

Georgia State University

**ScholarWorks @ Georgia State University**

---

Computer Science Dissertations

Department of Computer Science

---

8-8-2023

## **UAS Path Planning for Dynamical Wildfire Monitoring with Uneven Importance**

S M TOWHIDUL ISLAM

Follow this and additional works at: [https://scholarworks.gsu.edu/cs\\_diss](https://scholarworks.gsu.edu/cs_diss)

---

### **Recommended Citation**

ISLAM, S M TOWHIDUL, "UAS Path Planning for Dynamical Wildfire Monitoring with Uneven Importance." Dissertation, Georgia State University, 2023.  
[https://scholarworks.gsu.edu/cs\\_diss/201](https://scholarworks.gsu.edu/cs_diss/201)

This Dissertation is brought to you for free and open access by the Department of Computer Science at ScholarWorks @ Georgia State University. It has been accepted for inclusion in Computer Science Dissertations by an authorized administrator of ScholarWorks @ Georgia State University. For more information, please contact [scholarworks@gsu.edu](mailto:scholarworks@gsu.edu).

UAS Path Planning for Dynamical Wildfire Monitoring with Uneven Importance

by

S M Towhidul Islam

Under the Direction of Xiaolin Hu, PhD

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2023

## ABSTRACT

Unmanned Aircraft Systems (UASs) offer many benefits in wildfire monitoring when compared to traditional wildfire monitoring technologies. When planning the path of an UAS for wildfire monitoring, it is important to consider the uneven propagation nature of the wildfire because different parts of the fire boundary demand different levels of monitoring attention (importance) based on the propagation speed. In addition, many of the existing works adopt a centralized approach for the path planning of the UASs. However, the use of centralized approaches is often limited in terms of applicability and adaptability. This work focuses on developing decentralized UAS path planning algorithms to autonomously monitor a spreading wildfire considering uneven importance. The algorithms allow the UASs to focus on the most active regions of a wildfire while still covering the entire fire perimeter.

When monitoring a relatively smaller and spatially static fire, a single UAS might be adequate for the task. However, when monitoring a larger wildfire that is evolving dynamically in space and time, efficient and optimized use of multiple UASs is required. Based on this need, we also focus on decentralized and importance-based multi-UAS path planning for wildfire monitoring. The design, implementation, analysis, and simulation results have been discussed in detail for both single-UAS and multi-UAS path planning algorithms. Experiment results show the effectiveness and robustness of the proposed algorithms for dynamic wildfire monitoring.

INDEX WORDS: UAS Path Planning, Wildfire Monitoring, Dynamic Perimeter Surveillance, Decentralized Computing, Simulation

Copyright by  
S M Towhidul Islam  
2023

UAS Path Planning for Dynamical Wildfire Monitoring with Uneven Importance

by

S M Towhidul Islam

Committee Chair:

Xiaolin Hu

Committee:

Armin Mikler

Wei Li

Yichuan Zhao

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

August 2023

## DEDICATION

I dedicate this dissertation to my family, friends, supervisor, and colleagues who have supported me throughout the journey and inspired me to reach my destination.

## ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor Dr. Xiaolin Hu for his guidance and continuous support towards my research. His knowledge and experience guided me to achieve the goals of my research. It has been a great privilege to conduct research under his supervision.

I extend my gratitude to the members of my dissertation committee, Dr. Armin Mikler, Dr. Wei Li, and Dr. Yichuan Zhao. I am grateful to them for their valuable time, support, and guidance towards my dissertation.

## **FUNDING ACKNOWLEDGMENT**

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

## TABLE OF CONTENTS

<b>LIST OF FIGURES</b>	<b>vii</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Wildfire	1
1.2 Technologies in Wildfire Monitoring	2
1.3 Unmanned Aircraft Systems (UASs)	3
1.4 UAS Path Planning	4
<b>2 RELATED WORK</b>	<b>7</b>
2.1 UAS-based Surveillance and Monitoring	7
2.2 UASs in Wildfire Monitoring	9
2.3 Importance-based Path Planning in Wildfire Monitoring	10
2.4 Multi-UAS Coordination in Wildfire Monitoring	11
2.5 Literature Gap and Contributions	13
<b>3 UAS PATH PLANNING CONSIDERING UNEVEN IMPORTANCE</b>	<b>14</b>
3.1 Cell Importance	16
3.2 Importance-based Path Planning	22
3.3 Modulating Path Planning Sensitivity	26
<b>4 THE IMPORTANCE-BASED ON-BOARD PATH PLANNING ALGORITHM FOR SINGLE UAS</b>	<b>29</b>
4.1 Initialization of The On-board Fire Map	29
4.2 Fire Shape Reconstruction	31
4.3 Determining Optimal Flying Direction	33
<b>5 ANALYSIS OF THE SINGLE UAS PATH PLANNING ALGORITHM</b>	<b>34</b>

5.1	Quantitative Analysis . . . . .	34
5.2	Qualitative Analysis . . . . .	41
5.2.1	<i>Impact of <math>t_{sinceVisit}</math> on UAS's Flying Direction</i> . . . . .	41
5.2.2	<i>Impact of <math>t_{toCell}</math> on UAS's Flying Direction</i> . . . . .	43
6	SINGLE UAS PATH PLANNING RESULTS . . . . .	46
6.1	Wildfire Spread Simulation . . . . .	46
6.2	Demonstration of the Algorithm . . . . .	48
6.3	Path Planning for Non-uniform Fuel Loading . . . . .	51
6.4	Path Planning for Different Wind Conditions . . . . .	53
6.5	Comparison with a Baseline Method . . . . .	55
7	DECENTRALIZED AND IMPORTANCE-BASED MULTI-UAS PATH PLANNING FOR DYNAMIC WILDFIRE MONITORING . . . . .	59
7.1	Decentralized Importance-based Multi-UAS Path Planning Algo- rithm: Based on the Extension of the single-UAS Algorithm . . . . .	59
7.1.1	<i>Inter-UAS Communication</i> . . . . .	61
7.1.2	<i>Analyzing The Impact of Fleet Size Change</i> . . . . .	65
7.1.3	<i>Experiment Results</i> . . . . .	66
7.2	Decentralized Importance-based Multi-UAS Path Planning Algo- rithm: A novel Back-and-Forth Coverage Strategy . . . . .	69
7.2.1	<i>Initialization</i> . . . . .	71
7.2.2	<i>Back-and-forth Monitoring</i> . . . . .	72
7.2.3	<i>Handling Special Scenarios</i> . . . . .	75
7.2.4	<i>Multi-UAS collaboration</i> . . . . .	76
7.2.5	<i>Analysis of the Proposed Algorithm</i> . . . . .	78
7.2.6	<i>Experiment Results</i> . . . . .	79
7.3	Comparison Between the two Versions of the Multi-UAS Path Plan- ning Algorithm . . . . .	82
7.3.1	<i>Initialization Procedure</i> . . . . .	82
7.3.2	<i>Rapid Fire Spread on the Back Window</i> . . . . .	83

7.3.3 <i>Fire Peaks Might Not Get Proper Attention</i> . . . . .	83
7.3.4 <i>Computational Efficiency</i> . . . . .	84
8 CONCLUSIONS AND FUTURE WORK . . . . .	85
REFERENCES . . . . .	87

## LIST OF FIGURES

Figure 3.1	Sample Fire Spread Scenario . . . . .	15
Figure 3.2	Angle based trajectory cell tracking . . . . .	17
Figure 3.3	Importance Based Path Planning . . . . .	25
Figure 3.4	Heterogeneity adjustment of the fire spread . . . . .	27
Figure 4.1	On-board fire map . . . . .	30
Figure 4.2	On-board fire perimeter construction . . . . .	32
Figure 5.1	Incremental fire spread scenario . . . . .	34
Figure 5.2	The incremental fire spread model . . . . .	35
Figure 5.3	Pattern of $t_{sinceVisit}$ values along fire perimeter . . . . .	41
Figure 5.4	Pattern of $t_{toCell}$ values along fire perimeter . . . . .	44
Figure 6.1	Wildfire spread simulation using DEVS-FIRE . . . . .	47
Figure 6.2	UAS trajectory based on the proposed algorithm . . . . .	48
Figure 6.3	Illustration of back and front window importance . . . . .	50
Figure 6.4	UAS trajectory for different $\alpha$ values . . . . .	51
Figure 6.5	UAS trajectory based on non-uniform fuel loadings . . . . .	53
Figure 6.6	UAS trajectory in different wind conditions . . . . .	54
Figure 6.7	UAS trajectory based on the circling-based approach . . . . .	55
Figure 6.8	Distance error between the original fire shape and constructed fire shape	57
Figure 6.9	Comparison of the proposed algorithm and circling-based approach .	58
Figure 7.1	Importance-based path planning . . . . .	60
Figure 7.2	Inter-UAS communication strategy . . . . .	63

Figure 7.3	Updating the on-board maps . . . . .	64
Figure 7.4	Possible scenarios for dynamic fleet size change . . . . .	65
Figure 7.5	Impact of dynamic fleet size change . . . . .	65
Figure 7.6	On-board fire shape generation . . . . .	67
Figure 7.7	Demonstration of importance-based decision making . . . . .	68
Figure 7.8	Path planning results for different numbers of UASs . . . . .	69
Figure 7.9	Overview of the algorithm . . . . .	70
Figure 7.10	Initialization of the onboard map . . . . .	72
Figure 7.11	Scanning for the target angle . . . . .	73
Figure 7.12	Updating the skipped angles . . . . .	74
Figure 7.13	Extending the current trip . . . . .	76
Figure 7.14	Inter-UAV collaboration . . . . .	77
Figure 7.15	On-board map initialization . . . . .	79
Figure 7.16	Maintaining up-to-date on-board map . . . . .	79
Figure 7.17	Back and forth perimeter coverage . . . . .	80
Figure 7.18	Path planning results based on the proposed algorithm . . . . .	81

## CHAPTER 1

### INTRODUCTION

Technological advancements are enhancing the safety of human lives and properties in many ways. One of the major safety threats is natural disasters which often cause damages at large scales. Wildfire is one of the most common among them - thousands of wildfires occur every year which destroys millions of acres of land and properties around the world. The impact of this destruction is immense, and it is a great threat to human and wildlife, vegetation, and the ecosystem. Each year, a large number of human and wild lives are impacted and millions of acres of land are destructed by the wildfires. In addition, environmental pollution around the wildfire affected regions increases significantly which might lead to health problems of the local population. In many cases, the economic consequences in the wildfire affected regions are also substantial. 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 to prevent possible damages.

#### 1.1 Wildfire

Wildfire, one of the most dangerous natural disasters, poses a major threat to human and wildlife, and is responsible for the destruction of millions of acres of land and properties around the world (2). Tens of thousands of wildfires occur every year due to both natural activities such as lightning strikes and human activities such as the intentional or accidental start of an uncontrollable fire in the forest (3). To tackle this threat and minimize the amount of destruction, it is important to obtain real-time spread information about the fire

spread. For instance, if firefighters have access to such real-time information, they could deploy fire suppression resources in the right place at the right time. Furthermore, they can prepare a better evacuation plan for themselves and the occupants.

Wildfire spread is a dynamic process influenced by a variety of factors. The three important factors that influence wildfire behavior are fuel, terrain, and weather. The fuel refers to the composite of variables that describe the vegetation the fire is spreading through. The terrain is described by slope and aspect, where slope is the inclination of a land surface and aspect is the direction the surface is facing. The weather has a dynamic influence on wildfire behavior and includes three components: wind speed, wind direction, and moisture content. Because of the dynamic weather, non-uniform terrain, and different fuel loadings, wildfire spreading is a highly heterogeneous process with non-uniform spreading speed and fireline intensity in both space and time. 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, the more active regions of the fire are more important to be monitored and need to be visited more frequently; the less active regions have fewer changes in fire state and thus can be visited less frequently.

## **1.2 Technologies in Wildfire Monitoring**

To support wildfire management, collecting real-time data about the spreading fire is important. In the past, researchers and practitioners used a variety of methods and technologies

for wildfire data collection and monitoring. These include the use of manned aircraft, fire lookout towers, satellite imagery, remote sensors, ground-based sensors and so on. While these methods and technologies offer certain benefits in wildfire monitoring, their use and effectiveness are limited in many scenarios. These limitations include but are not limited to – lower effectiveness, lower safety, higher cost, and limited adaptability to dynamically spreading wildfires. For instance, manned aircraft system-based monitoring is costly and risky to operate. Fire lookout towers have limited coverage, have dependency on a human observer, and could be hazardous for the staff. The satellite images are limited in terms of resolution and accuracy. 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. Thus, a more effective and suitable solution has been sought by researchers in the past decades.

### **1.3 Unmanned Aircraft Systems (UASs)**

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 the wildfire. Furthermore, intelligent and adaptive algorithms can be applied to the UASs so that they can monitor the fire with minimal external supervision and communication. Altogether, they are suitable for a range of wildfire management tasks including but not limited to wildfire

detection, monitoring, and suppression.

For monitoring the wildfires that occur across a large area, it is more beneficial to use multiple UASs instead of a single UAS so that the collected wildfire data is more accurate. However, a firefighting team might have a limited number of resources (UASs) and it is important to make the best use of them to collect the most useful information about the wildfire. The location of the fire perimeter and the spreading speed across different parts of the perimeter are two of the most useful data that can be collected from a spreading wildfire. For this reason, many of the existing works in the literature simplify the wildfire monitoring task to the wildfire perimeter monitoring task. This simplification greatly enhances the monitoring capability of the UASs.

#### **1.4 UAS Path Planning**

UAS path planning is an important aspect of UAS-based wildfire monitoring that faces several unique challenges. First and foremost, the UAS lacks full knowledge about 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. Third, in many of the existing works, a ground station is considered responsible for the control of the UAV. However, it is not always feasible to have a ground station available due to various reasons such as inaccessible terrain, safety concerns for the ground staff, higher setup costs, and so on. Thus, there is a high need for UAVs with decentralized path planning capability. Fourth, more active segments of the fire perimeter demand more frequent visits by UAVs compared to the segments that are spreading slowly. Thus, it is important to consider this property of wildfire propagation to optimize the use of limited resources.

Many of the existing applications of multi-UAS based wildfire monitoring adopt a centralized approach where a central system (e.g. a ground station) performs the path planning and communicates the decisions to the UASs. In addition, the ground station is assumed to have the required information to perform the path planning task such as information about the terrain, fuel, and weather. While this strategy could be favorable in some scenarios, it is not always feasible to have a ground station available with all those information. In this work, we aim to develop decentralized UAS path planning algorithms where the UASs will have no dependency on a ground station. Furthermore, the uneven importance of the fire perimeter will be taken into consideration so that more active perimeter segments get more monitoring attention from the UASs.

Motivated by the need of importance-based and decentralized UAV path planning, the algorithms developed in this work dynamically construct the perimeter of a wildfire based on real-time collected data and then use it to perform the fire monitoring. This work addresses

both the single-UAS and multi-UAS path planning aspects. The single UAS path planning focuses on determining the UAS's best flying direction (clockwise or counterclockwise) along the fire perimeter to effectively monitor a spreading wildfire. A heterogeneity-adjusting factor has been used for modulating the tendency of staying at the active fire regions.

For multi-UAS path planning, we have presented two different algorithms. The first algorithm is based on the extension of the single-UAS path planning algorithm. The second version of the multi-UAS path planning algorithm employs a novel back-and-forth coverage strategy to monitor the wildfire. The remainder of the dissertation is organized as: Chapter 2 describes the related works in the literature. Chapter 3 describes the conceptual design of the UAS path planning considering uneven importance. Chapter 4 describes the details of the single-UAS path planning algorithm. Chapter 5 presents an in-depth analysis of the single-UAS algorithm from several important perspectives. Chapter 6 shows the single-UAS path planning results for different simulated fire scenarios and presents the comparison results of the algorithm with a baseline method. Chapter 7 presents two different multi-UAS path planning algorithms that have been developed based on different path planning principles. In addition, it shows the experiment results and performance analysis of those algorithms. Finally, Chapter 8 discusses the conclusion of the work and possible future works.

## CHAPTER 2

### RELATED WORK

Path planning is one of the important concerns in surveillance and monitoring tasks based on Unmanned Systems. Many of the works in the literature addressed this concern and proposed different solutions in the context of different monitoring tasks. Depending on the type of the monitoring task, the area of interest varies from task to task. In addition, the unmanned system to be used (e.g., robots, UASs, UAVs, UGVs, etc.) for the monitoring also differs depending on the type of the monitoring task. Please note that Unmanned Aircraft Systems (UASs) and Unmanned Aerial Vehicles(UAVs) are often used interchangeably in the literature. Accordingly, we use both terms in our work.

#### 2.1 UAS-based Surveillance and Monitoring

Generally, UAVs are regarded as a very suitable option for aerial surveillance tasks. The work of (4) 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. The work of (5) presents a taxonomy of various cooperative path planning problems and aims to classify the current research. A comprehensive list of existing works has been classified in terms of three factors – task types, UAV group,

and environmental diversity. Only a few of these works focus on the perimeter monitoring problem considering uneven propagation of the perimeter and/or decentralized computation.

An 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 the 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 the 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.

An application of full area coverage using multiple robots has been demonstrated in (8) where a given area is partitioned into smaller areas and then path planning is carried out for those smaller subareas. In many monitoring tasks, the area of interest is often the boundary of the area instead of the entire area (9; 10; 11). This is common in applications where the most useful information about the region to be monitored is the location of the perimeter, and there are a limited number of resources available to monitor the region. The work of (9) aims to distribute a team of robots to monitor a dynamic perimeter such that they keep an equal distance by communicating with their immediate neighbors. In

(10), the authors presented an interpolation method to define the boundary and proposed a navigation strategy to circumnavigate the boundary using robots. Though many of the perimeter monitoring works consider the time-varying property of the perimeter, these rarely consider its importance-varying property.

## 2.2 UASs in Wildfire Monitoring

To address the problem of wildfire monitoring, different technologies have been considered by the researchers including the use of on-board infrared camera images (12; 13), use of satellite images (14), use of on-board camera images (15), use of the UASs' location data (16) and so on. As the technical capabilities of UASs increased over the years, UAS-based wildfire monitoring has got more and more attention from researchers. Many of the recently proposed wildfire monitoring systems consider the use of UAVs in their solution. One of the major concerns 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.

Several previous works demonstrated the use of UASs in the context of wildfire monitoring. A forest fire perception system has been presented in (17) that uses a fleet of aerial vehicles. That system integrates information from several aerial vehicles to estimate real-time fire spread. The proposed system has been tested on controlled fire experiments and it shows the potential of using UASs in wildfire monitoring. Here, a significant part

of the computation is performed by two central systems - Central Perception System and Central Decision System. A deep learning-based forest fire monitoring system has been presented in (18). 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. Based on the past dataset, a centralized system recognizes wildfires. An UAV-based system that is capable of wildfire monitoring and wildfire data collection has been proposed in (19).

### **2.3 Importance-based Path Planning in Wildfire Monitoring**

Though some previous works addressed the problem of fire perimeter monitoring, very few of them investigated the problem considering the uneven importance of the fire perimeter. The work of (20) considered the use of autonomous quadrotor helicopters for forest fire surveillance. The fire perimeter is divided into equidistant portions based on the coordination of waypoints between the quadrotors. The work of (12; 13) studied the path planning problem for fire perimeter surveillance using multiple UAVs. The spatial heterogeneity of wildfire spread was not taken into account in this work, it assumed a homogeneous fire perimeter that needs to be monitored. A fire hotspot assignment scheme-based strategy was presented in (21) where multiple UASs participate to cooperatively monitor multiple forest fires in a region.

Importance-based path planning in wildfire perimeter monitoring is still a fresh concept and only a handful of works focus on this direction. The work of (16) addresses the problem

of UAV coordination for wildfire monitoring considering the uneven importance of visiting different parts of the fire perimeter. Here, the wildfire monitoring problem has been defined as a fire perimeter coverage problem. The goal of this work 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 that has been used here is to treat different parts of the fire 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 adopts a centralized approach and the necessary information for monitoring the fire is provided by a ground station. The research of (22) presents a wildfire monitoring system based on fixed-wing UAVs. This work considers the dynamic growth of the fire and the impact of wind on the motion of the UAV.

## 2.4 Multi-UAS Coordination in Wildfire Monitoring

Several of the previous works (23; 24; 25) focused on cooperative UAV path planning in different application scenarios. The authors of (23) considered the multi-UAS path planning problem as a multi-agent cooperation problem in a dynamic environment. In (26), the authors propose an algorithm to monitor an elliptical fire perimeter by distributing a team of UAVs evenly along the perimeter. The elliptical fire perimeter is generated by a ground station based on the UAV's sensory data. A distributed framework for UAV control has been proposed in (25) to coordinate a team of UAVs to support human-robot based

firefighting. For the patrolling problem with multiple UAVs, four different strategies have been discussed in (27) considering time, uncertainty, and communication. In this work, the authors presented an improved version of their previously developed centralized algorithm.

A small number of the existing works focus particular attention on wildfire perimeter surveillance using a group of UAVs. The work of (28) concentrates on dynamic wildfire boundary monitoring and proposes vector field-based solutions to guide the UAVs along the boundary. In (29), the authors present a decentralized solution for wildfire monitoring considering dynamic fire propagation. The authors of (30) proposed a distributed control framework for UAVs to monitor a spreading wildfire. A two-layer UAV coordination framework has been developed in (31) to support decentralized multi-UAV path planning. To achieve coordination, the UAVs negotiate their flight time using a consensus protocol.

Regarding the computing strategy, many applications use a centralized approach where a ground station performs most of the computations and communicates only the results to the monitoring agents. The use of the centralized approaches is often limited in terms of applicability and adaptability, consequently, some of the applications considered a decentralized approach for the path planning. In (32), the authors presented a decentralized solution to the cooperative perimeter surveillance problem using a team of UAVs based on information sharing. To effectively support decentralized path planning, sharing of some local information is found to be common in many applications, such as the work of (9). In (33), the authors presented a generalized decision-making framework to address the heterogeneous real-world domains. The authors emphasized that collaborative decision making offers more

benefits compared to the benefits provided by local decision making.

A distributed leader-follower coalition formation model has been presented in (34), 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 comprehensively support importance-based identification and monitoring of the most actively spreading region. Thus, a more sophisticated approach is required for the identification and monitoring of the most active fire region while supporting decentralized computation.

## 2.5 Literature Gap and Contributions

Given the most recent advancements of using UASs in wildfire monitoring, there is a high need for decentralized path planning algorithms that focus more closely on the most active region of wildfires. This work presents novel decentralized path planning algorithms that focus on single-UAS and multi-UAS path planning for wildfire monitoring considering the uneven fire propagation. The major contributions of this work include: i) designing the wildfire monitoring task as a perimeter coverage problem considering uneven importance ii) development of a real-time on-board single-UAS path planning algorithm iii) An in-depth analysis of the single-UAS path planning algorithm iv) designing a decentralized multi-UAS path planning and coordination framework for wildfire monitoring v) development of two novel multi-UAS path planning algorithms based on that framework.

### CHAPTER 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 for the fire managers. In this work, the proposed single-UAS path planning strategy focuses on the burning perimeter of the fire, which is consistent with many of the 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. 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 with 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 is spreading more actively demands more monitoring attention compared to a segment that is burning slowly. This work focuses on this path planning 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. 3.1. The red region represents the fire region, the white dot 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 regions 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.

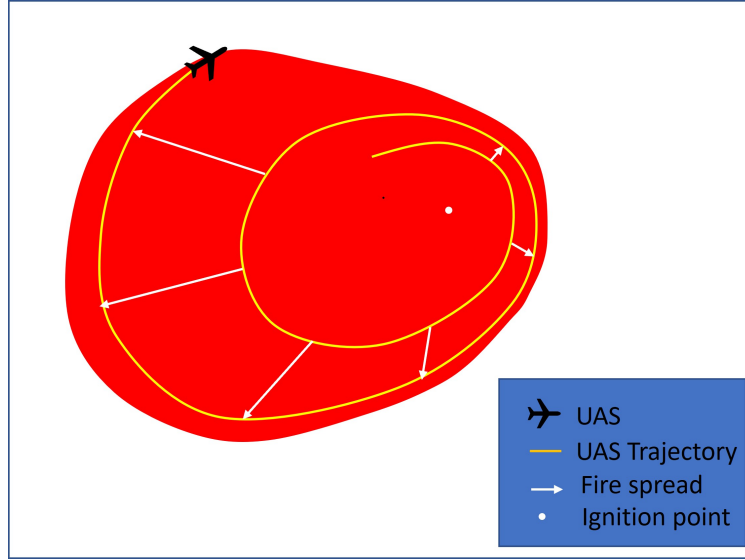


Figure 3.1: Sample Fire Spread Scenario

We represent the fire area as a 2D space and discretize it into a cell space (35), 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 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. The UAS calculates the importance of visiting each perimeter cell, denoted as cell importance, and utilizes those cell importance values for its decision making. 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 in 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 the 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.

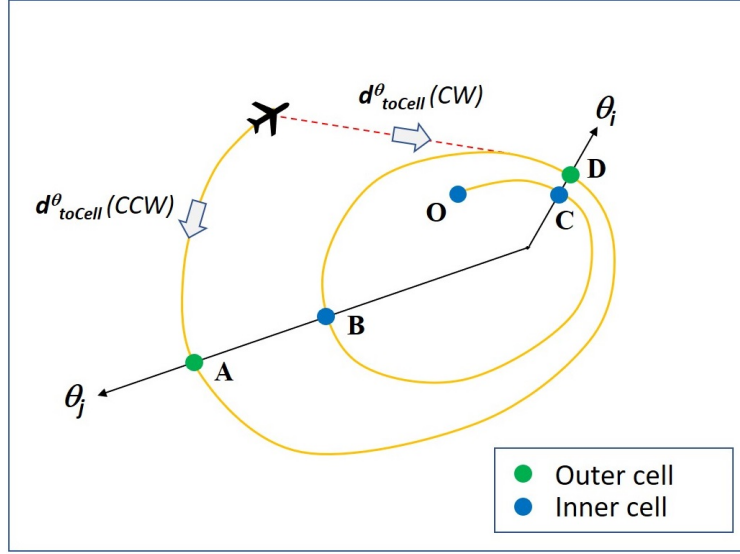


Figure 3.2: Angle based trajectory cell tracking

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 work. 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 gran-

ularity 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 adequate precision and efficiency. Fig. 3.2 illustrates four trajectory cells A, B, C, and D, and their corresponding angles  $\theta_j$  and  $\theta_i$ . Among them, cell A and B have the same angle  $\theta_j$ , and cell C and D have the same angle  $\theta_i$ . The angle lines are based on the cell  $I$ , which is the original ignition point of the fire. The 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  $\theta$  : 1) the rate of spread ( $ROS^\theta$ ); 2) the elapsed time since the cell was last visited ( $t_{sinceVisit}^\theta$ ), and 3) the time it will take for the UAS to reach the cell along the fire perimeter from its current location ( $t_{toCell}^\theta$ ).

First, the  $ROS^\theta$  is the measure of how fast the fire is spreading along the direction of angle  $\theta$ . To calculate  $ROS^\theta$ , we use the most recent trajectory cell along the angle  $\theta$  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  $\theta$ , the UAS keeps track of the distance of the outer cell and inner cell from the ignition point ( $d_{outer}^\theta$  and  $d_{inner}^\theta$ ) and the visit time of those cells ( $t_{outer}^\theta$  and  $t_{inner}^\theta$ ). When the UAS reaches a new cell at the same angle  $\theta$ , the previous outer cell at that angle becomes the

inner cell. The outer cells and inner cells at two sample angles  $\theta_i$  and  $\theta_j$  are displayed in Fig. 3.2. With the information of the outer cell and inner cell, the  $ROS^\theta$  is calculated using 3.1. Secondly, for each angle  $\theta$ , the time since last visit  $t_{sinceVisit}^\theta$  is measured by the difference between the current time ( $t_{current}$ ) and the visit time of the outer cell at the direction and calculated as per 3.2.

$$ROS^\theta = \frac{d_{outer}^\theta - d_{inner}^\theta}{t_{outer}^\theta - t_{inner}^\theta} \quad (3.1)$$

$$t_{sinceVisit}^\theta = t_{current} - t_{outer}^\theta \quad (3.2)$$

$$t_{toCell}^\theta = \frac{d_{toCell}^\theta}{UAS\_SPEED} \quad (3.3)$$

$$imp^\theta = ROS^\theta \times t_{sinceVisit}^\theta + ROS^\theta \times t_{toCell}^\theta \quad (3.4)$$

The multiplicity of the  $ROS^\theta$  and  $t_{sinceVisit}^\theta$  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  $ROS^\theta$  may dynamically change due to the varying condition of the fire area. In this work, we ignore that dynamics and use the most recent  $ROS^\theta$  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  $t_{toCell}^\theta$ , which is the time needed for the UAS to travel from its current location to the cell at the angle  $\theta$ . Since the UAS always flies CW or CCW along the fire perimeter, the  $t_{toCell}^\theta$  would be the “perimeter distance” to the cell, denoted as  $d_{toCell}^\theta$ , divided by the flying speed of the UAS, as shown in 3.3.

To calculate  $t_{toCell}^\theta$ , we need to know  $d_{toCell}^\theta$ . Several things are worthy of mention for computing the  $d_{toCell}^\theta$ . First, depending on if the UAS flies CW or CCW, the distance to a specific cell would be different. In the example of Fig. 3.2, if the UAS keeps moving CW, the  $d_{toCell}^\theta$  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  $d_{toCell}^\theta$  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  $d_{toCell}^\theta$  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. 3.2 (see section 4.2 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  $d_{toCell}^\theta$ . 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  $d_{toCell}^\theta$  that we calculate based on the “current perimeter” and the actual travel distance for reaching the direction of the target cell (in fact,  $d_{toCell}^\theta$  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  $d_{toCell}^\theta$ .

Based on the three measurements described above, we calculate the importance of a cell using 3.4. Here, the importance of a perimeter cell is determined by two factors:  $ROS^\theta \times t_{sinceVisit}^\theta$  and  $ROS^\theta \times t_{toCell}^\theta$ . The first factor represents the new spread distance at the angle  $\theta$  since the cell was last visited, and the second factor represents the new spread distance at the angle  $\theta$  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 an UAS. How

do these two factors impact the path planning of an UAS has been discussed in detail in Chapter 5.

### 3.2 Importance-based Path Planning

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. A planning window is a specific section of the fire perimeter from the current position of the UAS. We consider two identical-sized planning windows from the current position of the UAS - one window represents the perimeter region at the back of the UAS denoted as the back window and the other one represents the perimeter region in front of the UAS denoted as the front window. When the UAS is moving clockwise, the planning window in the clockwise direction is considered the front window, and the planning window in the counterclockwise direction is considered the back window. On the other hand, when the UAS is moving counterclockwise, the planning window in the counterclockwise direction is considered as the front window and the planning

window in the clockwise direction is the back window. The UAS computes and compares cumulative importance in the front and back window and makes its decision regarding its optimal flying direction.

The length of the planning windows is an important factor for UAS path planning. If it is too large e.g., greater than 50% of the perimeter size, the front and back planning windows will overlap, consequently, the overlapping cells will not make any contribution to decision making as they will be counted in both windows. On the other hand, if the length is very small (e.g., only 20% of the perimeter size), only a small number of perimeter cells that are close to the UAS' current location will be considered in the path planning. The cells that are far away from the UAS (e.g., those cells that are on the opposite side of the fire perimeter) will not contribute information to the path planning. This brings the risk that some cells that have high importance (e.g., not being visited for a long time) will never get a chance to participate in the path planning. In our work, the length of the planning window is set to half of the current perimeter length. The rationale is, 50% of the perimeter cells will be in the forward window and 50% remaining cells will be in the backward window, thus having no overlapping cells. Furthermore, every cell of the fire perimeter will be considered in the decision-making process, which ensures that the UAS will not keep oscillating at any local maxima.

Among 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 ( $IMPORTANCE_{BACK}$ ) and compare it to the total importance in the front window

( $IMPORTANCE_{FRONT}$ ). If  $IMPORTANCE_{BACK}$  is larger, it means that the 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 toward the best flying direction along the perimeter.

Assuming there are a total  $2N$  number of cells in the current perimeter,  $IMPORTANCE_{BACK}$  and  $IMPORTANCE_{FRONT}$  are calculated according to 3.5 and 3.6 respectively, where  $\theta_i$  represents the angle of the  $i^{th}$  perimeter cell. Here, the index of a perimeter cell( $i$ ; where  $1 \leq i \leq 2N$ ) 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, then it increases towards the opposite of UAS's flying direction along the perimeter and end at the cell that is just in front of the UAS. When the UAS move to another cell, the indices of the perimeter cells get updated dynamically.

$$IMPORTANCE_{BACK} = \sum_{i=1}^N imp^{\theta_i} \quad (3.5)$$

$$IMPORTANCE_{FRONT} = \sum_{i=N+1}^{2N} imp^{\theta_i} \quad (3.6)$$

The overall concept of decomposing the fire perimeter into the back and front planning window is illustrated in Fig. 3.3. Based on the fire spreading scenario presented in Fig. 3.1,

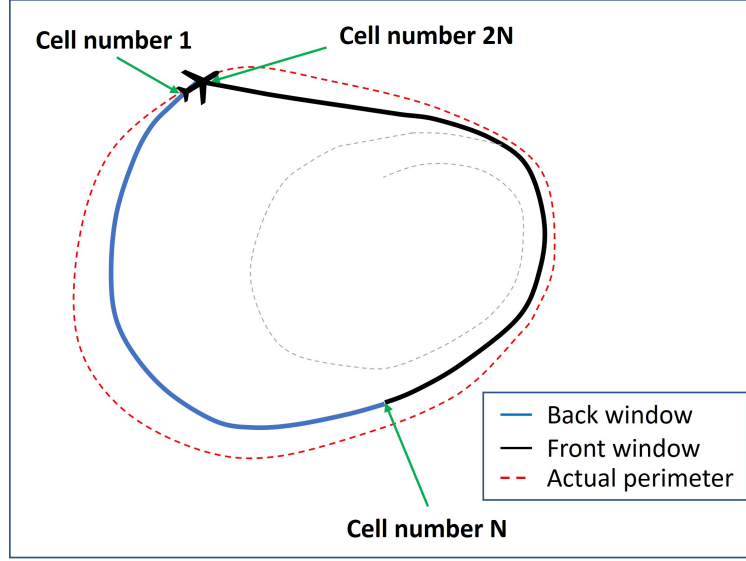


Figure 3.3: Importance Based Path Planning

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. 3.3), which is unknown to the UAS. The rest of the UAS trajectory marked by the grey dashed line is not a part of the reconstructed fire perimeter. Therefore, in this sample scenario,  $IMPORTANCE_{BACK}$  is the sum of the importance of the cells falling under the blue line (cell number 1 to N), and  $IMPORTANCE_{FRONT}$  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

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, the 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 ( $\alpha$ ), which provides the proposed algorithm with the capability of modulating the tendency of flying direction change.

In our fire spread model, the heterogeneity of the 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 3.7 shows how the adjustment is carried out for a cell, where  $ROS^\theta$  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  $ROS_{adjusted}^\theta$  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 showed in 3.7. 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  $\alpha$  and  $ROS_{avg}$ , the adjusted ROS values of the perimeter cells are calculated as per 3.7.

$$ROS_{adjusted}^{\theta} = \begin{cases} ROS^{\theta}, & \text{if } ROS^{\theta} < ROS_{avg} \text{ and } \alpha > 1 \\ ROS_{min} + \alpha(ROS^{\theta} - ROS_{min}), & \text{otherwise} \end{cases} \quad (3.7)$$

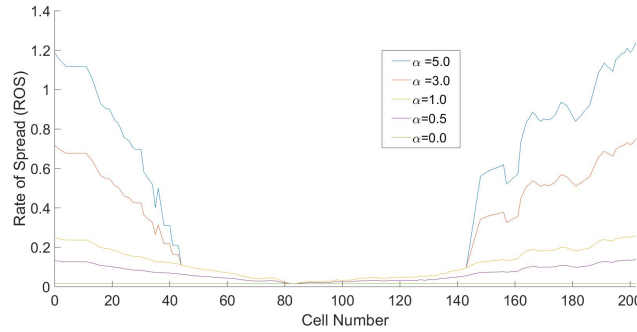


Figure 3.4: Heterogeneity adjustment of the fire spread

When  $\alpha = 0$ , the adjusted ROS values of every perimeter cell are constant ( $ROS_{adjusted}^{\theta} = ROS_{min}$ ), thus there is no heterogeneity in the adjusted ROS values. As  $\alpha$  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  $\alpha$  is larger than 1. Fig. 3.4 shows the heterogeneity adjustment for different  $\alpha$  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  $\alpha$  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.

## CHAPTER 4

### THE IMPORTANCE-BASED ON-BOARD PATH PLANNING ALGORITHM FOR SINGLE UAS

Based on the importance-based path planning concepts discussed in the previous section, this section describes the design and implementation details of the single-UAS 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 listing 1.

---

#### LISTING 1: 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  $\text{IMPORTANCE}_{\text{BACK}} > \text{IMPORTANCE}_{\text{FRONT}}$   
     Change flying direction  
     ELSE  
     Keep flying in the current flying direction
    - iv. Select the next cell along flying direction and fly over there
- 

#### 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 started. The UAS needs the initial fire spread information such as which cells are burning at the time of deployment. As the simplest solution, we can assume that 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. To eliminate this assumption, we have 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 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  $\theta_{init}$ . The UAS keeps flying along the fire perimeter until it returns back to  $\theta_{init}$  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 Fig. 4.1. 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 when there are multiple cells in between two discrete angles, the on-board map also keeps track of those intermediate cells. Keeping track of the intermediate cells is necessary for the precise calculation of the importance of the back and front windows.

Angle	Outer Cell	Inner Cell	Intermediate Cells
0	cell <sub>0</sub>	cell <sub>0</sub>	cell_list <sub>0</sub>
1	cell <sub>1</sub>	cell <sub>1</sub>	cell_list <sub>1</sub>
...	...	...	...
359	cell <sub>359</sub>	cell <sub>359</sub>	cell_list <sub>359</sub>

Figure 4.1: On-board fire map

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 the initial inner cell. Then, as the UAS moves along the perimeter, the outer cells, inner cells, and intermediate

---

**LISTING 2: PROCEDURE FOR ON-BOARD FIRE MAP  
INITIALIZATION**


---

**OUTPUT:** Initialized on-board fire map

Determine the angle  $\theta_{init}$  of the UAS deployment cell from the ignition point

**REPEAT UNTIL** the UAS comes back to  $\theta_{init}$  :

- i. Determine the angle  $\theta$  of the cell from the ignition point
- ii. Add the cell as the outer cell at the angle  $\theta$  in on-board fire map
- iii. Add ignition point as the inner cell at the angle  $\theta$
- iv. Move to the next cell along UAS's current flying direction

**END**

---

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 listing 2.

## 4.2 Fire Shape Reconstruction

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. 4.2a), 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 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.

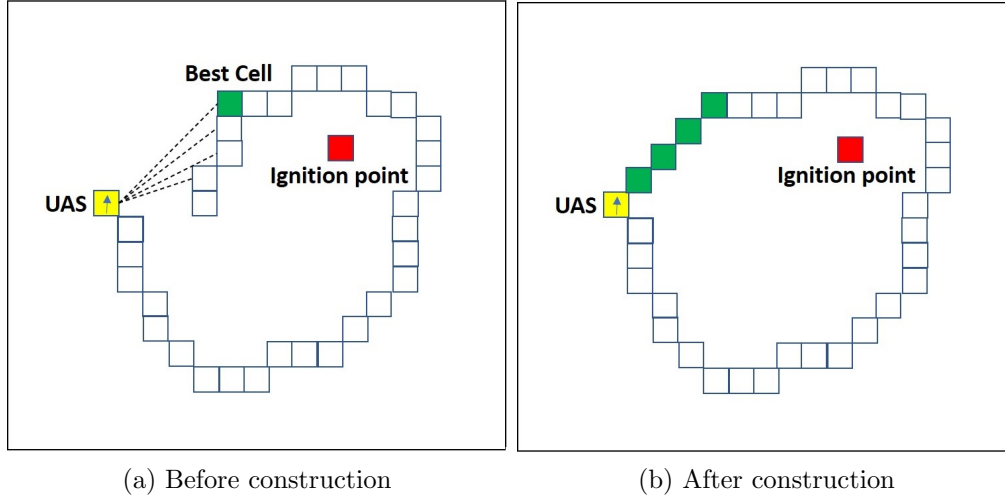


Figure 4.2: On-board fire perimeter construction

Fig. 4.2 illustrates how the on-board fire perimeter construction works. In Fig. 4.2a, 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. 4.2b. This way a smooth fire shape is generated using the on-board information collected by

the UAS.

### 4.3 Determining Optimal Flying Direction

---

**LISTING 3: 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  $ROS_{adjusted}^{\theta}$  of the *current\_cell*
- ii. Calculate  $t_{sinceVisit}^{\theta}$  of the *current\_cell*
- iii. Calculate  $t_{toCell}^{\theta}$  of the *current\_cell*
- iv. Determine  $imp^{\theta}$  of the *current\_cell*

**END FOR**

Calculate  $IMPORTANCE_{BACK}$  and  $IMPORTANCE_{FRONT}$

Change the flying direction if  $IMPORTANCE_{BACK} > IMPORTANCE_{FRONT}$

---

After the fire perimeter construction, the final step is to calculate  $IMPORTANCE_{BACK}$  and  $IMPORTANCE_{FRONT}$  and decide the optimal flying direction for the UAS. The decision-making procedure is shown in listing 3. 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 3.4. 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  $IMPORTANCE_{BACK}$  is larger than  $IMPORTANCE_{FRONT}$ .

## CHAPTER 5

### ANALYSIS OF THE SINGLE UAS PATH PLANNING ALGORITHM

In this section, we analyze the proposed path planning algorithm from several important perspectives. These analyses are important to have an in-depth understanding of how the proposed algorithm controls the path of an UAS. We 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

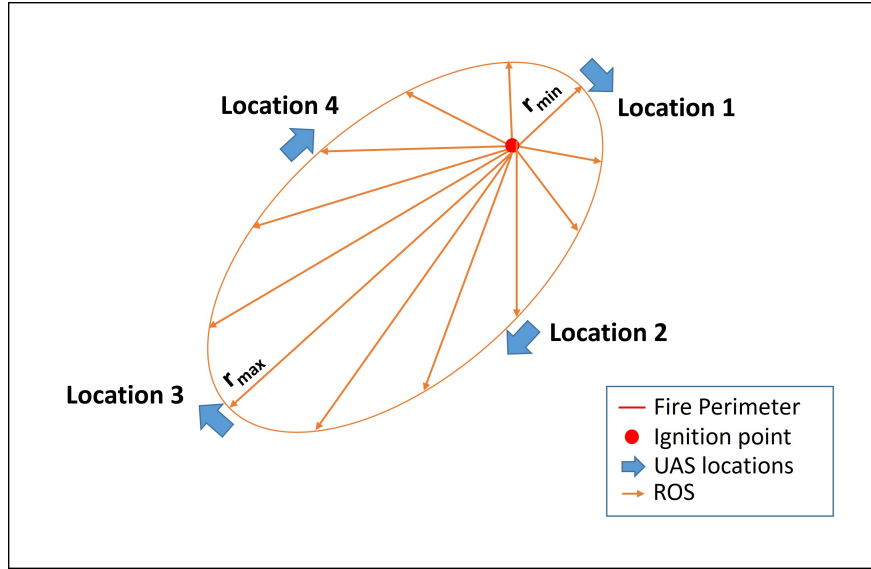


Figure 5.1: Incremental fire spread scenario

For monitoring a dynamically growing wildfire, the correctness of the path planning algorithm is very important as the trajectory of the UAS is determined by it. In this section,

we consider a representative fire spread scenario and examine the flying decisions of the UAS at different locations of the perimeter. We assume the speed of the UAS is  $s$ , the size of each perimeter cell is  $c$ , and the ROS of perimeter cell  $i$  is  $r_i$ .

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. 5.1, and the rate of spread (ROS) along the fire perimeter gradually increases from  $r_{min}$  to  $r_{max}$  as illustrated in Fig. 5.2. For a such pattern of the ROS values, we will denote this scenario as “incremental fire spread” scenario. As there are total  $N$  number of cells between  $r_{min}$  and  $r_{max}$ , the increment of ROS between each cell ( $\delta$ ) can be expressed 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  $\delta$  can be alternatively expresses as  $\delta = (ar_{min} - r_{min})/N$ . Thus, given the value of  $r_{min}$  and  $\delta$ , the rate of spread at any cell which is  $i$  cells away from  $r_{min}$  direction can be expressed as  $(r_{min} + i\delta)$ .

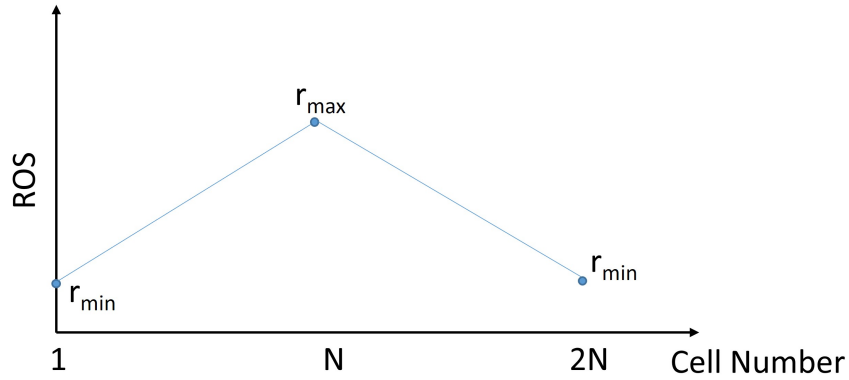


Figure 5.2: The incremental fire spread model

Based on the above description of the incremental fire spread model, we consider four locations on the fire perimeter as shown in Fig. 5.1 and analytically find out whether the

UAS should change its flying direction at those locations. Location 1 is the perimeter cell which is in the direction of  $r_{min}$ , location 3 is the cell that is in the direction of  $r_{max}$ . Location 2 and location 4 are the middle points between location 1 and location 3, located on the lower and upper flank of the fire perimeter respectively. We assume that the UAS is flying in the clockwise direction at those locations.

To analyze the UAS's decision regarding its flying direction change at those locations, we consider a metric that represents the difference between the total importance of the front and back planning window ( $IMP_{DIFF}$ ) which is calculated as:

$$IMP_{DIFF} = IMPORTANCE_{FRONT} - IMPORTANCE_{BACK}$$

If  $IMP_{DIFF}$  is positive at a location, it means  $IMPORTANCE_{FRONT}$  is larger than  $IMPORTANCE_{BACK}$  and the UAS should move forward at that location. Otherwise, the UAS should change its flying direction by turning back. Below we show the  $IMP_{DIFF}$  at those four locations analytically. To save some space, the specific procedures for deriving the final results are omitted.

**Location 1:**

$$IMPORTANCE_{BACK}$$

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

$$= (N^2 r_{min} + N r_{min} + \frac{\delta N^3}{3} + \frac{\delta N^2}{2} + \frac{\delta N}{6}) * \frac{c}{s}$$

$$IMPORTANCE_{FRONT}$$

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

$$= (2N^2 r_{min} + \delta N^3 + \delta N^2) * \frac{c}{s}$$

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

**Location 2:**

$$IMPORTANCE_{BACK}$$

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

$$= \left[ N^2 r_{min} + a r_{min} \frac{N(N+2)}{4} - r_{min} \frac{N(N+2)}{4} \right] * \frac{c}{s}$$

$$IMPORTANCE_{FRONT} = 2 \sum_{i=1}^{\frac{N}{2}} \left\{ r_{min} + \left( \frac{N}{2} + i \right) \delta \right\} 2N * \frac{c}{s}$$

$$= \left[ N^2 r_{min} + a r_{min} \frac{N(N+2)}{4} - r_{min} \frac{N(N+2)}{4} + \frac{aN^2 r_{min}}{2} - \frac{aN^2 r_{min}}{2} \right] * \frac{c}{s}$$

$$IMP_{DIFF} = \frac{N^2 r_{min}}{2} (a - 1) * \frac{c}{s}$$

**Location 3:**

$$\begin{aligned}
IMPORTANCE_{BACK} &= \sum_{i=1}^N (r_{min} + i\delta)(N - i)2 * \frac{c}{s} \\
&= \left( \sum_{i=1}^N 2Nr_{min} + \sum_{i=1}^N 2i\delta N \right. \\
&\quad \left. - 2r_{min} \sum_{i=1}^N i - 2\delta \sum_{i=1}^N i^2 \right) * \frac{c}{s} \\
IMPORTANCE_{FRONT} &= \sum_{i=1}^N (r_{min} + i\delta)2N * \frac{c}{s} \\
&= \left( \sum_{i=1}^N 2Nr_{min} + \sum_{i=1}^N 2i\delta N \right) * \frac{c}{s} \\
IMP_{DIFF} &= (2r_{min} \sum_{i=1}^N i + 2\delta \sum_{i=1}^N i^2) * \frac{c}{s}
\end{aligned}$$

**Location 4:**

$$\begin{aligned}
&IMPORTANCE_{BACK} \\
&= \left( \sum_{i=1}^{\frac{N}{2}} \left\{ r_{min} + \left( \frac{N}{2} + i \right) \delta \right\} 2i * \frac{c}{s} \right. \\
&\quad \left. + \sum_{i=1}^{\frac{N}{2}} \left\{ r_{min} + \left( \frac{N}{2} + i \right) \delta \right\} (N - i)2 * \frac{c}{s} \right) \\
&= \left[ \frac{N^2 r_{min}}{4} (3a + 1) + \frac{Nr_{min}}{2} (a - 1) \right] * \frac{c}{s}
\end{aligned}$$

$$\begin{aligned}
&IMPORTANCE_{FRONT} \\
&= 2 \sum_{i=1}^N (r_{min} + i\delta)2N * \frac{c}{s} \\
&= \left[ \frac{N^2 r_{min}}{4} (2a + 6) + \frac{Nr_{min}}{2} (a - 1) \right] * \frac{c}{s} \\
IMP_{DIFF} &= \frac{N^2 r_{min}}{4} (5 - a) * \frac{c}{s}
\end{aligned}$$

Therefore, for location 1, if the window size ( $N$ ) is larger than 1,  $IMP_{DIFF}$  is positive and the UAS should move forward. Practically,  $N$  is always to be larger than 1, hence the UAS should always move forward at location 1. 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. At location 3, the UAS should always move forward as well. Finally, at 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 counterclockwise direction.

The above analyses of the UAS's flying direction provide us with some important insights about the algorithm. First, based on the analysis for location 1, the UAS will always move forward when it is at the minimum spread direction of a fire (e.g., location 1). Similarly, based on the analysis for location 2, the UAS will also keep moving forward while it flies towards the maximum spread direction from the minimum spread direction (e.g., from location 1 towards location 3 in the clockwise direction). This is desired because the cells at the front of the UAS have high spreading speeds and have not been visited for a relatively longer time (compared to the cells at the back of the UAS that the UAS just visited). For Location 3, the analysis shows that the UAS will keep moving forward. This makes sense because a portion of the fire perimeter in front of the UAS still belongs to the "active region" of fire, and thus the UAS should not turn back at Location 3, which would be too early to turn back. For the remaining section of the fire perimeter, the UAS's flying decision will depend on the location

of the UAS and the value of  $a$ . As the UAS starts to fly from the maximum spread direction towards the minimum spread direction in the clockwise direction, the  $IMP_{DIFF}$  starts to decrease, thus 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 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. On the other hand, if the fire spreads very fast in a specific region, the UAS is likely to turn back after finishing visiting that region. The turning back makes the UAS to revisit the most active region of the fire, which is what we want.

Note that the above analysis is based on the scenario that the UAS did not turn back in the past round of the full perimeter. When the UAS turns back, the  $t_{sinceVisit}$  values for the front and back window cells would be changed, and the  $IMP_{DIFF}$  equations for the four locations would be different. This means that even though  $r_{max}$  could be much larger than  $r_{min}$ , after the UAS turns back and forth around the  $r_{max}$  region several times, the UAS will eventually keep moving forward to cover the  $r_{min}$  region because the cells in that region would have much larger  $t_{sinceVisit}$  values.

## 5.2 Qualitative Analysis

In this section, we qualitatively examine how two important properties of the fire perimeter,  $t_{sinceVisit}$  and  $t_{toCell}$ , impact the UAS's flying direction. When the UAS moves to a new cell on the fire perimeter, the ROS values of the perimeter cells do not get changed due to this location change. However,  $t_{sinceVisit}$  and  $t_{toCell}$  values are dynamically updated whenever the UAS moves to a new cell. Thus,  $t_{sinceVisit}$  and  $t_{toCell}$  have a dynamic impact on the path of an UAS. Below we analyze how these two properties dynamically influence the flying behavior of an UAS. These analyses are based on the incremental fire spread scenario described in the preceding section.

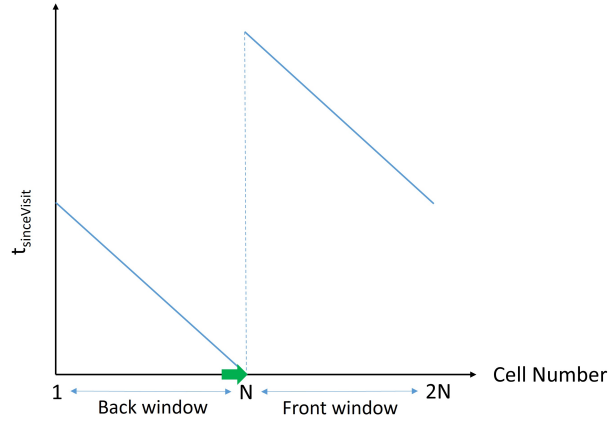


Figure 5.3: Pattern of  $t_{sinceVisit}$  values along fire perimeter

### 5.2.1 Impact of $t_{sinceVisit}$ on UAS's Flying Direction

In Equation 3.4, we can see how the ROS and  $t_{sinceVisit}$  values combinedly participate in 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.

To understand the impact of  $t_{sinceVisit}$ , we will qualitatively examine how this portion of the cell importance ( $ROS^\theta \times t_{sinceVisit}^\theta$ ) impacts the UAS's flying behavior. We will demonstrate the impact referring to the four locations on the fire perimeter as shown in Fig. 5.1. For demonstration purposes, we will refer to a segment of the fire perimeter starting from location  $A$  and ending at location  $B$  as  $SEG_{AB}$ .

- 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,  $t_{sinceVisit}$  values in the front window are always higher than the back window, as shown in Fig. 5.3. Thus, the UAS will move forward at this location because the total importance in the front window will be higher due to the higher  $t_{sinceVisit}$  values.
- As the UAS flies towards location 2, the total importance of the front window keeps growing. This is because the front window acquires cells with higher ROS values from  $SEG_{34}$  and releases cells with lower ROS values from  $SEG_{12}$ . Thus, at location 2, the UAS will keep moving forward based on the  $t_{sinceVisit}$  and ROS values of the front and back window.
- Next, when the UAS flies towards location 3, even though the front window releases cells with higher importance from  $SEG_{23}$  and acquires cells with lower importance from  $SEG_{41}$ , still the total ROS values in the front window will remain higher until the UAS arrives at location 3. At location 3, the ROS values for the front and back

windows will become equal. However, due to the higher  $t_{sinceVisit}$  values at the front window, the UAS will move forward at location 3.

- As the UAS moves towards location 4, the front window further releases cells with higher importance from  $SEG_{34}$ . When the UAS reaches location 4, the front window consists of  $SEG_{41}$  and  $SEG_{12}$ , which have lower ROS values compared to the segments of the back window. Therefore, at location 4, ROS values of the back window will influence the UAS to turn back, while the  $t_{sinceVisit}$  values of the front window will influence the UAS to move forward. If the ROS values at the back window are large enough to dominate the higher  $t_{sinceVisit}$  values at the front window, the UAS will turn back. Otherwise, the UAS will keep flying towards location 1.
- As the UAS flies towards location 1, the UAS might keep moving forward or turn backward depending on which has greater influence between ROS and  $t_{sinceVisit}$ .

From this analysis, we conclude that the  $t_{sinceVisit}$  influences the UAS to move forward along the fire perimeter. Therefore, in our formula for determining the cell importance 3.4, the first term of the equation is responsible for pushing the UAS forward along the fire perimeter.

### 5.2.2 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. 5.4. The values keep increasing identically around the location of the UAS. During cell importance

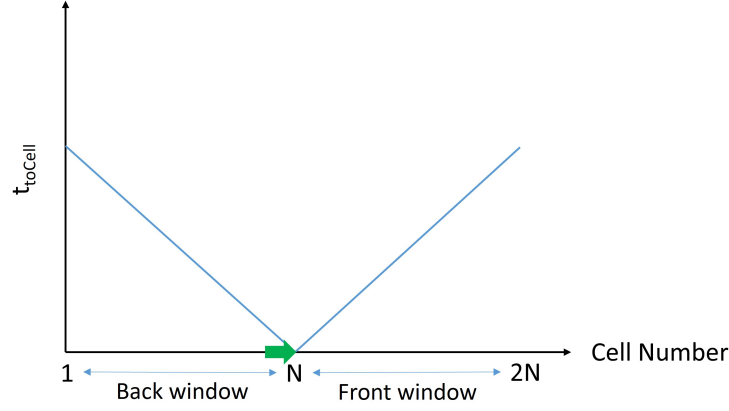


Figure 5.4: Pattern of  $t_{toCell}$  values along fire perimeter

calculation,  $t_{toCell}$  is multiplied by the ROS value, and another portion of the cell importance is obtained. In this section, we will discuss how this portion of the cell importance impacts the UAS's flying behavior.

- 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 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, the importance in the back window will become higher

again and it will turn back. Thus, the UAS will keep oscillating around the  $r_{max}$  direction.

- The flying behavior of the UAS is similar when its location is on  $SEG_{34}$  or  $SEG_{41}$  – the  $t_{toCell}$  values of the perimeter will bring the UAS to the  $r_{max}$  direction and keep the UAS oscillating over that region.

To conclude, the  $t_{toCell}$  parameter influences the UAS to stay on the most active region. Thus, in our formula for determining the cell importance 3.4, the second term of the equation is responsible for keeping the UAS on the most active region.

Based on the above analyses, it is evident that  $t_{sinceVisit}$  values of the fire perimeter influence the UAS to move forward to cover the fire perimeter cyclically, while  $t_{toCell}$  values influence the UAS to oscillate over the most active region. Thus, the UAS achieves the capability of both revisiting the most active fire region and covering the entire fire perimeter from time to time.

## CHAPTER 6

### SINGLE UAS PATH PLANNING RESULTS

The experiments are based on simulated wildfires; we have used DEVS-FIRE (36; 37) framework to simulate different fire spreading scenarios. In this chapter, 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) a demonstration of path planning for non-uniform fuel models iv) a 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 (36; 37), which is a discrete event simulation model developed based on the Discrete Event System Specification (DEVS) formalism (38). 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 (39) to compute the fire spread rate and direction within the cell. Fire spreading is modeled as a propagation process as burning cells ignite their unburned neighboring cells.

The DEVS-FIRE model has been validated by comparing with other wildfire simulation tools (40) and by simulating historical wildfires (41). 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.

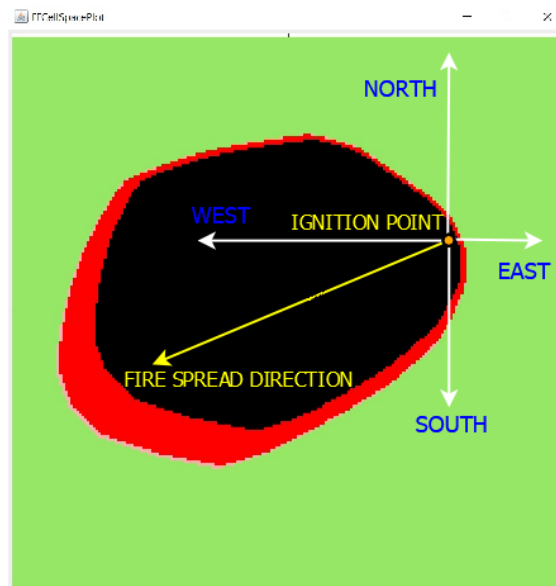


Figure 6.1: Wildfire spread simulation using DEVS-FIRE

A sample fire spread scenario using DEVS-FIRE has been shown in 6.1. 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. 10. 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 200x200 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 the Algorithm

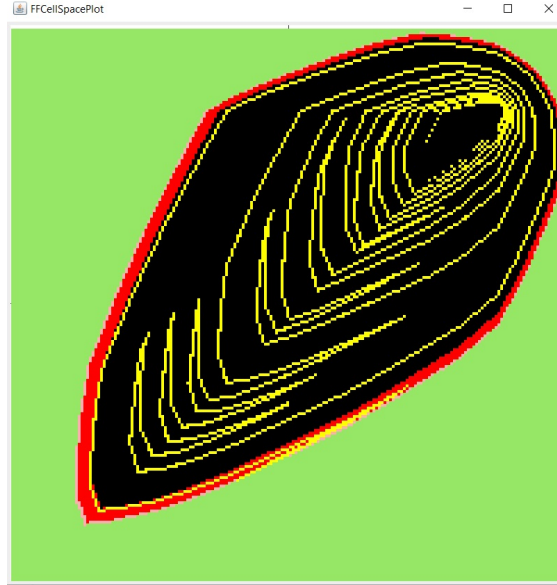


Figure 6.2: UAS trajectory based on the proposed algorithm

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. 6.2, the fire starts from the northeast side of the cell space and spreads towards the southwest direction over time. During the first 3 hours of the simulation, the fire was not spreading very fast, as a result, the UAS did not make too many revisits during that 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 in the southwest direction.

Fig. 6.3a 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. 6.3b. 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. 6.3c 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. 6.3d. 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 ( $\alpha$ ) is 1.0.

Fig. 6.4 compares the UAS trajectories for six different  $\alpha$  values for the same fire spread scenario presented in Fig. 6.2. When the value of  $\alpha$  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  $\alpha$  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  $\alpha$  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  $\alpha$ , the UAS changed its flying direction 0, 8, 11, 15, 20, and 43 times respectively as showed in Fig. 13. It should be noted that the UAS made no direction change when  $\alpha$  is very small (e.g., 0.0). In contrast, the UAS made

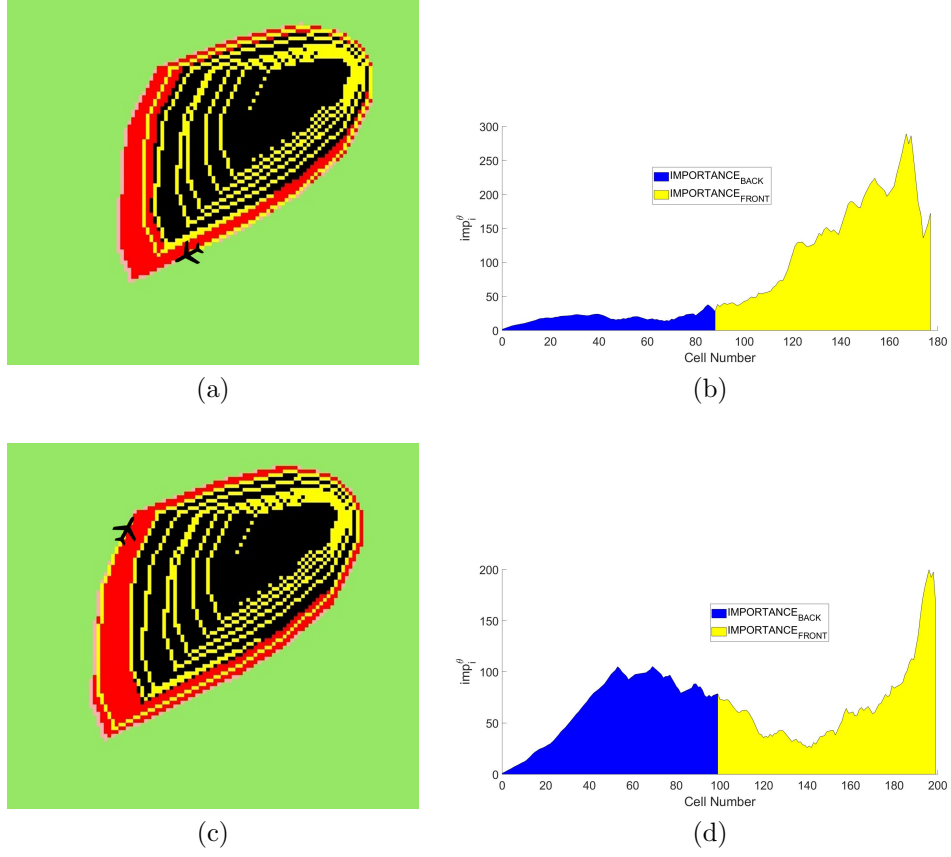


Figure 6.3: Illustration of back and front window importance

a large number of direction changes when  $\alpha$  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  $\alpha$ . 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  $\alpha$  ii) the most active fire front by setting a higher value of  $\alpha$ . iii) both the entire perimeter and the most active fire front by setting an intermediate value of  $\alpha$ .

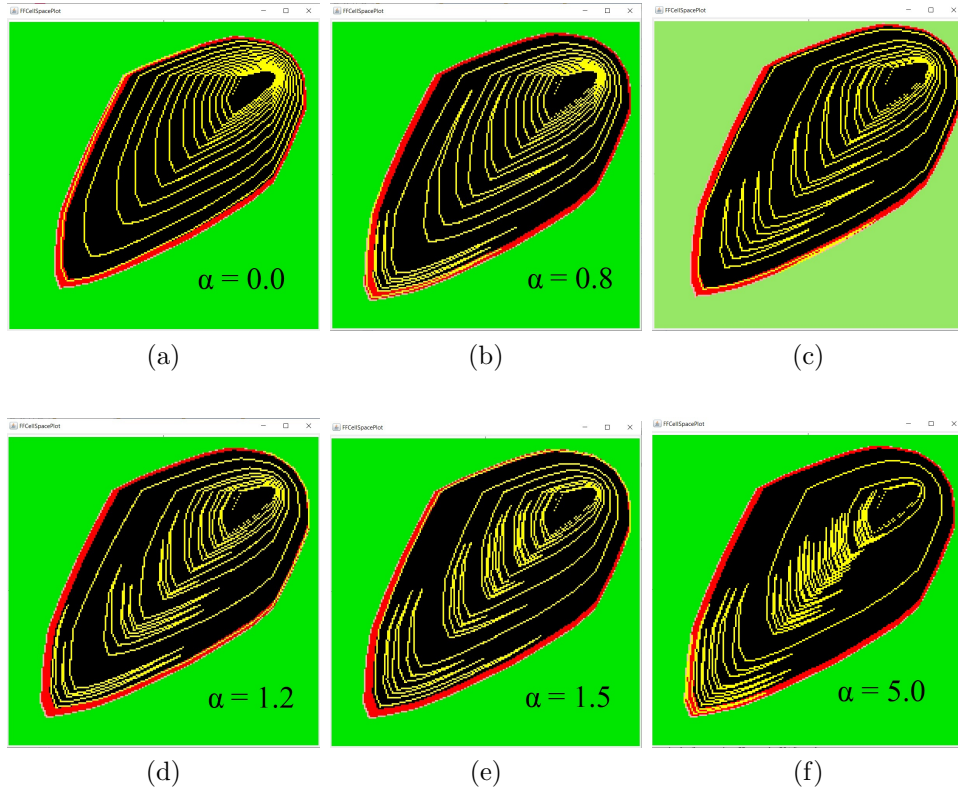


Figure 6.4: UAS trajectory for different  $\alpha$  values

### 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 (Fuel Model 5) (42) which has moderate burning characteristics, and another half of the cell space consists of short grass (Fuel Model 1) which has faster-burning characteristics. Fig. 6.5a shows the UAS trajectory while monitoring a fire in such a fuel 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 characteristics followed by fuel with slow-burning characteristics. 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. Fig. 6.5b 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.

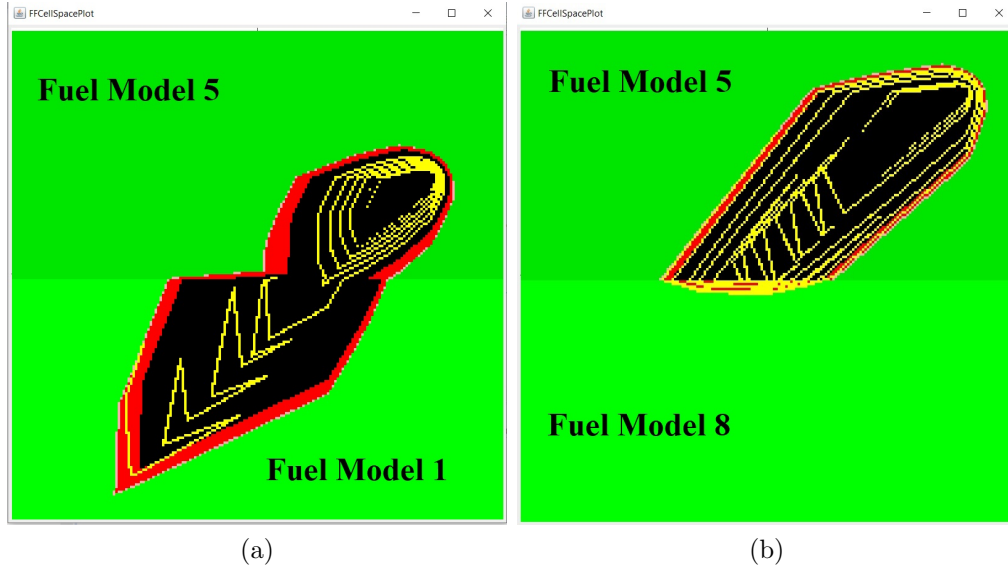


Figure 6.5: UAS trajectory based on non-uniform fuel loadings

#### 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. 6.6a. 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 is spreading slowly around the ignition point as shown in Fig. 6.6b. There is no significantly fast-spreading region in this scenario, therefore  $IMPORTANCE_{BACK}$  has never grown larger than  $IMPORTANCE_{FRONT}$ . 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 the 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. 6.6c, 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.

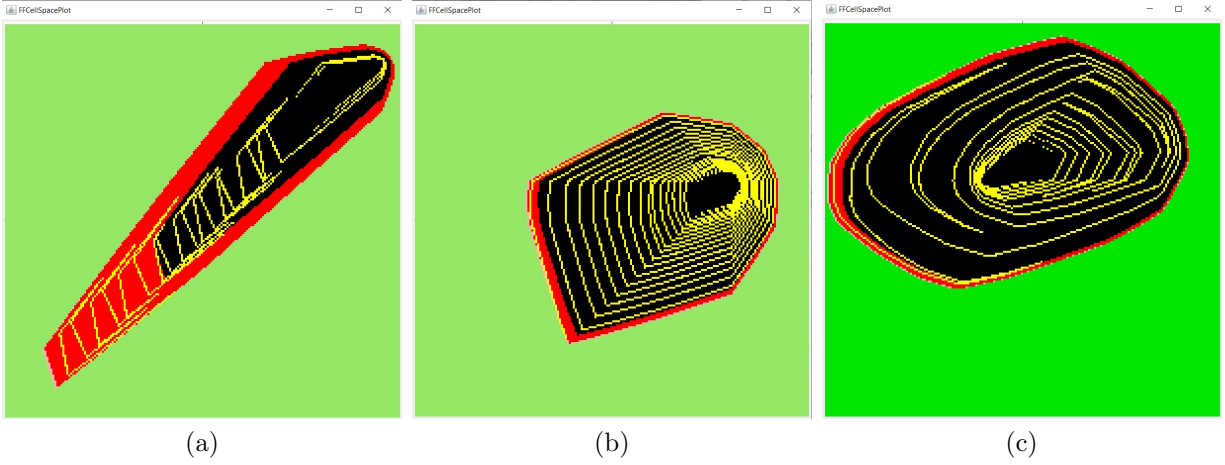


Figure 6.6: UAS trajectory in different wind conditions

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. However, if a section of the fire is spreading much faster, this approach is unable to provide higher monitoring attention to that region. Fig. 6.7a and 6.7b show the UAS trajectories based on the circling-based approach corresponding to the fire spread scenarios used in Fig. 6.2 and Fig. 6.6a respectively. For both scenarios, we can notice larger monitoring gaps on the faster-spreading southwest side of the fire compared to the proposed algorithm.

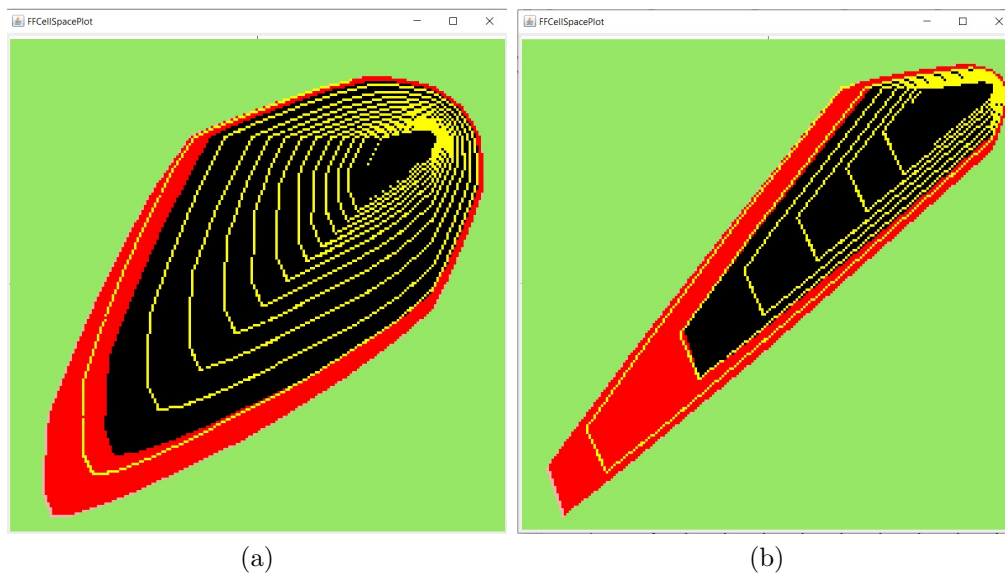
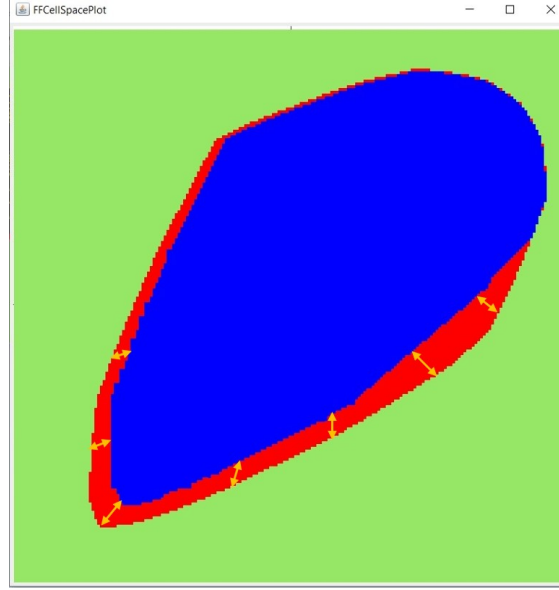


Figure 6.7: UAS trajectory based on the circling-based approach

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 the 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 work. Fig. 6.8 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 (43) 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. 6.8.

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. Fig. 6.9a compares the maximum distance error of these two approaches for the



(a)

Figure 6.8: Distance error between the original fire shape and constructed fire shape

fire spread scenario shown in Fig. 6.7a. From the comparison, we can see that the maximum distance error is similar for the first 3 hours 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.

Fig. 6.9b compares the maximum distance error for the fire spread scenario shown in Fig. 6.7b. 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 because,

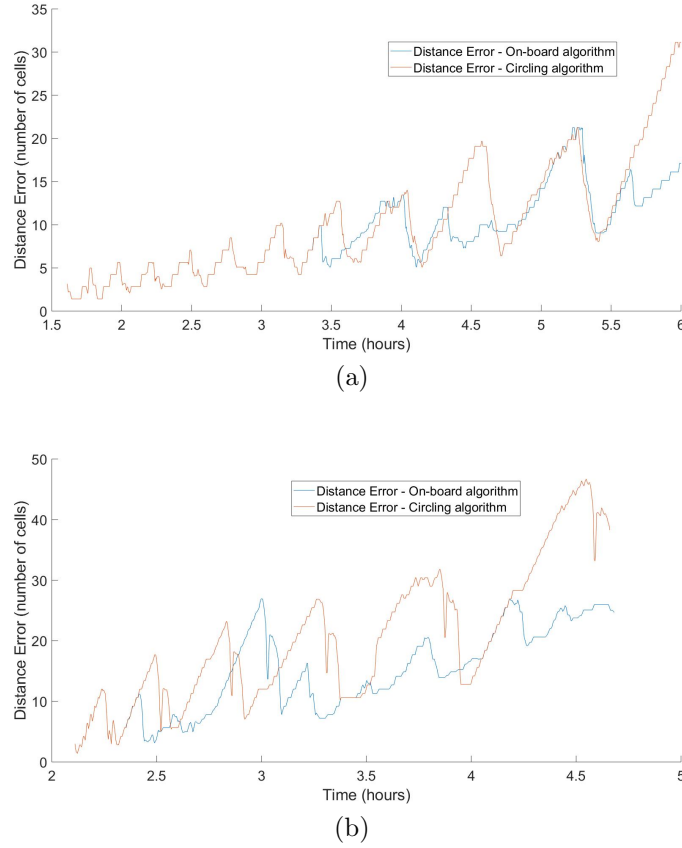


Figure 6.9: Comparison of the proposed algorithm and circling-based approach

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.

## CHAPTER 7

### DECENTRALIZED AND IMPORTANCE-BASED MULTI-UAS PATH PLANNING FOR DYNAMIC WILDFIRE MONITORING

When monitoring small and spatially wildfires, a single UAS might be adequate for the monitoring task. However, for monitoring the wildfires that occur across a large area, it is more beneficial to use multiple UASs instead of a single UAS so that the collected wildfire data is more accurate. More accurate data will help the firefighters to have a better fire monitoring and management plan. However, a firefighting team might have a limited number of resources (UASs) and it is important to make the best use of them to collect the most useful information about the wildfire. Multi-UAS path planning faces several additional challenges compared to single-UAS path planning. Among them, autonomous coordination and task allocation is one of the most challenging tasks. In this section, two different strategies for multi-UAS coordination and path planning have been presented. The first one is based on the "front" and "back" window comparison based approach. Here, the ideas from single-UAS path planning have been extended to accommodate the multi-UAS path planning requirements. The second approach is developed based on a novel back-and-forth coverage strategy.

#### **7.1 Decentralized Importance-based Multi-UAS Path Planning Algorithm: Based on the Extension of the single-UAS Algorithm**

In the single UAS path planning algorithm, the UAS divides the entire fire perimeter into two segments - the front window and the back window. Figure 7.1a shows a sample fire perimeter, the location of an UAS and its corresponding front and back window on top of a

sample cell space (of size 16x20). The UAS computes and compares the total importance of the front and back window, and decides its best flying direction (CW or CCW) by comparing these two.

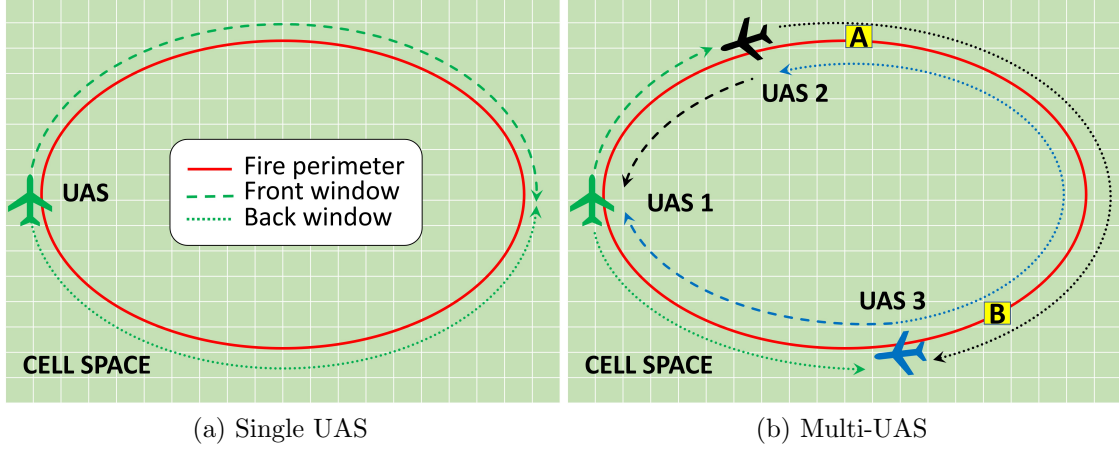


Figure 7.1: Importance-based path planning

For the multi-UAS path planning, we also use the concept of front and back windows to make the decision regarding UASs' best flying direction along the fire perimeter. For the single UAS path planning algorithm, the UAS is responsible for monitoring the entire perimeter. Thus, it is important for the UAS to incorporate the entire perimeter into its front and back windows during its decision-making. In contrast, for multi-UAS path planning, there are multiple UASs available for the monitoring task, therefore each UAS does not need to consider the entire perimeter for path planning. Let's consider the multi-UAS based monitoring scenario presented in 7.1b where three UASs are monitoring the fire perimeter. The location and flying direction of these UASs has been showed in the figure. Now, if the perimeter cell highlighted by *A* is needed to be visited, UAS1 can not reach the cell faster than UAS2. In fact, UAS1 can not reach any cell behind UAS2 faster than it. Similarly,

UAS1 can not also reach any cell behind UAS3, for example, the cell highlighted by  $B$ , faster than the UAS3. Thus, UAS1 can ignore the responsibility of the perimeter cells beyond its neighboring UASs.

In general, for importance-based multi-UAS path planning, the cells between every pair of neighboring UASs are important to be considered for path planning of the corresponding UAS pairs. Based on this idea, the front window of an UAS  $i$  is defined as - the perimeter cells between the UAS  $i$  and its neighboring UAS that is in front of it. On the other hand, the back window includes the perimeter cells between the UAS  $i$  and the neighboring UAS that is in the back of it. Figure 7.1b shows the front and back windows of the UASs by the dashed and dotted lines respectively (each UAS and its corresponding windows is highlighted in the same color). Please note how the length of the front and back window can vary depending on the location of the UAS on the fire perimeter.

### 7.1.1 *Inter-UAS Communication*

The proposed multi-UAS path planning algorithm does not depend on a centralized system or ground station for the data required for path planning. To support the importance-based decentralized path planning, the UASs need to know three pieces of information about each perimeter cell in real-time -  $ROS^\theta$ ,  $t_{sinceVisit}^\theta$ , and  $t_{toCell}^\theta$ . This information is used to calculate the importance of visiting each perimeter cell. These individual importance values are in turn used to calculate the front and back window importance.

To support the on-board calculation of this three path planning information, the UASs will generate and maintain an on-board map that keeps track of the perimeter cells and their

visit times. The information about a part of the fire such as the corresponding perimeter cells and their visit times, are not updated until the UAS flies over that part of the fire again. Thus, the on-board map is generated based on the best knowledge of the UAS about the fire spread. Consequently, there is a margin of error between the actual fire shape and the fire shape generated based on the on-board data.

For multi-UAS path planning, an UAS can take advantage of the other UASs location data to minimize the margin of error and have a more accurate on-board fire map. However, the UASs will need some intercommunication at regular intervals to maintain a more accurate on board map. Based on our design, broadcasting the current location of an UAS is sufficient to achieve this goal. In addition to the accuracy of the on-board map, there is another reason for this intercommunication. When a part of the fire is being visited by an UAS, the importance of that part becomes lower because the  $t_{sinceVisit}$  values of those cells become very small. Without any inter-UAS communication, other UASs will not know that that particular part of the fire is already been visited. As a result, that part of the fire will be treated with higher importance by the other UASs. For these two reasons, in our design, an UAS broadcasts its location data (coordinate of the cell that is occupied by the UAS) whenever they reach a new cell.

Figure 7.2 shows how two UAS exchanges their location data based on the underlying fire spread model. Let's assume two UASs, UAS1 and UAS2, are located on top of the cells  $(X_1, Y_1)$  and  $(X_2, Y_2)$  respectively. Now, when UAS1 receives the location data from UAS2, it determines the relative direction of UAS2. Next, it updates information for that

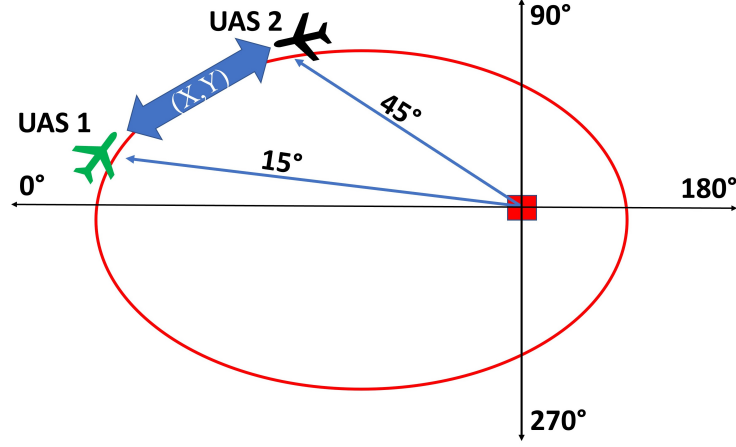


Figure 7.2: Inter-UAS communication strategy

particular direction on its on-board map. Similarly, UAS2 updates the information based on the location of UAS1. Figure 7.3 shows how their on-board maps are updated. Assuming the current location of UAS1 and UAS2 are at 15 deg and 45 deg respectively from the ignition point, the map of UAS1 is updated at index 45 and the map of UAS2 is updated at index 15. Please note the  $t_{sinceVisit}$  value is updated to 0 because the cell occupied by the other UAS is just being visited. Thus, the UASs communicate their location to each other, and they generate a more accurate fire shape collaboratively. There is some communication overhead of this collaboration, however, it is a small price to pay to support the decentralization and accuracy of the multi-UAS path planning algorithm.

After the UASs update their on-board map, next they will determine their front and back windows according to the principles illustrated in Fig. 7.1b. Next, the UAS computes the total importance of the front and back window by adding up the individual cell importance of the corresponding perimeter cells. The individual cell importance is calculated according to 3.4 where  $imp^\theta$  represents the importance of the cell at the angle  $\theta$ . There are three

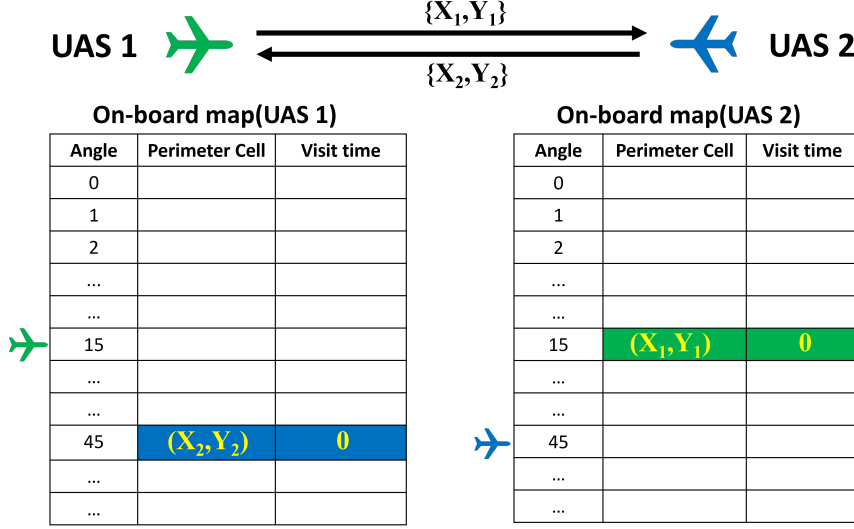


Figure 7.3: Updating the on-board maps

measurements involved in this equation:

#### 7.1.1.1 $ROS^\theta$

It is determined in the same way as the single UAS algorithm. The only difference is that the on-board map is updated collaboratively by the multiple UASs, thus the ROS values are determined based on a more accurate on-board map. Please refer to 3.1 for a detailed description of how  $ROS^\theta$  is calculated based on the on-board map data.

#### 7.1.1.2 $t_{sinceVisit}^\theta$

It is calculated based on the visit time of the perimeter cells available in the on-board map.

#### 7.1.1.3 $t_{toCell}^\theta$

This information represents the minimum time an UAS will require to reach a cell by following the fire perimeter. Accordingly, this information is calculated as - the shortest distance

to reach that perimeter cell divided by the speed of the UAS.

### 7.1.2 Analyzing The Impact of Fleet Size Change

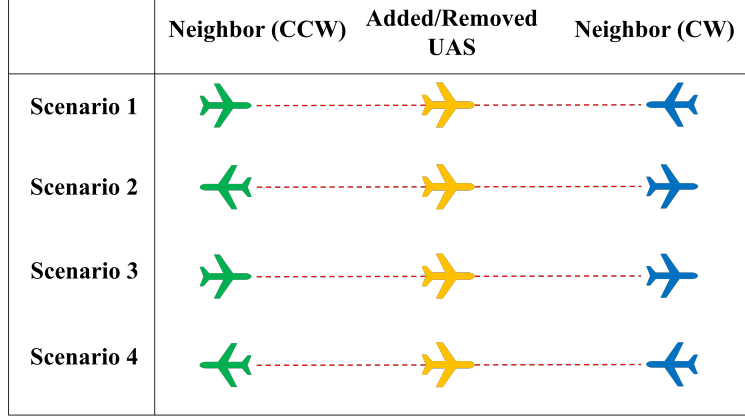


Figure 7.4: Possible scenarios for dynamic fleet size change

UAS \ Scenario		CCW Neighbor			CW Neighbor		
		$I_B$	$I_F$	Move towards targetCell?	$I_B$	$I_F$	Move towards targetCell?
Scenario 1	Add	Same	Reduced	NO	Same	Reduced	NO
	Remove	Same	Increased	YES	Same	Increased	YES
Scenario 2	Add	Reduced	Same	NO	Reduced	Same	NO
	Remove	Increased	Same	YES	Increased	Same	YES
Scenario 3	Add	Same	Reduced	NO	Reduced	Same	NO
	Remove	Same	Increased	YES	Increased	Same	YES
Scenario 4	Add	Reduced	Same	NO	Same	Reduced	NO
	Remove	Increased	Same	YES	Same	Increased	YES

Figure 7.5: Impact of dynamic fleet size change

In this section, we will qualitatively analyze the impact of: i) dynamically adding a new UAS to the fleet, and ii) dynamically removing an existing UAS from the fleet. According to the design of the proposed algorithm, the added/removed UAS will have a direct impact only on its neighboring UASs (CW and CCW neighbor). Regardless of the location on the fire

perimeter where an UAS is added/removed (*targetCell*), there could be four possible scenarios that can arise as showed in Figure 7.4. First, both neighboring UASs are moving towards the *targetCell*. Second, both neighboring UASs are moving outwards from the *targetCell*. Third, the CCW neighboring UAS is moving towards and the CW neighboring UAS is moving outwards the *targetCell*. Fourth, the CCW neighboring UAS is moving outwards and the CW neighboring UAS is moving towards the *targetCell*. Figure 7.5 presents the impact of the addition/removal of an UAS for these four scenarios. Based on the analysis presented here, we conclude that the addition of an UAS influences the neighboring UASs to fly away from the *targetCell*. From the monitoring point of view, this behavior is desirable because the newly added UAS will be taking care of the *targetCell* and its surrounding cells. In contrast, the removal of an UAS influences the neighboring UASs to fly towards the *targetCell*. This is also desirable because the neighboring UASs will take over the responsibility of the removed UAS.

### 7.1.3 *Experiment Results*

For the multi-UAS path planning experiments, we have used the DEVS-FIRE (37) framework to simulate the fire spread scenarios as well. In this section, we will present the multi-UAS path planning results based on the first version of our multi-UAS path planning algorithm.

Figure 7.6a shows a simulated wildfire monitoring scenario where three UASs has been used to monitor the fire. Here, the green region of the cellspace represents the unburnt area of the wildland, the red-colored cells represent the currently burning cells, and the black-

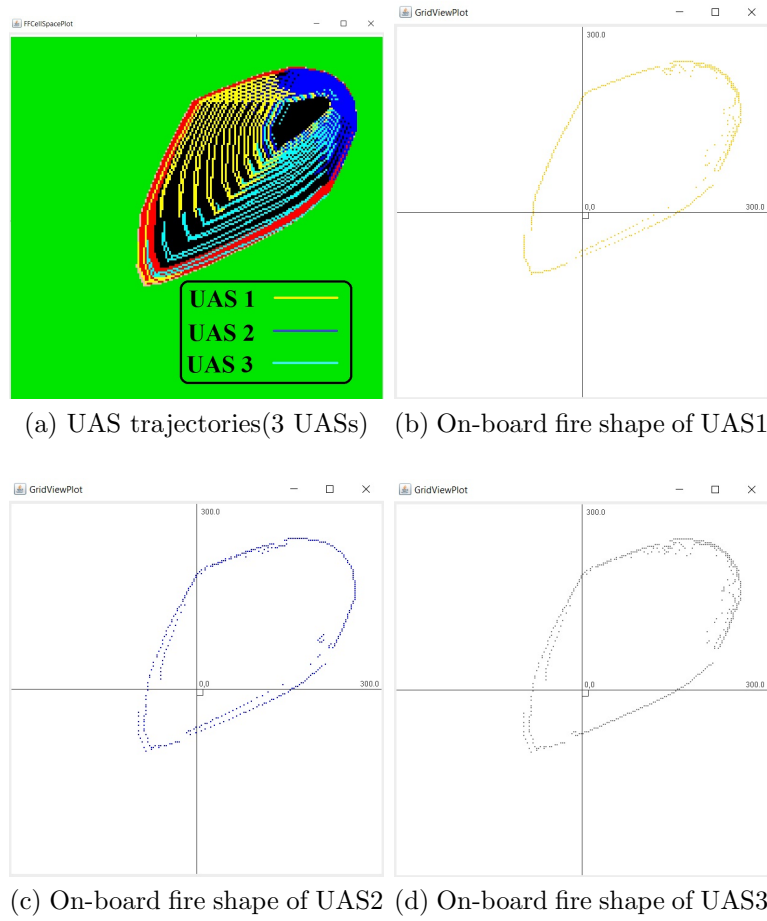
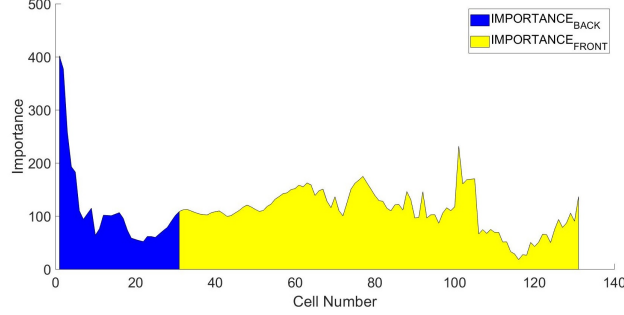


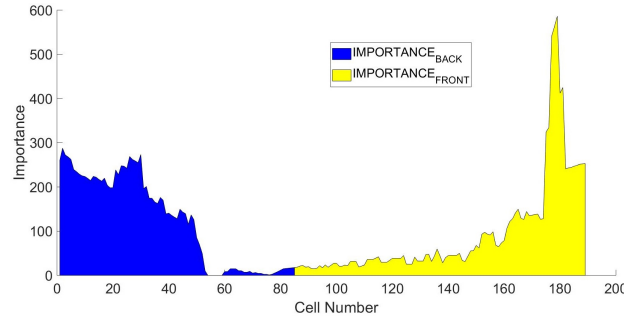
Figure 7.6: On-board fire shape generation

colored cells represent the already burnt-out cells. The trajectories of the three UASs have been marked by different colors as showed in the figure. At this particular time point of the monitoring, Figure 7.6b, Figure 7.6c, and Figure 7.6d shows the fire perimeter constructed from the on-board map data of the UAS1, UAS2, and UAS3 respectively. Though the UASs monitored different parts of the fire, due to communicating their location, they are able to generate a more accurate on-board fire shape. Based on these individual fire shapes, the UASs determine their respective back and front windows and compute their total importance

( $IMPORTANCE_{BACK}$  and  $IMPORTANCE_{FRONT}$ ).



(a) A scenario when the UAS should move forward



(b) A scenario when the UAS should turn back

Figure 7.7: Demonstration of importance-based decision making

Figure 7.7 demonstrates the  $IMPORTANCE_{BACK}$  and  $IMPORTANCE_{FRONT}$  for two possible decision scenarios (fly forward or turn back). First, Figure 7.7a shows an example where the UAS decides to move forward because the total importance in the front window is much larger. In contrast, Figure 7.7b shows a scenario where the total importance in the back window is larger and the UAS should change its flying direction. Please note that the length of the front and back window varies as the UASs move along the fire perimeter. Thus, depending on the total importance in the front and back window, the UASs decide their best flying direction (CW or CCW) along the fire perimeter. Figure 7.8 shows decentralized path planning results using 2, 3, and 4 UASs. The results are similar to the centralized importance-

based path planning approach presented in the work (16), but without requiring a ground station and boundary negotiation procedure between the adjacent UASs. Though there seems a gap between the trajectories of the adjacent UASs as can be seen in Figure 7.8c, the on-board fire maps always contain a complete fire shape as previously demonstrated in Figure 7.6.

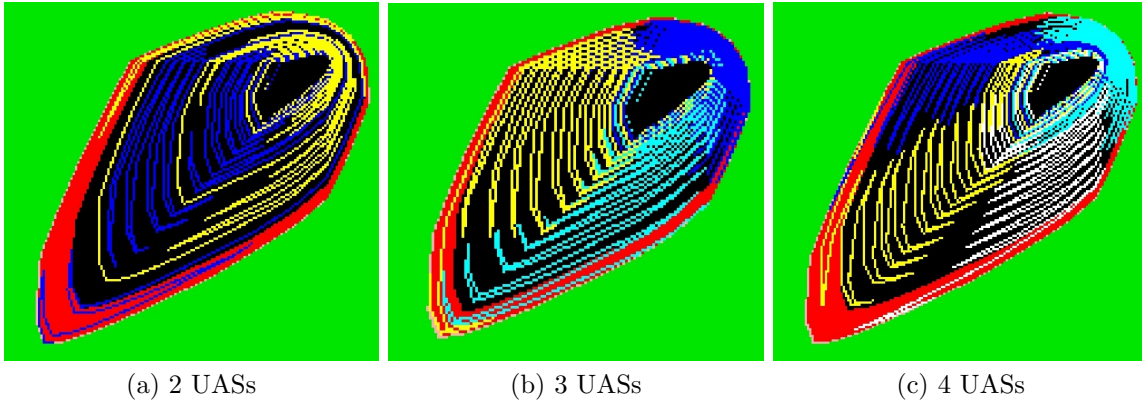


Figure 7.8: Path planning results for different numbers of UASs

## 7.2 Decentralized Importance-based Multi-UAS Path Planning Algorithm: A novel Back-and-Forth Coverage Strategy

One of the unique aspects of this work is the combination of importance-based path planning and decentralized computing for wildfire monitoring. Due to the variation in terrain, fuel, and weather conditions, different segments of the wildfire propagate with possibly different speeds. In importance-based path planning, more attention is provided towards the most active segment(s) of the fire. Please refer to Section 3.1 for a detailed description of how the importance is calculated and tracked onboard.

Decentralized computation in UAV path planning is necessarily the capability of UAVs

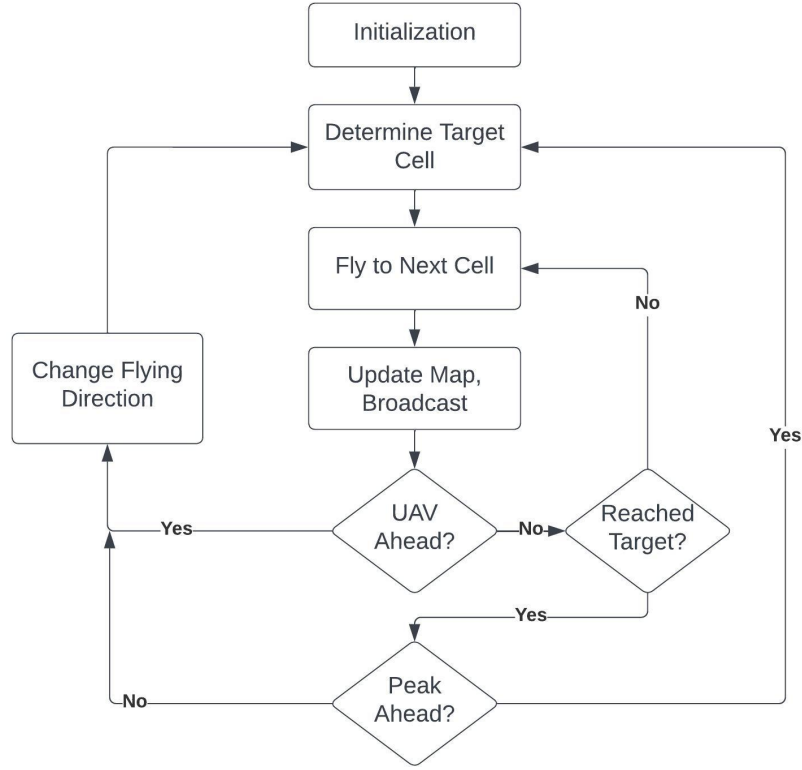


Figure 7.9: Overview of the algorithm

to operate without relying on a central controller. Many of the existing wildfire monitoring applications adopt a centralized approach where a ground station gathers the UAVs' states (location, direction, speed etc.) and provides the UAVs with the flying decisions. In decentralized approach, the computations required for the path planning are performed on-board by the UAVs themselves. In this approach, the UAVs might need to exchange some information (e.g. location, direction, etc.) among themselves for the coordination purpose.

To support decentralized computing in this work, the UAVs maintain an onboard map by utilizing their own observations that help to keep track of the fire propagation. The UAVs keep updating this map as they move from one cell to another. This map is based on 360

discrete angles around the ignition point (44). For each angle, the most recently visited cell (outer cell), second most recently visited cell (inner cell), and observation gap (Time) are tracked in the on-board map. The UAVs share their location with each other at regular intervals and cooperatively maintain their own copy of the onboard fire map that represents their best knowledge about fire propagation.

In this section, we provide a detailed description and visualization of the proposed algorithm based on back-and-forth coverage. We assume that the UASs are deployed at random locations of the fire boundary and they don't possess any prior knowledge about the fire spread. After the deployment, the path of the UASs are governed by the proposed algorithm. The steps of the algorithm have been illustrated in Fig. 7.9. First, the UASs perform an initialization procedure to estimate the current fire state. Next, the UASs adopt a back-and-forth coverage approach to monitor the fire. The length of the back-and-forth visits are dynamically determined based on the importance of the perimeter. In addition, the UASs share their location with other UASs in regular intervals to support decentralized path planning.

### **7.2.1 Initialization**

Once the UASs are deployed, they perform an initialization step to approximate the location of the current fire boundary. The goal of this step is to generate an initial onboard fire map based on the UASs' observations. For every angle  $\theta$ , the map contains the most recently observed cell  $(X_\theta, Y_\theta)$ , the second most recently observed cell  $(A_\theta, B_\theta)$ , and observation gap  $(T_\theta)$  for every angle around the ignition point as illustrated in Fig. 7.10. To carry out the

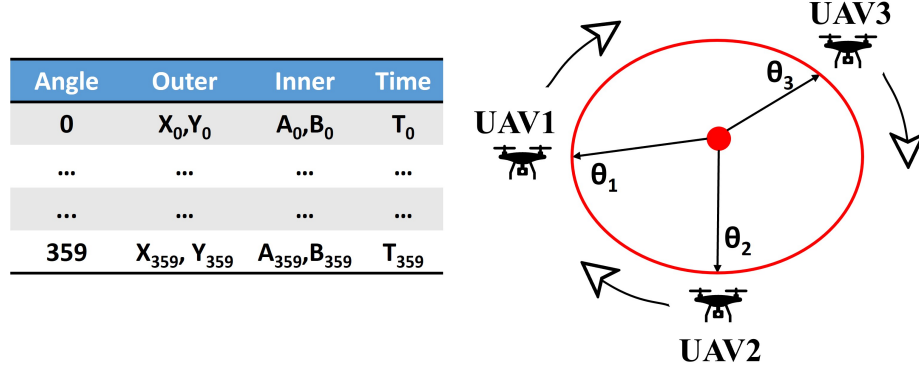


Figure 7.10: Initialization of the onboard map

initialization, the UAVs adopt a circling-based strategy - they follow the boundary in clockwise (CW) direction until the on-board map has information for every angle. For example, the UASs in Fig. 7.10 fly clockwise from their deployment angles  $\theta_i$  and keeps following the fire perimeter (showed by the red line) until the onboard map has information for every  $\theta$  ( $0 \leq \theta \leq 359$ ). While they are moving from one cell to another, they share their observation as discussed in Section 7.2.4. Thus, all the deployed UAVs complete their onboard map initialization cooperatively. Please note that this strategy works for the dynamic addition or removal of an UAS during the initialization process. If a new UAS is added, it needs to obtain the on-board map data generated so far from one of its neighboring UASs. Then, the remaining procedure will be the same. In case an UAS is removed during the initialization, no extra step is required - the other UASs continue the process as usual.

### 7.2.2 Back-and-forth Monitoring

After the initialization has been completed, the UASs adopt a back-and-forth coverage strategy to monitor the fire perimeter. In this strategy, an UAS starts to follow a segment of

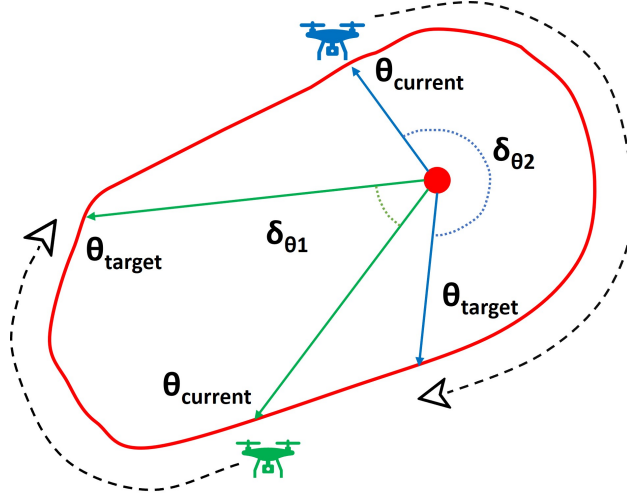


Figure 7.11: Scanning for the target angle

the fire boundary clockwise until its target angle is reached. After reaching that angle, the UAS changes its flying direction to counterclockwise and follows the boundary until the new target angle is reached, and so on. The target angle is dynamically determined based on the total importance of the fire boundary at the time of the decision-making. Assuming the UAS is at  $\theta_{current}$  angle relative to the ignition point and its current orientation is clockwise, a scan procedure is performed to find out the target angle  $\theta_{target}$  such that Equation 7.1 holds. The scanning process terminates once the target angle has been found or there is another UAS at the currently scanned angle. In the latter case, the last scanned angle is set as the target angle.

$$\sum_{\theta=\theta_{current}}^{\theta_{target}} imp_{\theta} \geq \sum_{\theta=0}^{359} imp_{\theta} \div S \quad (7.1)$$

Depending on the importance of the perimeter, the target angle can be found at different angular distances from the current location of the UAVs as illustrated in Fig. 7.11. If the

perimeter cells following the UAV has higher importance, the angular distance to find out the target angle could be smaller. Such a case has been illustrated in Fig. 7.11 where the green UAV finds the target cell after  $\delta_{\theta 1}$  angular distance. In contrast, The angular distance of the target angle could be larger when the fire spread following the UAV is not very fast. For example,  $\delta_{\theta 2}$  in Fig. 7.11 is comparatively larger than  $\delta_{\theta 1}$ . Thus, the back-and-forth coverage length is dynamically determined based on the importance of the fire perimeter.

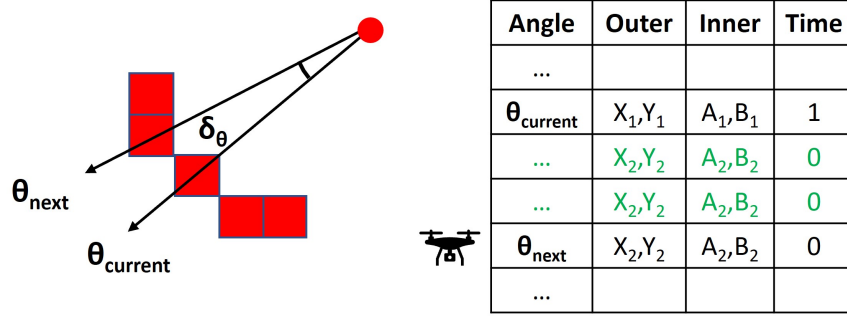


Figure 7.12: Updating the skipped angles

After the target angle has been determined, an UAV starts to fly toward that angle following the boundary. We assume that the UAV flies from the center of the current cell to the center of its neighboring perimeter cell. When it reaches the neighboring cell, the change in angular position ( $\delta_{\theta}$ ) of the UAV can vary by arbitrary amounts. This variation could be categorized into three types: first, the neighboring cell has the same angle as the current angle ( $\delta_{\theta}=0$ ), second, the neighboring cell has one degree difference from the current angle ( $\delta_{\theta}=1$ ), and third, it has more than one degree difference from the current angle ( $\delta_{\theta} \geq 2$ ). When  $\delta_{\theta} = 0$  or 1, the UAV just updates the on-board map record for the current angle ( $\theta$ ). However, when  $\delta_{\theta} \geq 2$ , i.e. the UAV has skipped some angles while flying to the

neighboring cell, the on-board map retains the older information for those skipped angles. To maintain an up-to-date map, the UAV updates the information of the skipped angles with the information of the current angle. This process has been demonstrated in Fig. 7.12 where  $\delta_\theta=3$ . Therefore, the on-board map information for two skipped angles is updated with the information available at  $\theta_{next}$ . Thus, the UAV maintains an up-to-date map while it is flying toward the target angle.

### ***7.2.3 Handling Special Scenarios***

Generally, an UAV will keep monitoring the fire boundary back-and-forth based on the dynamically determined target angles. However, it might change its coverage length in two special scenarios. First, when it is about to meet with another UAV it will reduce its current trip length. Second, when an UAV reaches the target angle and finds out that the peak(s) of the fire spread is close to the target angle, it will extend its trip to cover the peak(s).

#### ***7.2.3.1 Reduction of Trip Length***

When two UAVs are heading toward their target angles and they are about to meet each other from the opposite direction, it is better for them to cancel their current trip to avoid overlapping of coverage. Therefore, when two UAVs are about to meet each other they both change their flying direction, determine their new target angles, and start flying towards the new target.

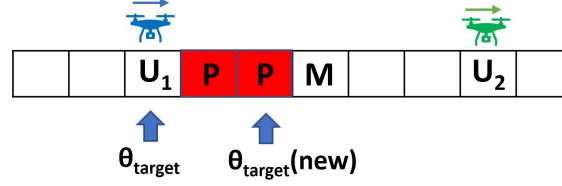


Figure 7.13: Extending the current trip

#### 7.2.3.2 Extension of Trip Length

When an UAV reaches the target angle and detects a significantly active fire region ahead, it extends its current trip to cover that region. To detect such a region, the UAV first identifies the peak(s) of the importance curve based on the Equation 3.4. Next, it checks if the peak(s) are near the current position of the UAV based on a dynamically determined location ( $M$ ) that represents the midpoint between the current UAV location and its neighboring UAV's location on the fire perimeter. As demonstrated in Fig. 7.13, if there are peaks( $P$ ) between the location of the UAV ( $U_1$ ) and the midpoint( $M$ ), the current UAV extends its trip until the last peak( $P$ ).

#### 7.2.4 Multi-UAS collaboration

The UAVs adopt a simple yet very effective approach to share their observation with each other and maintain a more accurate on-board fire map. When an UAV moves to a new cell it broadcasts its location to the other UAVs. Based on the receipt of the broadcast, the other UAVs update their onboard map at the corresponding angle. The process has been illustrated in Fig. 7.14. Considering three UAVs -  $UAV1(X_1, Y_1, \theta_1)$ ,  $UAV2(X_2, Y_2, \theta_2)$ ,  $UAV3(X_3, Y_3, \theta_3)$  where  $(X_i, Y_i)$  is their respective location and  $\theta_i$  is their respective angular position, the col-

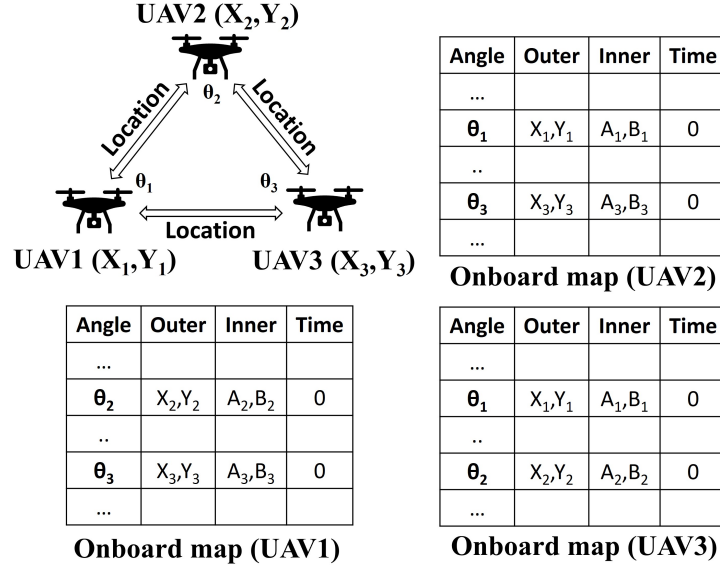


Figure 7.14: Inter-UAV collaboration

laboration has been showed in Fig. 7.14. For example, when the UAV1 receives the broadcast from UAV2 and UAV3, it updates the fire spread information at angle  $\theta_2$  and  $\theta_3$ . Thus, the UAVs share their observation with each other and maintain a more accurate knowledge of the fire spread.

### 7.2.5 Analysis of the Proposed Algorithm

For onboard computing, it is important to design algorithms that are optimized for efficient use of the limited processing speed and memory. Here, we briefly discuss the performance of the proposed algorithm in terms of time and space complexity.

#### 7.2.5.1 Time Complexity

Assuming the number of perimeter cells is  $N$  and the maximum number of skipped angles is  $\delta_{max}$ , The following table summarizes the time complexity:

Step	Complexity	Remarks
Initialization	$O(N)$	Scan the perimeter cells of length $N$
Determine $\theta_{target}$	$O(N)$	Scan the perimeter cells of length $N$
Update onboard map	$O(\delta_{max})$	Update skipped angles of length $\delta_{max}$
UAV ahead?	$O(1)$	Location of other UAVs are cached
Peak ahead?	$O(N)$	Scan the perimeter cells of length $N$
<b>Total</b>	<b><math>O(N)</math></b>	Overall time complexity sums to $O(N)$

#### 7.2.5.2 Space Complexity

Each UAV only needs to generate and maintain an onboard map of size  $360 \times 4$  as showed in Fig. 7.10. The size of this map remains the same regardless of the fire size. Thus, the proposed algorithm uses constant memory and its space complexity is  $O(1)$ .

To summarize, the proposed algorithm has linear time complexity and constant space complexity. Therefore, it is highly suitable for resource-constrained systems such as UAVs.

### 7.2.6 Experiment Results

For the experimentation, we have utilized the DEVS-FIRE (16) framework. UAVs are deployed at random locations over the fire perimeter at a designated time after ignition.

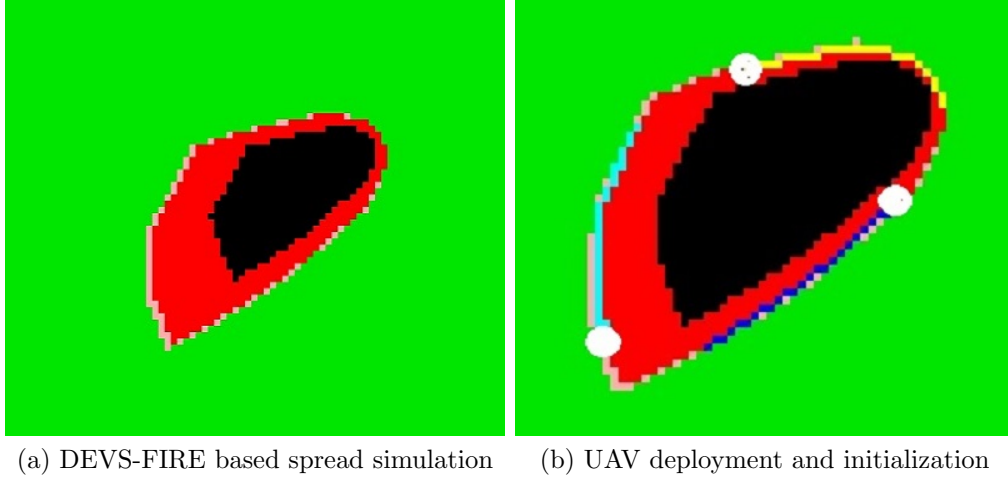


Figure 7.15: On-board map initialization

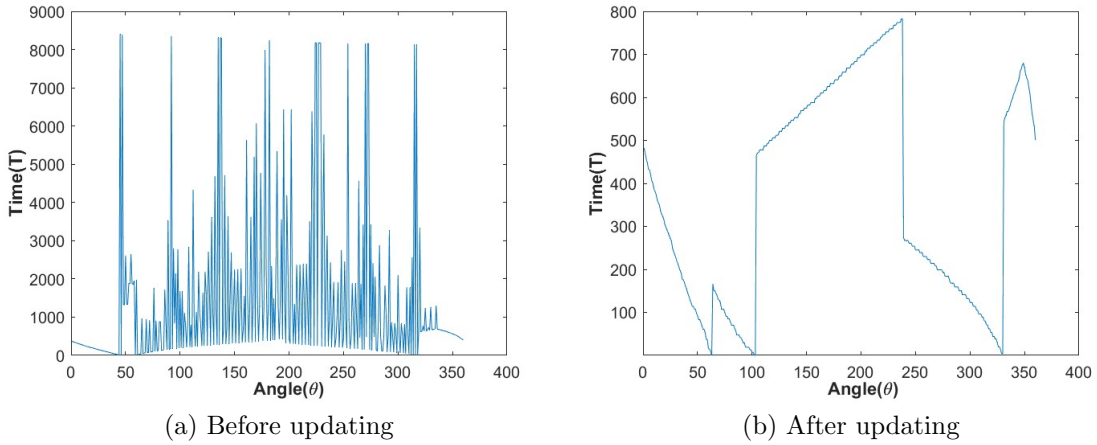


Figure 7.16: Maintaining up-to-date on-board map

A sample fire spread scenario simulated by DEVS-FIRE has been showed in Fig. 7.15a. Here, The green, red, and black regions represent the unburnt, currently burning, and burnt-

out regions of the wildland respectively. Fig. 7.15b shows the deployment locations of three UAVs (yellow, cyan, and blue) performing the circling-based initialization process. While the UAVs are flying from one cell to the next, it possibly skips a random number of angles as discussed in Section 7.2.2. Fig. 7.16 shows a plot of the observation gap time ( $T$ ) from the simulation experiment. Please note how some of the skipped angles in Fig. 7.16a have very high information gap time ( $T$ ) if skipped angles are not updated. By utilizing the smoothing procedure described in Section 7.2.2, the UAVs maintain consistent information about the fire spread across all angles as showed in Fig. 7.16b.

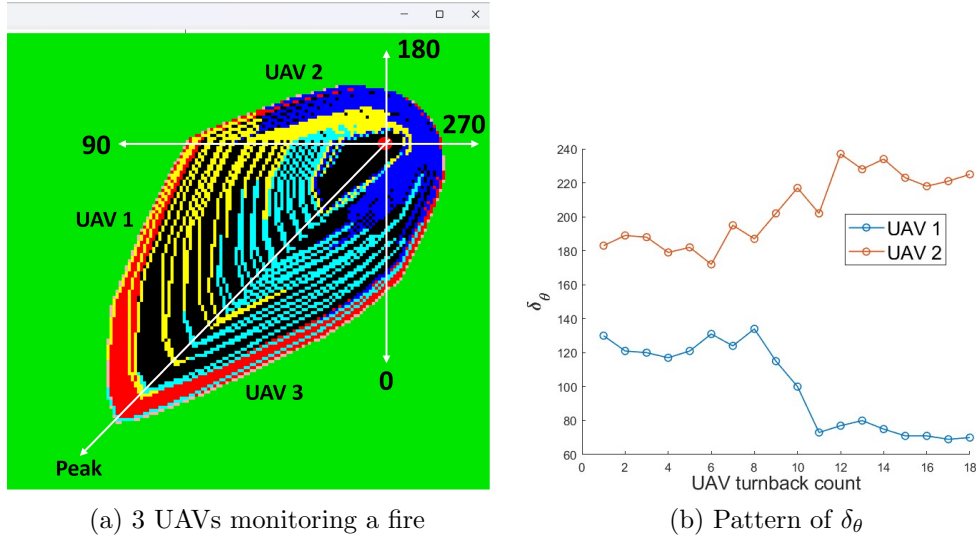


Figure 7.17: Back and forth perimeter coverage

Fig. 7.17a shows a snapshot from the simulation where 3 UAVs monitoring a fire. The ignition point has been marked by the red dot and the angles convention has been highlighted by the axes. The yellow, blue, and cyan lines represent the trajectory of UAV1, UAV2, and UAV3 respectively. In this monitoring scenario, the fire peak is towards 40 deg (approximately). From the trajectory, we can see that the UAV1 and UAV3 covered the fire front,

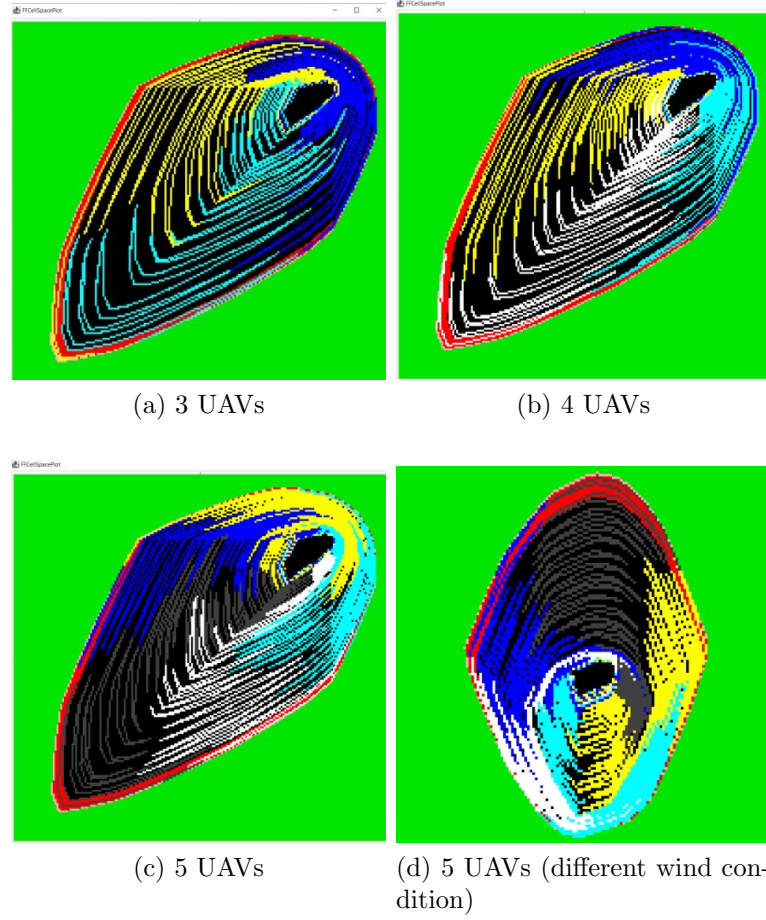


Figure 7.18: Path planning results based on the proposed algorithm

while the UAV2 covered the back of the fire. As the fire propagated, the total importance in the peak region increased compared to the back of the fire. Consequently, the  $\delta_\theta$  gradually decreased for the UAVs covering the peak of the fire (UAV1 and UAV3) and increased for the UAV covering the back of the fire (UAV2) as showed in Fig. 7.17b. Path planning results for different number of UAVs and different wind condition has been presented in Fig. 7.18 where simulation has been run for 6 hours for each experiment. The back-and-forth coverage trajectory of the UAVs has been showed by different colors. From their trajectories, we can

see how the UAVs covered the growing fire perimeter based on dynamic back-and-forth coverage approach. Compared to the importance-based and centralized path planning approach presented in (16), the proposed algorithm is also able to provide adequate coverage; however without relying on a ground station.

### **7.3 Comparison Between the two Versions of the Multi-UAS Path Planning Algorithm**

Based on the highly promising results from our single-UAS path planning algorithm, we first designed the multi-UAS path planning algorithm based on the path planning window concepts (front and back window). This first version of the multi-UAS path planning algorithm performs well in most scenarios 7.1.3, however, the performance of the algorithm is limited in some special cases. This leads us to think further and come up with the second version of the algorithm that addresses the limitations of the first version. Below we list the cases when the first version might have undesirable performance and discuss how the second version handles those cases.

#### ***7.3.1 Initialization Procedure***

During the initialization phase, the first version of the algorithm requires each UAS to complete a full circle along the fire perimeter. This process is optimized in the second version - the UASs only need to visit a segment of the perimeter. In addition, the second version considers the dynamic addition or removal of an UAS during the initialization process, which was not taken into account in the first version of the multi-UAS path planning algorithm.

In summary, the second version of the algorithm comes with an improved and adaptive initialization procedure.

### ***7.3.2 Rapid Fire Spread on the Back Window***

In the first version, the UAS doesn't commit to reach a specific cell, thus it might change its flying direction at any cell. Therefore, at any time, if the fire on the back window starts to spread rapidly, it will change its flying direction. This might leave an important segment from the front window unvisited. Consequently, this leads to monitoring gaps over time - which is not desired for a perimeter segment that is adequately active.

To address this issue, the second version of the algorithm adopts a back-and-forth coverage strategy. Based on this strategy, an UAS commits to reach a target cell ( $\theta_{target}$ ) unless there is a possible collision scenario (i.e. another UAS is ahead of it). Thus, the second approach minimizes the monitoring gaps and provides better performance.

### ***7.3.3 Fire Peaks Might Not Get Proper Attention***

The length of the front and back windows might vary significantly on the first approach. For instance, the front window might consist of a few cells while the back window can consist of dozens of cells. In such a case, if a fire peak is present on the shorter window, it may not be visited by the UAS due to the significantly larger size of the other window. From the fire monitoring perspective, this behavior is not very desirable because peak segments are very important to be visited. The second version of the multi-UAS path planning algorithm meets this requirement by integrating a scan procedure 7.2.3.2. When an UAS reaches its

target cell, it scans the perimeter and identifies the fire peaks based on the on-board map data. If one or more peaks are found to be nearby the UAS, it extends its trip such that those peaks get visited by the UAS.

#### ***7.3.4 Computational Efficiency***

Performance is an important aspect for any algorithm that runs on resource-constrained systems such as the UASs. Even though both algorithms have similar space complexity, the second version outperforms the first version in terms of time complexity. Assuming the number of perimeter cells at a given time is  $N$ , the first version has a time complexity of  $O(N^2)$  while the second version has a time complexity of  $O(N)$ . This is because the first version performs the decision-making whenever it moves from one cell to the other, while the second version performs the decision making only when it reaches the target cell. Thus, the second version of the multi-UAS path planning algorithm is more efficient computationally.

## CHAPTER 8

### CONCLUSIONS AND FUTURE WORK

In this work, we presented novel algorithms for real-time on-board path planning intended to be used for UAS-based wildfire monitoring. Our goal is to make the wildfire monitoring task important-based and decentralized. Real-time and on-board 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. In addition, an in-depth analysis of the single-UAS algorithm has been presented which provides some important insights into the importance-based path planning and decision-making. For the multi-UAS path planning, autonomous decentralized coordination between the UASs has been achieved based on a location sharing scheme. Based on the importance-based path planning and decentralized computing frameworks, two different path planning algorithms have been presented for multi-UAS and their performance has been evaluated. Simulation results for a wide range of fire spread scenarios have been presented to show the effectiveness of the proposed algorithms in different conditions.

It is highly essential to take into account the uneven fire propagation in UAS path planning to make the best use of available UASs. In addition, the decentralized approach in wildfire monitoring makes the algorithm suitable for challenging scenarios. In this work, we have focused on both these needs and proposed novel approaches for autonomous UAS path

planning. The path planning algorithms presented in this work can perform more advanced wildfire monitoring tasks and hold the potential to enhance wildfire management strategies. In the future, we plan to implement the developed algorithm on physical UAVs and conduct experiments on real wildfire scenarios. Moreover, the proposed frameworks and algorithms have the potential to be adapted for monitoring other similar phenomena such as oil spills, floods, volcanic eruptions, and so on. We will consider exploring these possibilities in our future work.

## 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] Total wildland fires and acres (1983-2022). *National Interagency Fire Center*. Available at <https://www.nifc.gov/fire-information/statistics/wildfires>.
- [3] Wildland fires: A historical perspective. *U.S. Fire Administration TOPICAL FIRE RESEARCH SERIES*, 1, 2001. Available at <https://nfa.usfa.fema.gov/downloads/pdf/statistics/v1i3-508.pdf>.
- [4] Tauã M. Cabreira, Lisane B. Brisolara, and Paulo R. Ferreira Jr. Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1), 2019.
- [5] H. Zhang, B. Xin, Lh. Dou, et al. A review of cooperative path planning of an unmanned aerial vehicle group. In *Front Inform Technol Electron Eng* 21, page 1671–1694, 2020.
- [6] Patrick Nolan, Derek A. Paley, and Kenneth Kroeger. Multi-uas path planning for non-uniform data collection in precision agriculture. In *2017 IEEE Aerospace Conference*, pages 1–12, 2017.
- [7] Fotios Balampanis, Ivan Maza, and Anibal Ollero. Spiral-like coverage path planning for multiple heterogeneous uas operating in coastal regions. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 617–624, 2017.
- [8] Jeong H. Lee, Jeong S. Choi, Beom H. Lee, and Kong W. Lee. Complete coverage path

- planning for cleaning task using multiple robots. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 3618–3622, 2009.
- [9] Alexander Jahn, Reza Javanmard Alitappeh, David Saldaña, Luciano C. A. Pimenta, André Gustavo dos Santos, and Mario Fernando Montenegro Campos. Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 273–278, 2017.
- [10] Vitor M. G. Barros and Douglas G. Macharet. Multiple dynamic perimeter surveillance by an autonomous robot. In *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, pages 1–6, 2020.
- [11] M. Kemp, A.L. Bertozzi, and D. Marthaler. Multi-uuv perimeter surveillance. In *2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No.04CH37578)*, pages 102–107, 2004.
- [12] D.W. Casbeer, R.W. Beard, T.W. McLain, Sai-Ming Li, and R.K. Mehra. Forest fire monitoring with multiple small uavs. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 3530–3535 vol. 5, 2005.
- [13] David Casbeer, Derek Kingston, Randal Beard, and Tim McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *Int. J. Systems Science*, 37:351–360, 05 2006.
- [14] Steven G. Xu, Seunghyun Kong, and Zohreh Asgharzadeh. Wildfire detection using

- streaming satellite imagery. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 2899–2902, 2021.
- [15] Kyle D. Julian and Mykel J. Kochenderfer. Image-based guidance of autonomous aircraft for wildfire surveillance and prediction, 2018.
- [16] Xiaolin Hu, John Bent, and Jiawei Sun. Wildfire monitoring with uneven importance using multiple unmanned aircraft systems. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1270–1279, 2019.
- [17] Luis Merino, Fernando Caballero, J. Ramiro Martinez-de Dios, Ivan Maza, and Anibal Ollero. An unmanned aircraft system for automatic forest fire monitoring and measurement. *Journal of Intelligent and Robotic Systems*, 65:533–548, 01 2012.
- [18] Seonghyun Kim, Wonjae Lee, Young-su Park, Hyun-Woo Lee, and Yong-Tae Lee. Forest fire monitoring system based on aerial image. In *2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 1–6, 2016.
- [19] Constantinos Heracleous, Panayiotis Kolios, and Christos G. Panayiotou. An integrated uav system for wildfire evolution monitoring and data acquisition. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1131–1138, 2022.
- [20] Kostas Alexis, George Nikolakopoulos, Anthony Tzes, and Leonidas Dritsas. *Coordination of Helicopter UAVs for Aerial Forest-Fire Surveillance*, pages 169–193. 01 2009.
- [21] P.B. Sujit, Derek Kingston, and Randy Beard. Cooperative forest fire monitoring using multiple uavs. In *2007 46th IEEE Conference on Decision and Control*, pages 4875–

4880, 2007.

- [22] Rafael Bailon-Ruiz, Simon Lacroix, and Arthur Bit-Monnot. Planning to monitor wild-fires with a fleet of uavs. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4729–4734, 2018.
- [23] Gang Lei, Min-zhou Dong, Tao Xu, and Liang Wang. Multi-agent path planning for unmanned aerial vehicle based on threats analysis. In *2011 3rd International Workshop on Intelligent Systems and Applications*, pages 1–4, 2011.
- [24] Diego Pasini, Charles Jiang, and Marie-Pierre Jolly. Uav and ugv autonomous cooperation for wildfire hotspot surveillance. In *2022 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pages 1–5, 2022.
- [25] Esmaeil Seraj and Matthew Gombolay. Coordinated control of uavs for human-centered active sensing of wildfires. In *2020 American Control Conference (ACC)*, pages 1645–1652, 2020.
- [26] Khaled A. Ghamry and Youmin Zhang. Cooperative control of multiple uavs for forest fire monitoring and detection. In *2016 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6, 2016.
- [27] K.S. Kappel, T.M. Cabreira, J.L. Marins, et al. Strategies for patrolling missions with multiple uavs. In *J Intell Robot Syst 99*, page 499–515, 2020.
- [28] L. Feng and J. Katupitiya. Radial basis function-based vector field algorithm for wildfire boundary tracking with uavs. In *Comp. Appl. Math. 41, 124*, 2022.
- [29] Derek Kingston, Randal W. Beard, and Ryan S. Holt. Decentralized perimeter surveil-

- lance using a team of uavs. *IEEE Transactions on Robotics*, 24(6):1394–1404, 2008.
- [30] Huy X. Pham, Hung M. La, David Feil-Seifer, and Matthew Deans. 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)*, pages 6648–6653, 2017.
- [31] Zhekun Cheng, Liangyu Zhao, and Zhongjiao Shi. Decentralized multi-uav path planning based on two-layer coordinative framework for formation rendezvous. *IEEE Access*, 10:45695–45708, 2022.
- [32] Derek Kingston, Randal W. Beard, and Ryan S. Holt. Decentralized perimeter surveillance using a team of uavs. *IEEE Transactions on Robotics*, 24(6):1394–1404, 2008.
- [33] Johana M. Florez-Lozano, Fabio Caraffini, Carlos Parra, and Mario Gongora. A robust decision-making framework based on collaborative agents. *IEEE Access*, 8:150974–150988, 2020.
- [34] Fatemeh Afghah, Abolfazl Razi, Jacob Chakareski, and Jonathan Ashdown. Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles, 04 2019.
- [35] S M Towhidul Islam and Xiaolin Hu. Real-time on-board path planning for uas-based wildfire monitoring. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 527–535, 2021.
- [36] Lewis Ntamo, Xiaolin Hu, and Yi Sun. Devs-fire: Towards an integrated simulation environment for surface wildfire spread and containment. *SIMULATION*, 84(4):137–155, 2008.

- [37] Xiaolin Hu, Yi Sun, and Lewis Ntaimo. Devs-fire: design and application of formal discrete event wildfire spread and suppression models. *SIMULATION*, 88(3):259–279, 2012.
- [38] Bernard Zeigler, Herbert Prähofer, and Tag Gon Kim. Theory of modeling and simulation 2nd edition. 01 2000.
- [39] R.C. Rothermel. *A Mathematical Model for Predicting Fire Spread in Wildland Fuels*. USDA Forest Service research paper INT. Intermountain Forest & Range Experiment Station, Forest Service, U.S. Department of Agriculture, 1972.
- [40] Feng Gu, Xiaolin Hu, and Lewis Ntaimo. Towards validation of devs-fire wildfire simulation model. pages 355–361, 01 2008.
- [41] N Dahl, H Xue, X Hu, and M Xue. Coupled fireatmosphere modeling of wildland fire spread using devs-fire and arps. *Natural Hazards*, 77(2):1013–1035, 2015.
- [42] H. E. Anderson. Aids to determining fuel models for estimating fire behavior. volume 122. US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station., 1982.
- [43] Franco P. Preparata and Michael I. Shamos. *Convex hulls: Basic algorithms. In Computational Geometry: An Introduction*. Springer-Verlag, Berlin, Heidelberg, 1985.
- [44] S M Towhidul Islam and Xiaolin Hu. Towards decentralized importance-based multi-uas path planning for wildfire monitoring. In *2022 17th Annual System of Systems Engineering Conference (SOSE)*, pages 67–72, 2022.