \frac{c}{s}$$

Therefore, similar to location 1 and 2, the UAS should always move forward at location 3.

Location 4:

$$\text{IMP}_{\text{DIFF}} = \frac{N_{\min}^{2r}}{4}(5-a) * \frac{c}{s}$$

Therefore, for location 4, the UAS might keep moving forward or turn backward depending on the value of a . If $a < 5$, IMP_{DIFF} is positive and the UAS should keep moving forward in its clockwise direction. On the other hand, if $a \geq 5$ the UAS should turn backward to fly in the counter-clockwise direction.

Thus, the UAS might turn back only when it is travelling from the maximum spread direction (location 3) towards the minimum spread direction (location 1) in the clockwise direction. This is because, IMP_{DIFF} starts to decrease within this segment, thereby increasing the tendency of turning back. Depending on the value of a , the IMP_{DIFF} becomes negative at a specific cell and then the UAS turns back. The turning back makes the UAS to revisit the most active region of the fire, which is what we want.

The smaller the value of a , the larger the distance that the UAS needs to fly to reach such a cell. However, if the value of a is lower than a minimum threshold, the UAS might never find such a cell and will keep flying forward and reach the minimum spread direction (location 1). This means that, if the heterogeneity of fire spreading is not significant enough, the proposed algorithm will work just like a basic circling algorithm. This is also desired because circling would be the best strategy if different parts of the fire perimeter spread more or less uniformly.

5.2 Qualitative analysis

In this section, we qualitatively examine how the two important properties of the fire perimeter, $t_{\text{sinceVisit}}$ and t_{toCell} , impact the UAS's flying direction. These analyses are based on the incremental fire spread scenario described in the preceding section. In the following description, we refer to a

segment of the fire perimeter starting from location A and ending at location B as SEG_{AB} .

Impact of $t_{\text{sinceVisit}}$ on UAS's flying direction:

In (4), we can see how the ROS and $t_{\text{sinceVisit}}$ values participate in the cell importance calculation (the first term of the equation). For each cell in the front and back window, these two values are multiplied and a portion of cell importance is obtained. When the UAS is at location 1, the back window consists of SEG_{14} and SEG_{43} , the front window consists of SEG_{12} and SEG_{23} . We can see that the back and front window have the same set of ROS values. However, $t_{\text{sinceVisit}}$ values in the front window are always higher than the back window, as shown in Fig. 7(a). Thus, the UAS will move forward at this location because the total importance in the front window will be higher due to the higher $t_{\text{sinceVisit}}$ values. As the UAS keeps moving forward, it reaches location 2 and then location 3 because the total importance in the front window remains larger at these locations. However, when the UAS reaches location 4, the front window consists of SEG_{41} and SEG_{12} , which has lower ROS values compared to the segments of the back window. Therefore, at location 4, ROS values of the back window pull the UAS to turn back, while the $t_{\text{sinceVisit}}$ values of the front window push the UAS to move forward. If the ROS values at the back window are large enough to dominate the higher $t_{\text{sinceVisit}}$ values at the front window, the UAS will turn back. Otherwise, the UAS will keep flying towards location 1.

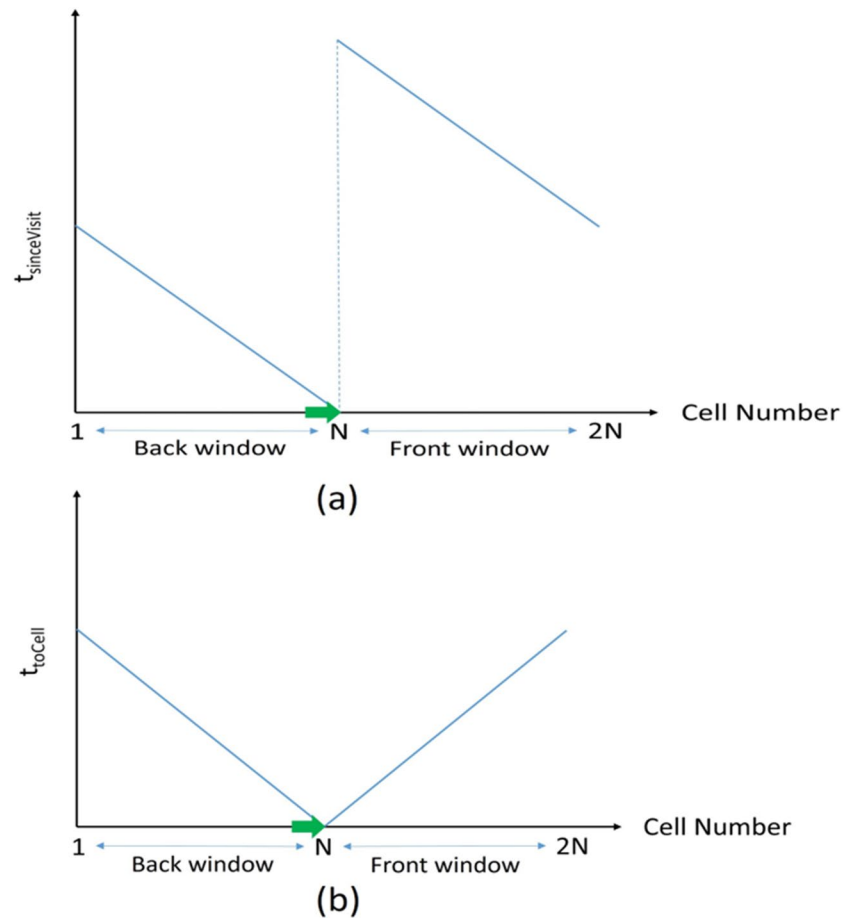
From this analysis, we conclude that the first term of (4) is responsible for pushing the UAS forward along the fire perimeter.

Impact of t_{toCell} on UAS's flying direction:

The t_{toCell} parameter indicates the time needed to reach a perimeter cell from the current location of the UAS. A typical pattern of the t_{toCell} values has been shown in Fig. 7(b). During the cell importance calculation based on (4), t_{toCell} is multiplied with the ROS value and another portion of the cell importance is obtained. Below we discuss how the UAS's flying behavior is influenced by only this portion of the cell importance.

When the UAS is at location 1, the front and back window has the same set of ROS and t_{toCell} values. Hence the importance of the front and back window will be the same and the UAS will move forward. As the UAS moves towards location 2, the importance in the front window will increase because it will acquire cells with higher ROS values. This is also true when the UAS keeps moving from location 2 to location 3. However, when the UAS moves one cell ahead of location 3, the importance in the back window will become larger as it releases a cell with a lower ROS value (the cell next to r_{\min} direction) and acquires a new cell with a higher ROS value (cell

Fig. 7 Pattern of $t_{\text{sinceVisit}}$ and t_{toCell} values along the fire perimeter



next to r_{max} direction). Therefore, the UAS will turn back and come to location 3 again in the counterclockwise direction. Now, when it moves one cell ahead of location 3 in the counterclockwise direction, importance in the back window will become higher again and it will turn back again. Thus, the UAS will keep oscillating around the r_{max} direction. To conclude, in our formula for determining the cell importance (4), the second term of the equation is responsible for keeping the UAS on the most active region.

In summary, these two parameters ($t_{\text{sinceVisit}}$ and t_{toCell}), together with the heterogeneous ROS values of different cells, make the UAS have the capability of both revisiting the most active fire region and covering the entire fire perimeter from time to time.

6 Experiment results

The experiments are based on simulated wildfires; we used DEVS-FIRE model [19] to simulate different fire spreading scenarios. In this section, we will present: i) a brief description of wildfire spread simulation for our experiments ii) an in-depth demonstration of how the algorithm works iii)

demonstration of path planning for non-uniform fuel models iv) demonstration of path planning for non-uniform wind conditions v) comparison of the algorithm with a baseline method.

6.1 Wildfire spread simulation

Wildfire spread simulation is used to test and demonstrate the proposed path planning algorithm. The wildfire spread simulation uses the DEVS-FIRE model [19], which is a discrete event simulation model developed based on the Discrete Event System Specification (DEVS) formalism. DEVS-FIRE uses a cellular space to represent a wildland area, where each cell has its own terrain and fuel (vegetation) data corresponding to the sub-regions in the area. All cells are coupled to a weather model to receive weather data (wind speed and wind direction) dynamically. Thus, the cellular space model incorporates spatial fuel data, terrain data, and temporal weather data into the simulation of wildfire behavior across both time and space. Fire spreading is modeled as a propagation process as burning cells ignite their unburned neighboring cells. Once a cell is ignited, it uses Rothaermel's model [20] to compute the fire spread rate and direction within the cell. This model has been extensively used

and tested by researchers and practitioners and is proven to be very robust. When implementing the UAS' path planning in the wildfire spread simulation-based experiment, the UAS is modeled as a mobile agent flying along the fire perimeter. The UAS agent obtains information from the cellular space of DEVS-FIRE. It dynamically constructs an on-board map of the fire perimeter and makes decisions about its flying direction based on the on-board data.

A sample fire spread scenario using DEVS-FIRE has been shown in Fig. 8. The green regions represent the part of the land that is not impacted by the fire yet. The red region represents the currently burning area and the black region represents the area that has already burnt out. To describe the fire spread and path planning results, we specify the directions of the cell space as shown in Fig. 8. In this sample scenario, the fire is mostly spreading towards the west and south-west direction since it started from the ignition point.

For the experiments, the wildland has been modeled as a 200×200 cell space using DEVS-FIRE. A specific cell is ignited in the beginning and starting from there, the fire spreads as per the fire spread model and weather configuration. The UAS is deployed on the fire perimeter 1 hour after the fire has started.

6.2 Demonstration of real-time autonomous path planning algorithm

First, we are going to demonstrate the path planning result in detail through a simulated fire spread scenario. In this scenario, as shown in Fig. 9, the fire starts from the northeast side of the cell space and spreads towards the southwest direction over time. During the first 3 hour of the simulation, the fire was not spreading very fast, as a result, the UAS did not make too many revisits during that



Fig. 9 UAS trajectory based on the proposed algorithm

period. However, for the remaining simulation time, the fire was spreading significantly faster. During that time, the UAS made frequent back and forth revisits to cover the most spreading part of the fire towards the southwest direction.

Figure 10(c) presents a snapshot from the simulation where the UAS is moving towards the most active fire region; the spread in the back of the UAS is very slow. The corresponding importance values of the constructed fire perimeter cells have been plotted in Fig. 10(a). From the plot, we can see that the total importance in the back region is significantly smaller than in the front region. Therefore, it is better for the UAS to keep flying in the same direction in this scenario. In contrast, Fig. 10(d) presents a snapshot where the fire is spreading much faster behind the UAS. The importance of the constructed fire perimeter cells has been plotted in Fig. 10(b). In this case, the total importance in the back region has just got larger than the total importance in the front region. Hence, the UAS needs to change its flying direction to cover the most active region of the fire. In this experiment the value of the heterogeneity-adjusting factor (α) is 1.0.

Figure 11 compares the UAS trajectories for six different α values for the same fire spread scenario presented in Fig. 9. When the value of α is smaller than 1, the UAS changes its flying direction less frequently because the fire spread is less heterogeneous in this case. In contrast, when the value of α is larger than 1, the UAS changes its flying direction more frequently to put more monitoring attention on the most actively spreading region of the fire. As the value of α increases, this tendency of changing the flying direction also increases. With 0.0, 0.8, 1.0, 1.2, 1.5 and 5.0 as the values of α , the UAS changed its flying direction 0, 8, 11, 15, 20,

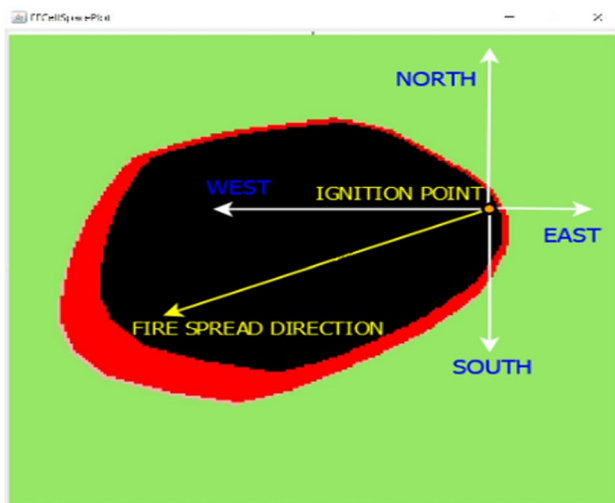
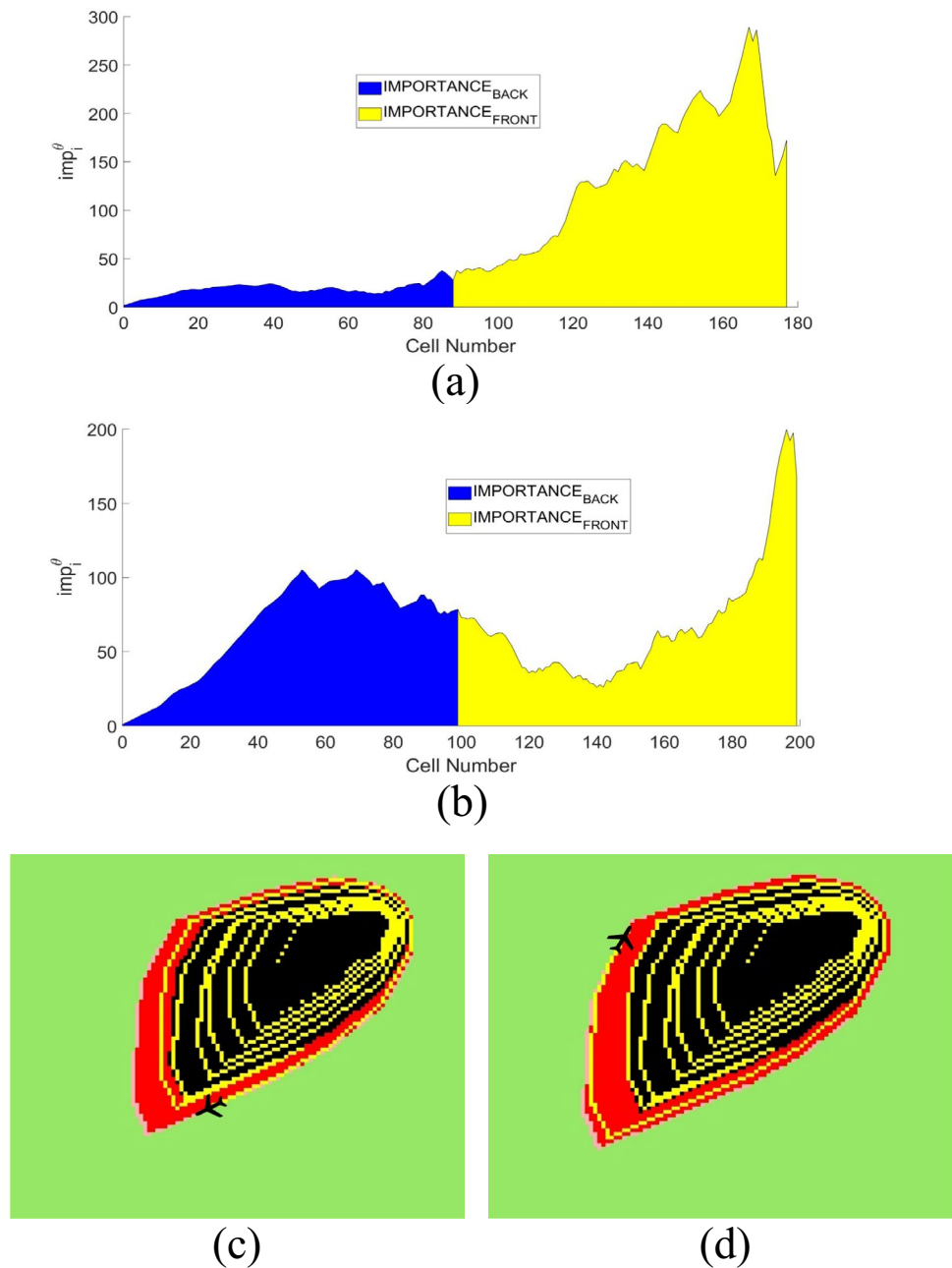


Fig. 8 Wildfire spread simulation using DEVS-FIRE

Fig. 10 Illustration of back and front window importance



and 43 times respectively as shown in Fig. 11. It should be noted that the UAS made no direction change when α is very small (e.g., 0.0). In contrast, the UAS made a large number of direction changes when α is much higher than 1.0 (e.g., 5.0). Thus, the sensitivity of the UAS's flying direction change can be controlled by adjusting the value of α . This feature greatly improves the flexibility of the algorithm as one can make the UAS to focus on the—i) entire perimeter of the fire by setting a very small value of α ii) most active fire front by setting a higher value of α . iii) both the entire perimeter and the most active fire front by setting an intermediate value of α .

6.3 Path planning for non-uniform fuel loading

In this section, we consider fire spread in two different types of fuel combinations, thus two different spreading characteristics. In the first type of fuel combination, we consider fuel with relatively slow-burning characteristics followed by fuel with relatively fast-burning characteristics. We consider one half of the cell space consists of young brush which has moderate burning characteristics, and another half of the cell space consists of short grass which has faster burning characteristics. Figure 12(a) shows the UAS trajectory while monitoring a fire in such a fuel

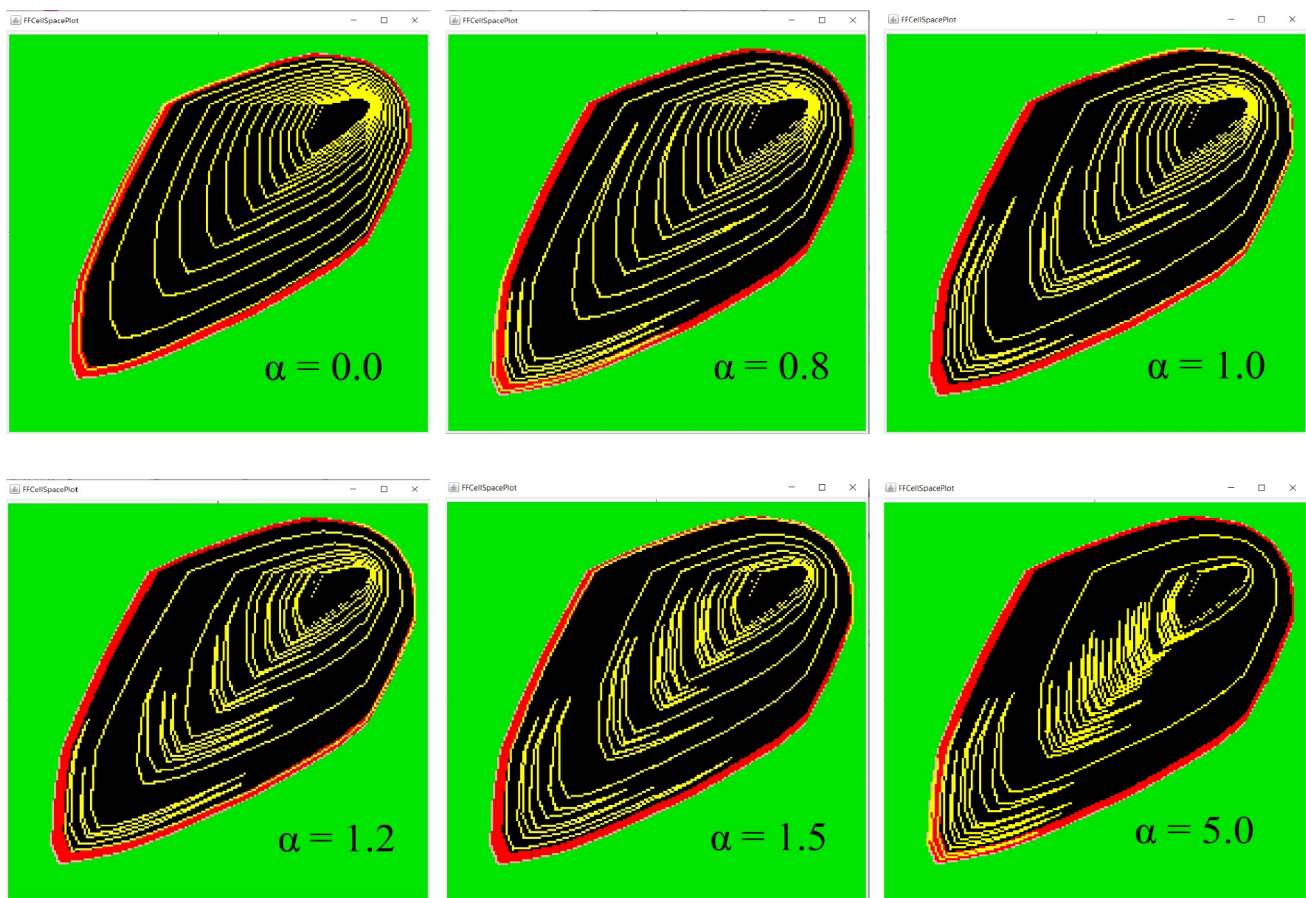


Fig. 11 UAS trajectory for different α values

combination. From the trajectory, we can see that the UAS circled the fire while it spread on the upper half of the cell space because the spreading speed was relatively slow. However, when the fire spread much faster on the second half of the cell space, the UAS made back-and-forth visits over the most actively spreading fire front. Thus, the proposed algorithm captured the fire spreading speed difference in different fuel types and adjusted its monitoring strategy accordingly.

In the second type of fuel combination, we consider fuel with relatively fast-burning characteristic followed by fuel with slow-burning characteristic. Here, we consider one half of the cell space consists of young brush (Fuel Model 5) and another half consists of short needle timber litter (Fuel Model 8) which has slow-burning characteristics. Figure 12(b) shows the UAS trajectory for this type of fuel combination based on the proposed algorithm. The fire front was spreading much faster on the upper half of the cell space, consequently, the UAS made back-and-forth visits over the fire front. However, when the fire front reached the lower half of the cell space, the fire spreading speed is significantly reduced due to the slow-burning fuel type. Consequently, the

UAS did not make any further back-and-forth visits, instead, it monitored the fire by circling the perimeter.

For both types of fuel combinations considered in this section, the UAS automatically adjusted its monitoring strategy based on the fire spreading speed. This capability helped the UAS to obtain less monitoring gap on the faster spreading part of a fire. Thus, the proposed algorithm works adaptively for non-uniform fuel combinations.

6.4 Path planning for different wind conditions

We consider three different fire spreading behaviors resulting from three different wind conditions and evaluate how the proposed algorithm performs on those fire spread scenarios. Here, we consider uniform fuel loading for the entire cell space. First, we consider a scenario where fire spreads very fast towards a specific direction as shown in Fig. 13(a). Here, the fire started from the northeast corner of the cell space and spread towards the southwest direction much faster. As a specific region of this fire is spreading very fast, the UAS revisits that fire front very frequently to put more monitoring attention. Second, we consider a scenario where the fire

Fig. 12 UAS trajectory based on non-uniform fuel loadings

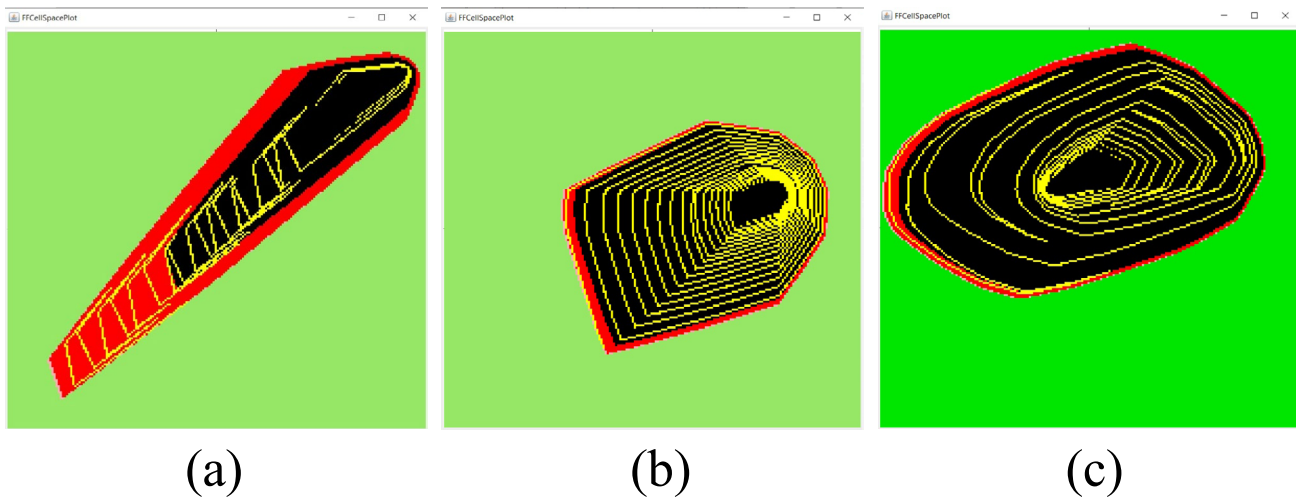
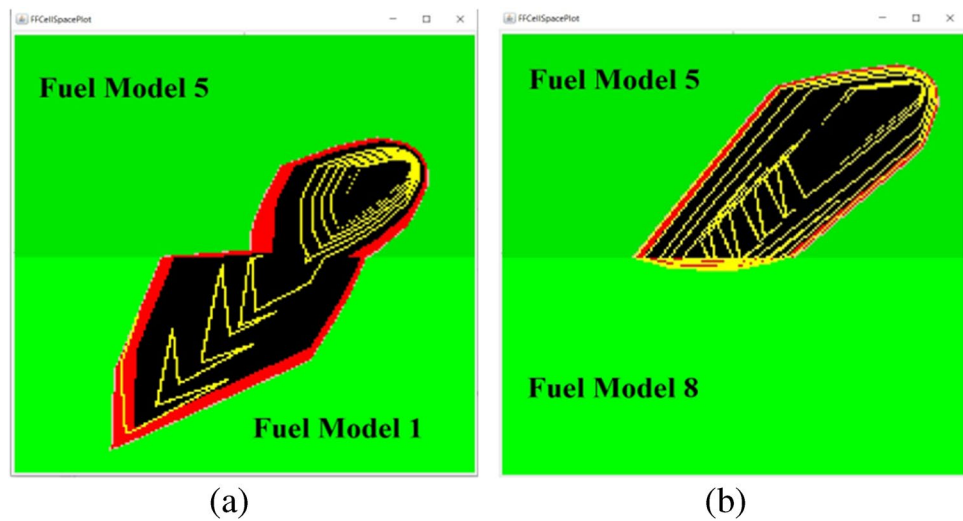


Fig. 13 UAS trajectory in different wind conditions

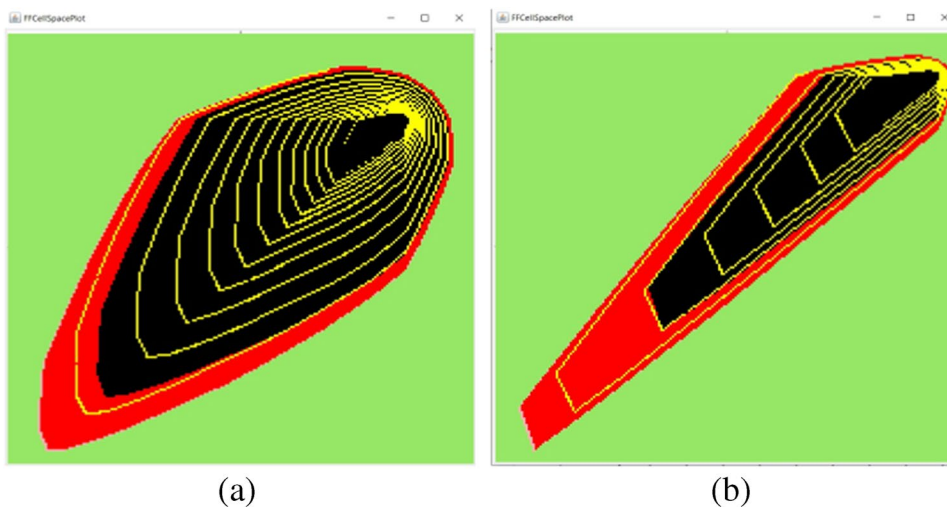
is spreading slowly around the ignition point as shown in Fig. 13(b). There is no significantly fast-spreading region in this scenario, therefore IMP_{BACK} has never grown larger than IMP_{FRONT} . Consequently, the UAS always moved forward cyclically while monitoring this slowly spreading fire. Third, we consider a fire spreading scenario where the wind significantly changes its direction during the fire spread. In this experiment, the wind blows towards the north during the first three hours of simulation and then changes its direction towards the south for the remaining three hours. Consequently, the fire spread more actively on the north side during the first three hours and on the south side during the last three hours. From the UAS trajectory showed in Fig. 13(c), we can see that the UAS made back-and-forth visits on the north side initially and later on the south side of the fire. Thus, the proposed algorithm is able to shift its monitoring attention depending on the location of the most active region.

In summary, the proposed algorithm is robust towards different fire spread behaviors resulting from different wind conditions. When the fire spread is slow, the proposed algorithm works like a basic circling algorithm. In contrast, when a region is spreading significantly faster, the UAS puts more monitoring attention on the most active region of the fire. Furthermore, the proposed algorithm is capable of shifting its monitoring attention in case the most active region gets shifted due to non-uniform wind conditions.

6.5 Comparison with a baseline method

One of the baseline approaches for monitoring a wildfire is just circling the perimeter for the entire time. This method is straightforward and doesn't consider the uneven importance of the fire perimeter. This approach makes sense when the fire is spreading slowly and evenly at all angles around the fire center.

Fig. 14 UAS trajectory based on the circling-based approach



However, if a section of the fire is spreading much faster, this approach is unable to provide higher monitoring attention to that region. Figure 14(a) and (b) show the UAS trajectories based on the circling-based approach corresponding to the fire spread scenarios used in Figs. 9 and 13(a) respectively. For both scenarios, we can notice larger monitoring gaps on the faster-spreading southwest side of the fire compared to the proposed algorithm.

While the UAS is visiting a section of the perimeter, the fire is possibly spreading in other sections with different spreading speeds. Therefore, the UAS lacks knowledge about those unvisited parts until it visits those sections again. This knowledge gap results in inaccuracies between the actual fire shape and fire shape constructed from UAS data. To quantify this inaccuracy for both the circling-based method and the proposed algorithm, we measure the difference between the original fire shape and the constructed fire shapes at each of the 360 discrete angles. We name this difference as “distance error” in this paper. Figure 15 illustrates the concept of distance

error for a sample fire monitoring scenario. The original fire shape has been highlighted by the red region and the fire shape constructed from the UAS trajectory has been marked by the blue region. To construct fire shape from UAS trajectory, first, we used the convex hull algorithm [21] to construct the convex hull (the smallest convex set containing a set of points) of the on-board perimeter cells. Then, we have filled in that convex hull to get constructed fire shape. Between these two shapes, a different amount of distance error is present in different parts of the fire as shown by the yellow arrows in Fig. 15.

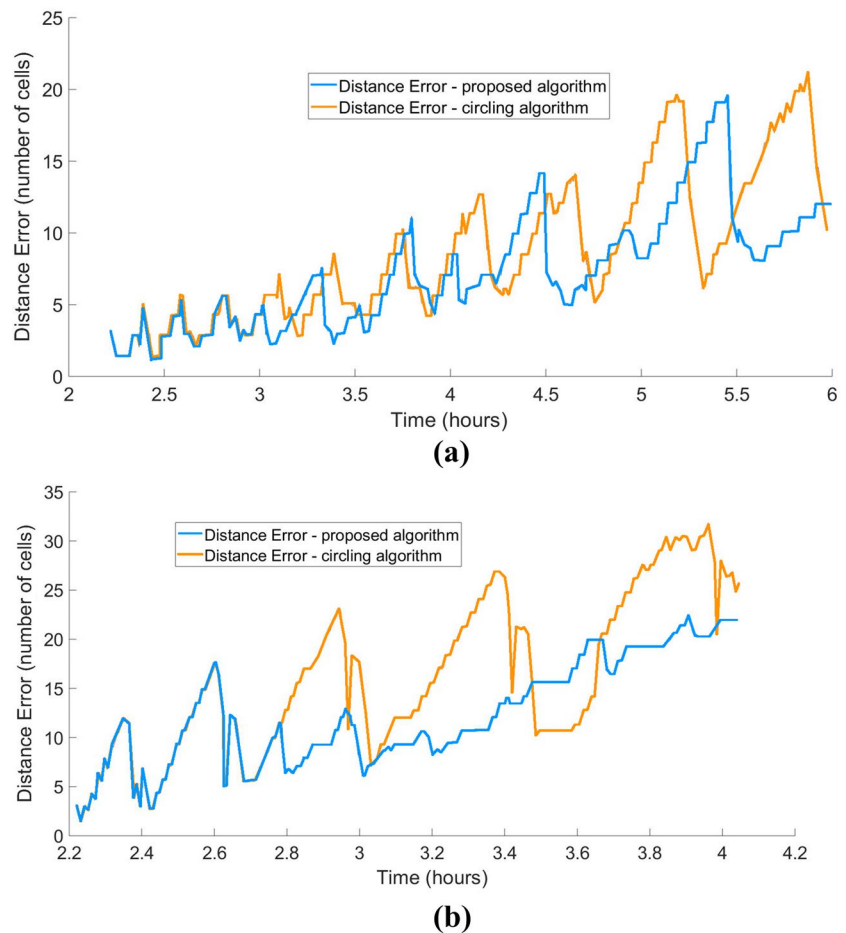
One of the major goals of UAS based wildfire monitoring is to construct a fire shape from the collected data in real-time as accurately as possible. The maximum distance error for a more accurate fire shape will be smaller compared to a less accurate fire shape. To quantitatively compare the proposed algorithm with the circling-based method, we have recorded this maximum distance error during the simulation whenever the UAS moved to a new cell. Figure 16(a) compares the maximum distance error of these two approaches for the fire spread scenario shown in Fig. 14(a). From the comparison, we can see that the maximum distance error is similar for the first 3 h of the simulation as the fire was spreading slowly during that time. However, when the fire started to spread faster after that time period, the difference between the average maximum distance error starts to increase. At the end of the simulation, the average maximum distance error for the proposed algorithm and the circling-based method is 7.3 and 8.5 respectively. The lower average distance error implies that the fire shape constructed from the real-time UAS data is more accurate.

Figure 16(b) compares the maximum distance error for the fire spread scenario shown in Fig. 14(b). In this scenario, the fire was spreading very fast towards the southwest direction from the very beginning. From the comparison, we can see that the circling-based approach has much larger maximum distance errors for this fast-spreading scenario. This is



Fig. 15 Distance error between the original fire shape and constructed fire shape

Fig. 16 Comparison of the proposed algorithm and circling based approach



because, when the UAS was visiting the slower spreading northeast side of the fire to make full cycles, the fire on the southwest side was spreading very fast, which resulted in higher inaccuracy for the circling-based method. In contrast, the proposed algorithm was making frequent revisits to the faster-spreading southwest side, which resulted in smaller maximum distance errors. When the simulation is finished, the average maximum distance error for the proposed algorithm and the circling-based method is 12.2 and 15.8 respectively. Therefore, for a faster spreading fire scenario, the proposed algorithm is capable to construct the fire shape with a larger accuracy margin compared to the baseline approach.

7 Conclusion and future work

In this work, we presented a new approach for real-time autonomous path planning intended to be used for UAS based wildfire monitoring. Real-time and autonomous path

planning can be a very useful mechanism in challenging scenarios like lack of a ground system, unknown environment, lack of information about the land and weather, and many more. The proposed algorithm takes the uneven nature of wildfire spread into account and puts more monitoring attention for faster spreading segments of the fire. Simulation results for a variety of fire spread scenarios have been presented to show the effectiveness and adaptiveness of the proposed algorithm. In addition, an analysis of the algorithm has been presented which provides some important insights about the proposed algorithm. In the future, we want to take more complex fire spread factors (such as the spatiotemporal uncertainty of fire spread) into account and enhance the algorithm to make it more robust. In addition, we want to extend the algorithm to support multiple UASs. To conclude, the path planning algorithm presented in this paper can perform more advanced wildfire monitoring tasks and holds the potential to enhance wildfire management strategies.

Appendix A

Location 1:

$$IMP_{BACK} = (N_{min}^{2r} + Nr_{min} + \frac{\delta N^3}{3} + \frac{\delta N^2}{2} + \frac{\delta N}{6}) * \frac{c}{s}$$

$$IMP_{FRONT} = (2N_{min}^{2r} + \delta N^3 + \delta N^2) * \frac{c}{s}$$

Location 2:

$$IMP_{BACK} = \left[N_{min}^{2r} + ar_{min} \frac{N(N+2)}{4} - r_{min} \frac{N(N+2)}{4} \right] * \frac{c}{s}$$

$$IMP_{FRONT} = [N_{min}^{2r} + ar_{min} \frac{N(N+2)}{4} - r_{min} \frac{N(N+2)}{4} + \frac{aN_{min}^{2r}}{2} - \frac{aN_{min}^{2r}}{2}] * \frac{c}{s}$$

Location 3:

$$IMP_{BACK} = (\sum_{i=1}^N 2Nr_{min} + \sum_{i=1}^{N^2} \delta N - 2r_{min} \sum_{i=1}^N i - 2\delta \sum_{i=1}^N i^2) * \frac{c}{s}$$

$$IMP_{FRONT} = (\sum_{i=1}^N 2Nr_{min} + \sum_{i=1}^N 2i\delta N) * \frac{c}{s}$$

Location 4:

$$IMP_{BACK} = \left[\frac{N_{min}^{2r}}{4}(3a+1) + \frac{Nr_{min}}{2}(a-1) \right] * \frac{c}{s}$$

$$IMP_{FRONT} = \left[\frac{N_{min}^{2r}}{4}(2a+6) + \frac{Nr_{min}}{2}(a-1) \right] * \frac{c}{s}$$

Acknowledgements A preliminary version of this algorithm was presented at the 2021 International Conference on Unmanned Aircraft Systems (ICUAS '21), Athens, Greece, June 15-18, 2021.

Author Contribution S M Towhidul Islam methodology, software, formal analysis, writing original draft, visualization. Xiaolin Hu conceptualization, validation, writing – review and editing, supervision.

Funding This work is supported by the US Department of Agriculture (USDA) National Institute of Food and Agriculture (NIFA) under grant number 2019–67021-29011.

Data Availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests The authors report no declarations of interests.

References

1. A. L. Westerling. Increasing western us forest wildfire activity: sensitivity to changes in the timing of spring. *Philos Trans R Soc Lond B Biol Sci.* 371:20150178, 2016. Available at <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4874415/>.
2. Bogdos N, Manolakos ES (2019) Crowd-sourced wildfire spread prediction with remote georeferencing using smartphones. *IEEE Access* 7:102102–102112. <https://doi.org/10.1109/ACCESS.2019.2931456>
3. X Hu, J Bent and J Sun, Wildfire monitoring with uneven importance using multiple unmanned aircraft systems, 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 2019, pp 1270–1279, <https://doi.org/10.1109/ICUAS.2019.8797793>
4. S. M. Towhidul Islam and X. Hu, Real-time on-board path planning for UAS-based wildfire monitoring, 2021 International Conference on Unmanned Aircraft Systems (ICUAS), 2021, pp 527–535, <https://doi.org/10.1109/ICUAS51884.2021.9476725>
5. Cabreira, T.M.; Brisolara, L.B.; Ferreira Jr., P.R. Survey on coverage path planning with unmanned aerial vehicles. *Drones* 2019
6. Nolan P, Paley DA, Kroeger K (2017) Multi-UAS path planning for non-uniform data collection in precision agriculture. *IEEE Aerospace Conference* 2017:1–12. <https://doi.org/10.1109/AERO.2017.7943794>
7. Balampanis F, Maza I, Ollero A (2017) Spiral-like coverage path planning for multiple heterogeneous UAS operating in coastal regions. *International Conference on Unmanned Aircraft Systems (ICUAS) 2017*:617–624. <https://doi.org/10.1109/ICUAS.2017.7991461>
8. S. Kim, W. Lee, Y. Park, H. Lee and Y. Lee, Forest fire monitoring system based on aerial image, 2016 3rd International conference on Information and Communication Technologies for Disaster Management (ICT-DM), Vienna, 2016, pp 1–6, <https://doi.org/10.1109/ICT-DM.2016.7857214>
9. A. Viseras, M. Meissner and J. Marchal, Wildfire front monitoring with multiple UAVs using deep Q-Learning, In: *IEEE Access*, <https://doi.org/10.1109/ACCESS.2021.3055651>.
10. Chen X et al (2022) Wildland fire detection and monitoring using a drone-collected RGB/IR image dataset. *IEEE Access* 10:121301–121317. <https://doi.org/10.1109/ACCESS.2022.3222805>
11. O M Bushnaq, A Chaaban and T. Y. Al-Naffouri, The role of UAV-IoT networks in future wildfire detection, In: *IEEE Internet of Things Journal*, vol 8, no. 23, pp 16984–16999, 1 Dec.1, 2021, <https://doi.org/10.1109/JIOT.2021.3077593>.
12. C. Heracleous, P. Kolios and C. G. Panayiotou, UAV-based system for real-time wildfire perimeter propagation tracking, 2023 31st Mediterranean Conference on Control and Automation (MED), Limassol, Cyprus, 2023, pp 13–18, <https://doi.org/10.1109/MED59994.2023.10185693>.
13. Ullah A et al (2021) A survey on continuous object tracking and boundary detection schemes in IoT assisted wireless sensor networks. *IEEE Access* 9:126324–126336. <https://doi.org/10.1109/ACCESS.2021.3110203>
14. Shaffer JA, Carrillo E, Xu H (2018) Hierarchical application of receding horizon synthesis and dynamic allocation for UAVs fighting fires. *IEEE Access* 6:78868–78880. <https://doi.org/10.1109/ACCESS.2018.2885455>
15. R Bailon-Ruiz, S Lacroix and A Bit-Monnot, Planning to monitor wildfires with a fleet of UAVs, 2018 IEEE/RSJ International conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp 4729–4734, <https://doi.org/10.1109/IROS.2018.8593859>.

16. Shrestha K, Dubey R, Singandhupe A, Louis S, La H, Multi objective UAV network deployment for dynamic fire coverage, (2021) IEEE Congress on Evolutionary Computation (CEC). Kraków, Poland 2021:1280–1287. <https://doi.org/10.1109/CEC45853.2021.9504947>
17. Kappel KS, Cabreira TM, Marins JL et al (2020) Strategies for patrolling missions with multiple UAVs. *J Intell Robot Syst* 99:499–515. <https://doi.org/10.1007/s10846-019-01090-2>
18. F. Afghah, A. Razi, J. Chakareski and J. Ashdown, Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles, IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 2019, pp 835–840, <https://doi.org/10.1109/INFOCOMW.2019.8845309>
19. Hu X, Sun Y, Ntamo L (2012) DEVS-FIRE: design and application of formal discrete event wildfire spread and suppression models. *Simulation* 88(3):259–279
20. Rothermel RC (1972) A mathematical model for predicting fire spread in wildland fuels, Vol. 115. Intermountain Forest & Range Experiment Station, Forest Service, US Department of Agriculture, Ogden. <https://www.fs.usda.gov/treearch/pubs/32533>
21. Preparata FP, Shamos MI (1985) Convex hulls: basic algorithms. In: Computational geometry. Texts and Monographs in Computer Science. Springer, New York. https://doi.org/10.1007/978-1-4612-1098-6_3

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

S M Towhidul Islam is currently a Lecturer at the Department of Computer Science, Georgia State University (GSU), Georgia, USA. Previously, he was a member of the Systems Integrated Modeling and Simulation (SIMS) Lab at GSU, led by Dr. Xiaolin Hu. His research interests include Path Planning, Distributed Computing, Modeling and Simulation, and Autonomous Systems.

Xiaolin Hu is a Professor at the Department of Computer Science, Georgia State University, Georgia, USA. His research interests include modeling and simulation, complex systems science, agent and multi-agent system, and advanced computing in parallel and cloud environments. His work covers both fundamental research and applications of computer modeling and simulation.