UNIVERSITY OF BIRMINGHAM

Managing Distributed Situation Awareness in a Team of Agents

by

Sagir Muhammad Yusuf

A thesis submitted to

The University of Birmingham

For the degree of

DOCTOR OF PHILOSOPHY

School of Computer Science

The University of Birmingham

July 2022

ABSTRACT

The research presented in this thesis investigates the best ways to manage Distributed Situation Awareness (DSA) for a team of agents tasked to conduct search activity with limited resources (battery life, memory use, computational power, etc.). In the first part of the thesis, an algorithm to coordinate agents (e.g., UAVs) is developed. This is based on Delaunay triangulation with the aim of supporting efficient, adaptable, scalable, and predictable search. Results from simulation and physical experiments with UAVs show good performance in terms of resources utilisation, adaptability, scalability, and predictability of the developed method in comparison with the existing fixed-pattern, pseudorandom, and hybrid methods. The second aspect of the thesis employs Bayesian Belief Networks (BBNs) to define and manage DSA based on the information obtained from the agents' search activity. Algorithms and methods were developed to describe how agents update the BBN to model the system's DSA, predict plausible future states of the agents' search area, handle uncertainties, manage agents' beliefs (based on sensor differences), monitor agents' interactions, and maintains adaptable BBN for DSA management using structural learning. The evaluation uses environment situation information obtained from agents' sensors during search activity, and the results proved superior performance over well-known alternative methods in terms of situation prediction accuracy, uncertainty handling, and adaptability. Therefore, the thesis's main contributions are (i) the development of a simple search planning algorithm that combines the strength of fixed-pattern and pseudorandom methods with resources utilisation, scalability, adaptability, and predictability features; (ii) a formal model of DSA using BBN that can be updated and learnt during the mission; (iii) investigation of the relationship between agents search coordination and DSA management.

DEDICATION

For my parents and me.

ACKNOWLEDGEMENTS

My gratitude to Prof. Chris Baber for his patience, advice, support, encouragement, and guidance throughout my doctoral journey. Indeed, I will never forget the impact of his contributions to the progress of my research. I am also grateful to Dr. Mohan Sridharan and Bob Hendley for their valuable comments and feedback. My special thanks to Nigeria's Petroleum Technology Trust Fund (PTDF) for the sponsorship and support.

To my mum and dad, thank you so much for the prayers, support, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

Abbreviation	Full Term
SA	Situation Awareness
DSA	Distributed Situation Awareness
BBN	Bayesian Belief Network
EM	Expectation-Maximisation
GD	Gradient Descent
DCOP	Distributed Constraint Optimisation
DCOPs	Distributed Constraint Optimisation Problems
MGM	Maximum Gain Message
DPOP	Distributed Pseudotree OPtimisation
UAV	Unmanned Aerial Vehicle
SME	Subject Matter Expert
SOP	Standard Operating Procedure
API	Application Programming Interface
IDE	Integrated Development Environment
PC	Picture Compiler
CI	Conditional Independence
GMM	Gaussian Mixture Model
GP	Gaussian Process
XML	eXtensible Markup Language
MVC	Model-View -Controller
CSV	Comma-Separated Values
ACF	Auto-Correlation Function
PCF	Partial auto-Correlation Function

GLOSSARY

Term	Definition
Resources	Parameters possessed by agents or missions that define their operational success e.g., battery, processing power, etc. for UAVs.
Parameters	Characterisation variables possessed by an agent, mission, or search area e.g., current location, battery level, computational power, etc.
Search area	A bounded space to be explored by the agents.
Phenomenon	Search area parameters e.g., trees, buildings, wind speed, wind direction.
Coverage	The portion of search area explored, i.e., the portion of the search area with agents' sensing.
Path divergence	Measure of how the search path is spread across the search area.
Protocols	Set of rules that control the algorithm's outcome.
Team of UAVs	A group of UAVs, mostly up to four.
Paths elements	Controllable parts of a path plan, these are: angles, edges, and quadrants.
Method	It depends on the context. It means the way of doing the thing, e.g., search method refers to search plan generation algorithm.
Search plan	Set of waypoints generated to explore an area.
Search mission or search	Exploration of a search area to detect a prescribed target.
Waypoint	Geodetic location within a search area. This can be referenced using longitude and latitudes or planar coordinates.
Predictability	Ability to predict the future situation e.g., for search mission, it is an ability to predict the location of the agents.
Scalability	Ability of an algorithm to process large number of agents or data with a stable resource.
Adaptability	The ability of an algorithm to be flexible based on various inputs e.g., given varying tasks, data, etc.
SA modelling tool	A tool used to model Situation Awareness of the system based on the agents' information.

Knowledge	Agents acquired phenomena information based on sensor states.
Automation	Non-human agents, e.g., UAVs, that relied on algorithms for their task performance
Distributed agents	Agents with the capability to act independently, however, there could be a presence of sparse interaction based on proximity (limited sensor range) with other agents. Thus, distribution in this thesis refers to a partial distribution.
Constraint	Imposed system limitations e.g., for UAVs, these can be limited battery, communication range, processing power, etc.
Dynamic	Characterised by changing values overtime e.g., a dynamic search area has a changing wind speed, wind direction, fuel types, etc. configuration from one location to another over time.
Information	Agents gathered sensor phenomena states values based on sensor outcome
Coordination	Effective generation of search plans for multiple agents e.g., to avoid redundant search, etc, that control the agents search mission
Belief	Agent's verified sensor information
Interaction	Information exchange among agents
Fixed-pattern	Agents waypoint generation using defined geometric shapes e.g., sector search
Simple agents	Low-capicity automation agents specifically mini-or micro UAVs based on classification in Chapter 1 Table 1
Picture compilers	Medium-capacity agents responsible for managing specific group of information (gathered by the simple agents) for a specific concept understanding.
Host	High-capacity agents that can manage the system information
Critical weight	SMEs weight assignments for information importance
Degree of relevance	The degree of relations among two nodes
Environment	Depends on the context.

Thesis Publication

This thesis development leads to the following publications.

1. Yusuf S.M and Baber.C., 2021. Formalizing Distributed Situation Awareness in Multi-Agent Networks. IEEE transactions in Human-Machine System**. Journal. Published**

   Contributions: SA modelling using Bayesian Belief Network

   i. Yusuf and Baber developed model.
   ii. Yusuf implemented model.
   iii. Yusuf designed and implemented experiments.
   iv. Yusuf analysed the results.
   v. Yusuf writes the paper.
   vi. Yusuf and Baber revised the paper.

2. Yusuf S.M and Baber.C., 2022. DIMASS: a Delaunay-Inspired Multi-agent Search Strategy (DIMASS), Hybrid Approach to a Team of Agents Search Strategy. Frontiers in AI and Robotics. **Journal. Accepted**

   Contributions: Agents Efficient Search Algorithms Development

   i. Yusuf developed and implemented the model.
   ii. Yusuf designed and implemented experiments.
   iii. Yusuf analysed the results.
   iv. Yusuf writes the paper.
   v. Yusuf and Baber revised the paper.

3. Yusuf S.M Baber.C., 2022. Distributed Situation Awareness for Multi-agent Mission in Dynamic Environments: A Case of Multi-UAVs Wildfires Searching. Presented at the 35th International Conference of the Association for the Advancement of Artificial Intelligence (AAAI), Doctoral Consortium Section, Vancouver, Canada. **Conference**. Published

   Contributions: Problem definition and methodology

   i. Yusuf and Baber defined the problem and methodology.
   ii. Yusuf designed and implemented the experiments.
   iii. Yusuf writes the paper.

iv. Yusuf presented the paper.
v. Yusuf and Baber revised the paper.

4. Yusuf S.M and Baber.C., 2020. Conflicts Resolution and Situation Awareness in Heterogeneous Multiagent Missions using Publish-subscribe Technique and Inferential Reasoning. Presented at 12th International Conference on Agents and Artificial Intelligence (ICAART), Valetta, Malta. **Conference.** Published

   Contributions: Agents sensor states conflicts handling

   i. Yusuf and Baber developed and implemented the model.
   ii. Yusuf designed and implemented experiments.
   iii. Yusuf analysed the results.
   iv. Yusuf writes the paper.
   v. Yusuf presented the paper.
   vi. Yusuf and Baber revised the paper.

5. Yusuf S.M and Baber.C., 2020. Handling Uncertainties in Distributed Constraint Optimisation Problems using Inferential Reasoning. Presented at 12th International Conference on Agents and Artificial Intelligence (ICAART), Valetta, Malta. **Conference.** Published

   Contributions: Agents collected sensor information uncertainty handling and predictions

   i. Yusuf developed and implemented the model.
   ii. Yusuf designed and implemented experiments.
   iii. Yusuf analysed the results.
   iv. Yusuf writes the paper.
   v. Yusuf presented the paper.
   vi. Yusuf and Baber revised the paper.

6. Yusuf S.M and Baber.C., 2020. Inferential Reasoning in Dynamic and Uncertain Distributed Constraint Optimisation Problem: A Case Study of Multi-UAV Mission for Forest Lookouts. Presented at 12th International Conference on Agents and Artificial

Intelligence (ICAART), Valetta, Malta. Open Communication. Published in the book of abstract

Contributions: Agents' resources utilisation formalisation with the Distributed Constraint Optimisation.

vii. Yusuf developed and implemented the model.
viii. Yusuf designed and implemented experiments.
ix. Yusuf analysed the results.
x. Yusuf writes the paper.
xi. Yusuf presented the paper.
xii. Yusuf and Baber revised the paper.

7. Yusuf S.M and Baber.C., 2020. Human-agents Interactions in Multi-agent Systems: A Case Study of Human-UAVs Team for Forest Fire Lookouts. Presented at 12th International Conference on Agents and Artificial Intelligence (ICAART), Valetta, Malta. **Conference.** Published

   Contributions: Agents role specifications and interaction architecture

   i. Yusuf developed and implemented the model.
   ii. Yusuf designed and implemented experiments.
   iii. Yusuf analysed the results.
   iv. Yusuf writes the paper.
   v. Yusuf presented the paper.
   vi. Yusuf and Baber revised the paper.

8. Sagir M. Yusuf and Baber.C., 2020. Probabilistic Approach of Dealing with Uncertainties in Distributed Constraint Optimisation Problems and Situation Awareness for Multi-agent Systems. Presented at 14th International Conference on Multi-Agent Systems and Robotics(ICMASR), London, United Kingdom. **Conference.** Published, Best Presentation Award

   Contributions: Formalising Distributed Situation Awareness (DSA) with Bayesian Network and Learning

i. Yusuf and Baber developed and implemented the model.
ii. Yusuf designed and implemented experiments.
iii. Yusuf analysed the results.
iv. Yusuf writes the paper.
v. Yusuf presented the paper.
vi. Yusuf and Baber revised the paper.

9. Sagir M. Yusuf and Baber.C., 2020. Multiagent Searching Adaptation using Lévy Flight and Inferential Reasoning. Presented at 14th International Conference on Multi-Agent Systems and Robotics(ICMASR), London, United Kingdom. **Conference.** Published

   Contributions: Multi-agent search methods performance analysis

   i. Yusuf and Baber developed and implemented the model.
   ii. Yusuf designed and implemented experiments.
   iii. Yusuf analysed the results.
   iv. Yusuf writes the paper.
   v. Yusuf presented the paper.
   vi. Yusuf and Baber revised the paper.

10. Sagir M. Yusuf and Baber.C., 2020. Inferential Reasoning for Heterogeneous Multi-agent Mission. Presented at 14th International Conference on Multi-Agent Systems and Robotics(ICMASR), London, United Kingdom. **Conference.** Published, Best Paper Award

    Contributions: Agents' sensor heterogeneity management

    i. Yusuf and Baber developed and implemented the model.
    ii. Yusuf designed and implemented experiments
    iii. Yusuf analysed the results.
    iv. Yusuf writes the paper.
    v. Yusuf presented the paper.
    vi. Yusuf and Baber revised the paper.

1 Chapter 1 Introduction

This chapter introduces the use case for this thesis: a distributed team of Unmanned Aerial vehicles (UAVs) is tasked to perform forest fire monitoring. For the use case and the thesis, the key questions relate to how this team is able to coordinate its search activity and maintain Situation Awareness, even when communications between team members is limited. This use case highlights the challenges of Distributed Situation Awareness (DSA) and how factors such as dynamic parameters (e.g., wind speed, wind direction, fuel types, etc.), limited resources (time, energy, computational power, etc.), and system constraints can affect the management of DSA within the system. The use case frames the research questions for the thesis.

1.1 The Choice of Use Case: Forest Fire Monitoring

Forest fire is still a big problem for humanity resulting in loss of lives and properties (Bjurling et al., 2020; Ingle, 2011; International Forest Fire News, 2006; Peter Hirschberger, 2016; S. Wang et al., 2021). For instance, in 2020 alone, the US recorded 58,950 forest fire cases with over 10.1 million acres burned and loss of properties worth over 7 billion US dollars[1]. To address this challenge, constant forest monitoring (searching for the fire and general SA management) is required. Thus, addressing this challenge requires a dynamic search area exploration. The challenge becomes more complicated when automation agents (e.g., UAVs) are tasked to conduct the mission instead of purely humans. Application of automation agents to address the challenge proved to be more effective in terms of cost and risk management with limitations of coordination and SA management challenges (Bailon-Ruiz et al., 2022; Bjurling et al., 2020; Bouguettaya et al., 2022; Casbeer et al., 2005; Cummings et al., 2007; Ghamry and Zhang, 2016; Merino et al., 2006.; Mohd Daud et al., 2022; Ozkan and Kilic, 2022; Rabinovich et al., 2018; Rocha et al., 2022; Zhou et al., 2018). Thus, these outlined challenges (i.e., distributed agents coordination and DSA management) are the thesis main motivations.

[1]https://www.iii.org/fact-statistic/facts-statistics-wildfires#:~:text=2020%3A%20In%202020%20there%20were,4.7%20million%20acres%20in%202019.

It is proposed that, while the thesis use-case is specific, features of the example applies to various domains, e.g., missing person finding, search and rescue, disaster management, business monitoring, etc. (Bevacqua et al., 2015; Cooper, 2020; Drew, 2021; Kanistras et al., 2013; Ozkan and Kilic, 2022; Weick, 1995).

The primary sources of forest fires are meteorological, e.g., lightning and thunderstorm, mechanical, e.g., trains, and human activities, e.g., campfires (Alkhatib, 2014; Ingle, 2011; Smith, 2017). Thus, predicting the occurrence of a fire is challenging due to the random behaviour of these sources. As such, effective constant search must be maintained to allow early detection for an effective response (Ingle, 2011; Marjovi et al., 2009; Vincent and Rubin, 2004). The traditional solution relied on trained rangers (Smith, 2017). This evolved gradually to horse-based search and trees/towers observation (Figure 1), satellite imaging, helicopters, and UAVs approach (Bailon-Ruiz et al., 2022; Bouguettaya et al., 2022;Alkhatib, 2014; Baek and Lim, 2018; Bouvry et al., 2016, 2016; Breejen et al., 1998; Casbeer et al., 2005; Chuvieco et al., 2019; Ghamry and Zhang, 2016; Ingle, 2011; Peter Hirschberger, 2016; Rocha et al., 2022; Smith, 2017). Helicopter and satellite imaging methods offer broader coverage than UAVs. However, these approaches are costly, and satellite imaging has some delays during information processing (Chuvieco et al., 2019). Due to these limitations, surveillance cameras have been applied (Breejen et al., 1998) with the main limitation of occlusion and lack of manoeuvrability. More recently, UAVs have been applied to tackle the problem (Casbeer et al., 2005; Ghamry and Zhang, 2016; Haksar and Schwager, 2018; Merino et al., 2006). UAVs offer cheaper solutions, better manoeuvrability, and ease of operations (Bjurling et al., 2020; Casbeer et al., 2005; Ghamry and Zhang, 2016; Merino et al., 2006; Ozkan and Kilic, 2022); however, they are constrained by limited resources, coordination challenges, and DSA management (Alkhatib, 2014; Baek and Lim, 2018; Casbeer et al., 2005; Chuvieco et al., 2019; Ghamry and Zhang, 2016; Khan et al., 2015; Merino et al., 2010, 2006).

Figure 1: Fire Searching using Towers (Photo by: Lookout handbook)(US Department of the Interior, National Park Services).

1.2 Use Case Specifications

The thesis use-case assumes a team of distributed agents. The specification of the agents align with the definitions in Table 1. Micro or mini UAVs (classified by the UK Ministry of Defence[2]) are responsible for generating the system's information based on the mounted sensors; Picture Compilers (PCs) are responsible for specific information organisation and comprehension, e.g., weather control; the host performs mission management; and specialised Subject Matter Experts will be responsible for system supervision and management (although these are not included in the teams modelled in the thesis, Table 2 gives an indication of the roles of the human team members that are assumed).

Table 1: Agents Types and Mission Roles

Agent Type	Classification	Payload	Role	Information Source	
Simple UAVs: Low-level agent	Micro	>200g - 2kg	Gather and act on unique information (perception task)	Simple Agent: Fire Detector	Sensors, e.g., infra-red OR temperature OR visual camera OR spectrum camera
	Mini	2kg – 20kg		Simple Agent: Weather Detector	Sensors, e.g., wind OR rain OR temperature OR humidity OR snow
					Time (day / night) OR fog OR mist
Picture Compilers: medium capability UAVs	Small	> 20kg - 150kg	Collate, organise, level-based planning, and share information (low-level comprehension)	Simple Agents	

[2] https://www.gov.uk/government/publications/unmanned-aircraft-systems-jdp-0-302

Host	Computer base station	n/a	Compile information and high-level mission planning (high-level comprehension)	Picture Compilers	
Human experts	Suitably qualified and Experienced Practitioners (see Table 2)		Manage mission	Host	

Table 2: Human Experts and Their Goals

Human Expert	Information required	Goal
Navigation Officer	Fire presence, location, terrain type, wind speed, wind direction, and composite material.	Navigation plan
Fire Guards	Fire presence, location, wind speed, wind direction,	Fire response plan
Fire Patrol Officer	Fire presence, location, environmental condition	Understanding of the fire
Evacuation Officers	Fire presence, location, wind speed, wind direction, terrain, and composite fuel.	Rescue plan
Contingency Planners	Fire fighting resources, agents' tasks, and available information	Contingency Plan
Resource Officers	Fire, location	Manage resources.
Asset Officer	Fire presence, fuel type, location, wind speed, wind direction, road type	Manage assets.

The agents' architecture has simple agents (e.g., mini or micro UAVs) responsible for information generation and submitting information to their respective Picture Compilers (PCs) subject to limited communication range. The PCs will then perform an onboard analysis and submit their information to the host for analysis and human presentation. Summarily, the proposed entities settings have the following constraints (Table 3) and assumptions

Table 3: System Constraints

Constraints	**Measurement**
Sparse interactions between simple UAVs, PCs, and hosts, i.e., based on the limited communication range.	This will be measured in terms of number of interactions and sensor range (in metre square).
Limited resources for the agents. That is limited energy (power), memory, and processing power for the UAVs (both simple UAVs and PC, although PCs have higher capacities than the simple UAVs).	Each resource parameter can be measured differently e.g., energy can be measured using the agents' battery consumption in %/s. Detail description of resources parameters and their measurements was discussed in Chapter 3 Table 5.
Limited sensing and communication range.	This can be measured based on the sensor coverage and communication range in kilometres square (km^2) e.g., agents can communicate only when they are within $2m^2$ apart.
Dynamic search area, i.e., changing phenomena states over time, e.g., varying wind speed, wind directions, seasons, e.g., dry, rainy, cloudy, snowy, foggy, etc.	This is characterised by the changing parameters of the environment and how predictable they could be e.g., wind speed, wind direction, fuel types, etc. Changes values over location and time. This is measured by the effectiveness of the prediction of the dynamic parameters.

Each UAV activity consumes appropriate resources, e.g., power, processing ability, memory, etc.	Measured using the resources parameters
Single sensor per simple agent, varying sensors, and absence of sensor for PCs	This is measured by the ability to handle the sensor heterogeneity.

1.2.1 Assumptions that inform the modelling in the thesis

i. Different types of agents with varying abilities, e.g., simple UAVs, PCs, host, and human experts as described in Table 1. Considering the case of forest fire monitoring, information perception is assigned to simple UAVs (Table 1). This is because of their manoeuvrability capacity. Similarly, PCs and host perform small and large information management respectively based on their processing capability. The SMEs play the role of system management and high-level control. Thus, the system comprises various types of agents with specific functionalities across various situations.

ii. Different sources of information (i.e., different sensors for detecting the same and different environmental phenomena) due to the presence of a dynamic environment. Unlike the SMEs sensing mode (e.g., using eyes and brain interpretations), UAVs relied on sensors. The sensor performance varies across various situations of the search area. For example, fire detection using a visual sensor is less reliable in the presence of an object with similar colour (e.g., dried grasses). Thus, the sensors' reliability differences need to be managed. Therefore, this leads to conflicts (contradictions) among agents' information

iii. Agents have different roles: as discussed in (i) and (ii), agents need to have varying roles due to environmental dynamism and limited capacity e.g., a set of dedicated UAVs tasked to monitor fire presence, weather monitors, etc., as describe in Table 2.

iv. Constant agent sensing: agents sensing poll can be periodic, location-based, or constant. I assumed constant sensing because of the high speed of the perception agents (i.e., the simple UAVs as fully discussed in Chapter 6).

v. Possibility for sensor or agents failure: due to agents distribution and hardware/software reliability issues, information can be missing. This is inevitable in any system.

Summarily, the overall challenge of the thesis is to ensure the efficient management of a team of distributed agents operating in a dynamic search under the outlined constraints. For the purposes of this thesis, efficiency is not only defined in terms of optimal resource usage but also in terms of understanding the changing situation that the team faces. This latter is considered through the application of the concept of Situation Awareness, which is borrowed from the domain of Human Factors.

1.3 Introducing the Concept of Situation Awareness

Situation Awareness (SA) means up-to-the-minute cognisance or awareness required to move about, operate equipment, or maintain a system (Sarter and Woods, 1991). For the use case of this thesis (Section 1.2), the agents' environment (search area) phenomena need to be perceived, comprehended, and projected against the plausible future state before any decision or action. Endsley (1995) categorises SA into three (3) stages (i) *Perception* of essential features of the search area; (ii) *Comprehension* of the agent's sensor information; and (iii.) *Projection* into the plausible future state of the search area's situations. Endsley's SA model is popular due to its ability to present the agents' mental model in various situations, i.e., from perception to projection (Endsley, 2015; Jones and Endsley, 1996; Park et al., 2016; Nguyen et al., 2019; Salmon et al., 2015; Endsley, 2015). The arrangement of the SA stages could be arbitrary; however, perception is typically the first task (Endsley, 2015). The perceived situation of the environment leads to the comprehension and projection stages. For instance, pilots plan their trip based on the assumed parameters (e.g., weather reports) and use in-flight information to update their SA (i.e., a cycle of perception, comprehension, and projection followed by decision-making and actions (Endsley, 2015, 1999; Nguyen et al., 2019)). Thus, SA management in a dynamic search area is a continuous process that requires understanding and interpretation of the perceived agents' situations (Endsley, 2015, 2000; Erdelj et al., 2017). As such, SA management within a system is critical, especially in an aspect of complex disaster management e.g., forest fire fighting. Endsley's model paved the way for the development of many other models of SA, such as Distributed Situation Awareness (DSA), team SA, Sensemaking, situated SA, and shared SA (a detailed discussion of these models will be presented in Chapter 2). This thesis models the SA of the agents as Distributed Situation Awareness (DSA) and describes the process in a use case involving forest fire monitoring.

Distributed Situation Awareness (DSA) assumes that SA is distributed within a system of agents, sensor states, and roles (Stanton et al., 2006). From this perspective, SA arises from the transactions that occur between agents. Thus, DSA claims that each agent in the system has a local SA of its situation, and this contributes to and are constrained by a high-level system SA (Salmon et al., 2009, 2008, 2006; Stanton et al., 2006, 2009). From this perspective, there is a system-level awareness of a situation, such that the activity of other agents can compensate for the loss of SA by one agent. This means SA is loosely-coupled within DSA in terms of access to information and communications (Kitchin and Baber, 2017, 2016; Stanton et al., 2006). As such, the DSA system is comprised of a variety of agents with varying roles operating in various situations in which each agent is responsible for its SA management upon which the whole system SA depends (i.e., the collection of individual agents SA is responsible for the entire system SA management). The distributed agents' perceptions (i.e., sensor data collection) in a DSA system are organised through the agent's coordination algorithm (e.g., for the search mission, the coordination algorithm will be responsible for generating a search plan for SA perception).

At its most superficial level, coordination of multiple entities can be characterised as flocking (Reynolds, 1987; Vásárhelyi et al., 2014; Wei and Chen, 2020). In this, an individual entity's Situation Awareness (SA) is detecting one's neighbours and adjusting activity accordingly. Extending this to the DSA system, we might assume that the flock coordinates its activity towards a common goal, e.g., conducting a forest fire search. Here, there is a need to allocate functions between agents, e.g., to monitor fire, weather, etc., in consideration of the agents' distribution constraints (as described in Table 3). As such, members of the flock need to perform specific tasks in different locations, ensure that the fire is monitored constantly, etc. These actions rely on different information. The agent's challenge is to generate efficient search plans that utilise their resources and support SA management. Thus, the flock (rather than a single agent) maintains DSA. This thesis focuses attention on DSA management in consideration of the agents' organisation in the presence of various constraints by taking the aspect of forest fire monitoring by a team of UAVs as the use case.

1.4 Example of Situation Awareness in Forest Fire Monitoring

In this section, the use-case is illustrated by a case study that has been much discussed in the literature on human teams and which illustrates the importance of Situation Awareness.

1.4.1 The Mann Gulch Forest fire incident

The Mann Gulch Forest fire incident of 1949 will be used to describe the complexity of managing Distributed Situation Awareness in forest fire scenes and reemphasise the need for UAVs to address the challenge. In the book '*Young Men and Fire*', Maclean, (1992) describes how a team of firefighters were sent to fight fire at the Mann Gulch in August 1949, in which 13 of them lost their lives. Maclean starts by asking the question, *What should the structure of a small group be when its business is to meet sudden danger and prevent disaster?* Addressing this question requires good SA management (Weick 1995). Thus, this serves as the main motivation behind the thesis's primary research questions, with the main focus on agents coordination and SA management subject to outlined constraints and assumptions. In the Mann Gulch incident, a lookout ranger spotted a fire on a mountain 30 miles away (i.e., SA's perception of the current situation). Subsequently, a team of sixteen (16) firefighters equipped with radio and fire-fighting tools under the leadership of the well-experienced warden, Wagner Dodge, and a second in command, William Hellman, were sent to the scene. The team are new to one another, and the mission requires SA at both individual (each firefighter's level) and global levels (the whole team).

The initial plan (from the base station) was to control the fire before 10 am the following day after they arrived at around 3:00 pm. At around 4 pm, the fire escalates and spreads faster at probably 200 meters per minute. The head, Dodge, yells to his second in command to move the team towards the north of Mann Gulch. Dodge noticed that the fire kept approaching them due to the high wind speed. Later on, Dodge commanded the team to come to his side and lie down on the ground, burnt by an escape fire (an intentional fire set to clear the fuel and give a safe place) set by him. Dodge overheard one of the men saying, "To hell with that, I am getting out of here", while they climbed the Mann Gulch. Two wardens (Robert Salle and Walter Rumsey) made it to the top of Mann Gulch, Dodge survived in the ashes of his escape fire, and the other thirteen (13) died at around 6 pm.

Maclean's interview with the survivors reveals what happened, and concerning the thesis aims and objectives, the following points were marked:

i. The team structure (who does what task?) is significant in managing DSA.
ii. Had UAVs been deployed at the Mann Gulch scene, the worst story we could hear might be: five UAVs were burnt while monitoring the Mann Gulch fire. The reasons for this are:
    a. UAVs could be deployed to monitor the spread rate for the ground wardens, and host agents could generate possible actions quickly (e.g., by learning from previous missions'). But how could this happen? This thesis tackles these issues in Chapters 3-8 through efficient search algorithm development and DSA management tools.
    b. A swarm of 10 micro drones (e.g., Parrot Bebop or DJI Phantom 3) with 30 to 45 minutes of flying time would cost the same as three months' payment for one basic-grade firefighter in the UK[3].

The following discussion is structured using Endsley's 3-stage framework of SA, and this is expanded to consider how and where Distributed Situation Awareness is relevant.

1.4.2 Forest Fire SA Perception Stage

This thesis use-case assumes that SA perception is assigned to simple agents, i.e., micro or mini UAVs using various sensors (as outlined in Table 1). In this case, individual agents acquire sufficient information to support their own view of the situation. The 'view' depends

[3] https://www.prospects.ac.uk/job-profiles/firefighter

on the location of the agent, its sensor capabilities and the goals that it has been set. For each of the simple agents, SA perception is performed under the imposed constraints (Section 1.2 and Table 1), efficient search plan algorithms are developed (Chapter 3). Thus, the agent's search plan is part of SA perception. The outcome addressed resource utilisation and coordination. It should be noted, of course, the SA perception could also be performed by more sophisticated agents, such as human Subject Matter Experts. However, the premise for this thesis is that SA perception is local (both to the agent's physical location and their goals and capabilities). This raises the challenge of how local SA perception becomes Distributed through the team.

1.4.3 Forest Fire SA Comprehension Stage

The local SA perception from simple agents is transmitted to the interpreting agents (for this thesis, the interpreters are: Picture Compilers and host agents, as defined in Table 1). In the SA comprehension stage, information is logically organised by considering the history of the situation and presented to support decisions, e.g., firefighting, evacuation, assets and firefighter coordination, mission plan, etc. (i.e., model SA). Thus, comprehension transforms perceived SA into a meaningful, understandable, and presentable form representing the system's SA model. The thesis assumes a hierarchy through which information moves from simple agents to more complicated ones. Again, it should be noted that these stages could be combined in agents of sufficient capability, e.g., human Subject Matter Experts. However, in a dispersed team, there will always be the challenge of combining comprehension across different agents. In this thesis, Picture Compilers will communicate with simple agents that are relevant to their specific goals (rather than every single simple agent, Picture Compilers only need to acquire information from simple agents that will enhance the situation picture being compiled by that specific Picture Compiler agent).

Different approaches have been applied to address SA comprehension, such as propositional networks, concept maps, ontologies, semantics, fuzzy logic, etc., as in (Burov, 2021; Butakova et al., 2019; Galton and Worboys, 2011; Kokar et al., 2009; Patron et al., 2008; Stanton et al., 2009, 2006; Zhang et al., 2021). These approaches are unable to flexibly measure agents' beliefs (e.g., sensor state based on reliability), support agents' prediction, and handle uncertainties and sensor heterogeneity.

1.4.4 Forest Fire Projection

SA projection in forest fire monitoring refers to estimating plausible future states of the forest to allow effective decisions and actions. This might involve a high-level overview of the 'situation' as a whole. Thus, rather than managing local (i.e., location or goal) SA, projection could involve a view of the interconnected views of a situation in order to make predictions of how the overall situation is likely to develop. This does not preclude projection at local SA levels; e.g., Picture Compilers might need to determine the most suitable regions to search and send coordinates to the simple agents.

Considering the Mann Gulch incident, the projection task involves estimating what will happen next and general decisions/actions (who will do what, where to start fighting, etc.) based on the perceived and comprehended situations. For the forest fire use case, the projection tasks are:

i. Ability to estimate the plausible future situation of the search area phenomena (e.g., fire presence, wind speed, etc.) despite the agent's distribution.

ii. Ability to estimate a missing value (e.g., sensor failure, hardware faults, agents distribution, etc.).

Thus, managing SA in dynamic search requires handling perception, comprehension, and projection. For instance, to address the Mann Gulch incident, Weick (1995) said, "There is a need for the development of a resilient group that is capable of performing improvisation, wisdom, respectful interaction, and communication in a suddenly changed and incomprehensible situation."(Weick, 1993, p2). In this case, improvisation is the ability to quickly restructure an agent's understanding to face the current situation (i.e., SA comprehension). A clear question to Weick is: how could that be possible for automation agents considering the changing nature of the search area phenomena (i.e., changes every short period)? A possible answer to this is to develop an effective SA modelling tool that adapts to the situation and can predict what could happen and suggest a viable plan. For instance, when the Mann Gulch fire-fighters team noticed that the wind speed was high and heard a louder sound of the burning trees, they realised that they were in trouble (comprehension after perception) and expected a sudden danger (projection). To address this challenge, Chapters 3, 4, 5, 6, 7, and 8 develop the agents' effective search plan generation, DSA modelling tool, prediction (forecasting issues early), missing information handling, and adaptable DSA management model.

1.5 Research Questions

The research questions to be investigated in this thesis are grouped under three research topics inspired by the above thesis use case. Each topic has a specific research question and representative sub-questions. The sub-questions are decomposed in later chapters:

RQ1. How can we define a constraint-based search method for agents with limited resources operating in dynamic search areas?

The main focus of this question is the development of a search method that can be applied to a team of distributed agents with minimal resources (as described by the system assumptions and constraints above), i.e., comprising simple and low-cost agents

RQ2. How can we manage the Situation Awareness of distributed agents?

The main focus of this question is to obtain an adaptable, resilient, and predictable SA management tool.

RQ3. How could agents' search support SA management?

These questions raise many research problems; however, this thesis focuses only on the following issues:

i. Coordination of low-level automation agents, e.g., UAVs constrained by limited energy, processing power, communications, etc., to conduct search activity;
ii. Distribution of SA in consideration of agent's contributions towards a system goal (i.e., DSA management);
iii. Relationship between search plan and SA management.

1.6 Objectives of the Thesis

As outlined in Section 1.5, the thesis addresses three overarching research questions. Bearing in mind the constraints and assumptions described in Section 1.2, these research questions are decomposed into objectives as follows:

RQ1. How can we obtain a constraint-based search method for agents with limited resources operating in dynamic search areas?

i. Develop an efficient way of coordinating the automation agents to conduct search activity.

RQ2. How can we manage the Situation Awareness of the distributed agents?

i. Develop effective tools for SA management.

ii. Propose strategies for handling uncertainties, predictions, and conflicts (contradictions in agents' beliefs) in DSA system

iii. Developing an adaptable SA model tool

RQ3. How could agents' search plan support SA management?

i. Depict the relationship between the search plan and SA management

ii. Describe agents' interaction methods for a better SA management

Table 4 summarises the thesis objectives, their importance, and the thesis chapters in which they are addressed.

Table 4: Research Objectives and Addressed Chapters

Research Objectives	Thesis Chapter	Importance	Outcome
Develop an efficient way of coordinating the automation agents to conduct search activity.	3	Efficient searching plan is very important due to UAVs limited resources constraint	Existing solutions have difficulty in team of agents' coordination or resources utilisation. This is based on their reliance on either pseudorandom or structured geometric paths. The proposed algorithm provides a solution that combine the strength of both pseudorandom and structured approaches.

i. Develop effective tools for SA management. ii. Propose strategies for handling uncertainties, predictions, and conflicts in DSA system iii. Developing an adaptable SA model tool	4 , 5	DSA has been criticised of lacking effective SA modelling tool (an information presentation tool that will allow easy perception, comprehension, projection, prediction, decision making, and uncertainty handling).	Presenting SA information using a propositional network, concept maps, ontologies, semantics, fuzzy logic, etc., demonstrates inability to measure agents' belief, agents prediction, uncertainty handling, and agents heterogeneity handling. In this thesis, I proposed the use of the Bayesian Belief Network and describe methods and algorithms to address the outlined challenges in addition to being simple, presentable, scalable, and adaptable.
Ensure agents efficient interactions (information exchange)	5	Agent interaction is essential for SA maintenance. However, unnecessary interactions need to be detected and filtered out.	Existing agents' interactions analysis focussed attention on the agents' consensus process and omits the resource utilisation. This thesis proposes a method that considers the agents' resources utilisation during interactions.
(i) Depict the relationship between search plan and SA managements. (ii) Describe agents interaction methods for a better SA management	 6	Understanding the effect of search methods on the agents DSA will allow us to identify the best search method to use for a better system SA. Similarly, monitoring agents interactions will reduce the system resources consumption by filtering useless interaction	This aspect of the thesis describes all the methods adopted by the thesis and investigate the relation between the agents acquired data and predictions to support the system SA projection. Results show that fixed-search method could result on simple prediction when the search area is structured). The chapter applied Shannon entropy and the formal properties of the BBN to monitor agents' interactions (e.g., differentiate between useful and useless agents interaction).
Propose strategies for handling uncertainties,	7	DSA projection state (estimation of future plausible states of the	The outcome of this thesis describes how DSA systems' phenomena states can be predicted either solely or in combination with other events (i.e., information from

predictions, and conflicts in DSA system		search area phenomena i.e., prediction and uncertainty) is critical in DSA system due to agents distribution.	other entities). To my knowledge, this is a first move towards addressing prediction and uncertainty handling in DSA systems using Bayesian learning approach.
Developing an adaptable SA model tool	8	The version of DSA developed in this thesis is operating in a dynamic search area, which means SA model requires context analysis (i.e., situation assessment) based on environmental conditions. Therefore, there will be a requirement for updating the SA based on perceived information.	The outcome of this objective presents an adaptable, scalable, resource-efficient, multiple-agent-supported (i.e., accommodating both humans and automation agents), and adaptable Bayesian Network structural learning algorithm to show how SA can be updated within the DSA system. The existing solutions lack the outlined features due to their focus on BBN structure development.

1.7 Approaches to Research

Research question number 1 (RQ1) was tackled using a protocol-based algorithm (Chapter 3) derived from Delaunay-triangulation (Boissonnat et al., 2013; Chen and Xu, 2004; Cignoni et al., 1998; Demyen, 2006) of selected waypoints (Chapter 3). Bayesian Belief Network and Bayesian learning are applied to tackle RQ2. I believe that the benefits of using the BBN for SA modelling could not only provide solutions for the limitations of the existing strategies such as the concepts map, propositional networks, fuzzy logic, ontologies, etc., (Galton and Worboys, 2011; Hutchins, 1995; Park et al., 2016; Raymundo et al., 2014; Stanton et al., 2006) but also provide better DSA modelling. Some of the solutions address Endsley's critique of DSA) (Endsley, 2015), such as the need for entities interaction analysis (which is discussed in Chapter 5), individual SA merging to form the system SA (addressed in Chapter 4), lack of framework for information management in a distributed fashion (addressed in Chapter 4 and 6).

The work on Distributed Situation Awareness (DSA) reported in this thesis originated from the field of Humans Factors and Ergonomics. Typically, this has involved the study of human operators working in teams and has used various forms of concept maps to qualitatively describe the information that team members use (Kitchin and Baber, 2017; Salmon et al., 2009, 2008, 2006; Stanton et al., 2006, 2009). However, the focus here is on the study of how a team of agents could collaboratively perform their respective tasks within a system such that SA is managed not only at the individual level but also at the system level (i.e., based on respective agents' tasks). The approach adopted is different from the existing works, which have fallen under Unmanned Aerial Vehicles (UAVs) coordination and Humans Factors and Ergonomics. The research in this thesis addresses the system's technical development.

The evaluation process tests the ability of the proposed methods to perform various tasks. For example, the knowledge presentation tool was evaluated against its ability to measure agents' beliefs, handle uncertainty, prediction, adaptability, scalability, etc., abilities. Similarly, the agent search coordination approach was evaluated based on how the plan utilises the available resources, scalability, adaptability, and predictability. All the evaluation processes are described using the UAVs' mission for forest fire monitoring as the use case under simulation and physical experiments (as discussed in Section 1.2)

1.8 Major Contributions and Novelties

Exploration of the outlined research questions led to significant contributions in the field, such as:

i. Development of a novel algorithm for coordinating agent's search activities. The novelties are the metrics definition, success measuring processes, developed protocols, mathematical modelling, adaptability, scalability, developed propositions, and resource utilisation.

ii. Development of a scalable, adaptable, and agents' belief measurable (belief measurement using probabilities) way of modelling the system DSA. The novelty was the ability to measure agents' beliefs, handle sensor conflicts, description of DSA emergence process, incorporate human contributions, and accommodate agents' differences, adaptability, and learning for prediction and uncertainty handling.

iii. Formalisation of BBN with DSA. The novelty is the mathematical formalisation, develop methods, and algorithms development.

iv. Modelling of DSA as DCOP to show resource utilisation. The novelty lies in how agents' resources could be utilised and defined mathematically.

v. Developing a way of analysing DSA agents' interactions. The novelty of this approach is the ability to classify useful and useless agents' interactions.

vi. Development of various approaches (both single and multiple information approaches) for making predictions within the DSA system. The novelty is that agent-based suitability and efficacy were investigated, in addition to new metrics definition.

vii. The development of an efficient way of handling uncertainties in the DSA system was investigated. The novelty of this approach is that an effective way of handling uncertainties in single or multiple forms of information was analysed.

Additionally, the scalability of the best approach was tested, and the behaviours of the algorithm were mathematically formalised with the system's requirement (as outlined in Section 1.2).

viii. Development of a way of estimating the number of clusters for the classifications algorithms, e.g., Gaussian Mixture Model(GMM), K-means, etc. The novelty of this approach is the significant reduction in computational demand and scalability.

ix. Formalising DCOP with learning and DSA. The novelty is modelling the agents' resource utilisation in the DSA system.

x. Descriptions of how agents' simulated operations can be transformed into physical agents' operation. The novelty comes from the description of the more realistic dynamic system modelling.

xi. Development of a DSA-Based BBN structural learning algorithm. The novelty of this approach is its adaptability, scalability, and efficiency (resources management).

2 Chapter 2 Literature Review

In this chapter, Distributed Situation Awareness is reviewed. Also, the aspects of DSA concerning the agents and their coordination are reviewed in detail. This is in line with elaborating the position of the thesis with respect to agents' coordination, DSA management, and the selected use case (forest fire monitoring) literature. Thus, this leads to the point-by-point justification of the thesis direction.

2.1 Distributed Situation Awareness Concepts

In its original conception, SA described the knowledge used by pilots to understand the state of their aircraft. Different information can be interpreted to understand the flight situation, e.g., speed (Hutchins, 2001). Endsley's model of SA became one of the most popular models and is widely used to describe SA within a system. To Endsley (Endsley, 1995), SA comprises the *current situation* (perception), interpretation of the current situation using the history of previous situations (comprehension stage), and the *projection* of plausible states. These three stages lead to the system's decision-making and actions. One of the critical limitations of Endsley's model is that, Situation Awareness (SA) was assumed to reside in individual agents or team (Endsley, 2015, 1999, 1995; Endsley and Jones, 1996; Jones and Endsley, 1996). However, there is a version of SA with the notion of SA distribution among agents upon which system SA is derived. That is, SA is distributed across agents based on their roles, capacities, and situation, which gave birth to the field of Distributed Situation Awareness (Stanton et al., 2006).

Distributed Situation Awareness is not a 'component' view of SA but a 'systems' view in which SA arises from the interactions between agents (Salmon et al., 2015). Thus, as discussed in Chapter 1, Distributed Situation Awareness (DSA) involves the ability of diverse agents (an agent can be human or automation, e.g., UAVs) to perceive, comprehend, and act on information towards achieving a common goal, such that the process of goal achievements follows from an individual level (phenotype schema) to system-level (genotype schema). This is inspired by the concept of DSA developed in (Stanton et al., 2006), and differs from other theories of SA such as team

SA, shared SA, and situated SA (Chiappe et al., 2012; Danczyk et al., 2016; Endsley and Jones, 1996;). Thus, DSA agents have different roles to play within the system in a distributed fashion (i.e., with varying locations, roles, goals, and views), in which the complete system SA is realised at both agents' local and system level in a compatible and transactive manner (Stanton et al., 2009).

Compatibility in DSA means striving towards achieving the system goal by contributing to the assigned role. According to (Cacace et al., 2014; Cox and Zhang, 2005; Ferguson and Allen, 2007; Gómez and Green, 2017; Makonin et al., 2016; Stanton et al., 2009; Tecuci et al., 2007), compatibility requires perception, understanding, goal-based assimilation (role and goal-based interpretation and allocation), and situation assessment. In contrast, a transactive manner means the successful exchange of information among agents with varying goals, views, and activities (Chiappe et al., 2012; Salmon et al., 2009, 2006; Salmon and Plant, 2022; Stanton et al., 2006, 2017; Stewart et al., 2008). Because agents have different goals, locations, and views in the DSA system, the SA modelling technique needs to allow the restructuring of the information to suit agents' respective goals. For instance, considering the forest fire monitoring use case introduced in Chapter 1, an agent responsible for forest fire mapping (fire spreading control agent) considers wind speed as a more important information than an agent accountable for assets monitoring (determining the available firefighting resources, e.g., vehicles, drones, etc.). Thus, exchanging information across agents requires different situation assessment (Chiappe et al., 2012; Stanton et al., 2006, 2009). This has to acknowledge the current situation and other team members' roles (Baek and Lim, 2018; Berger et al., 2021; Lu et al., 2016).

To draw together the agents' transactions, one can create a communications network in which all agents share and update each others' views or assume that the collation of information occurs in a structured manner. The former can be very costly due to possible hardware/software failure, communication breakdown, information misinterpretation, possible mistakes, and agents' goal variations, i.e., what is needed by agent A could be very different from the requirement of agent B (Chiappe et al., 2012; Stanton et al., 2009; Yang et al., 2022; Zadeh et al., 2021). Additionally, communication demands could be costly due to excessive demand on resources (e.g., memory, processing ability, bandwidth, etc.) during the process of information transfer from one agent to another and security risks

(Chiappe et al., 2012; Cummings and Mitchell, 2008; Danczyk et al., 2016; Lu et al., 2016; Salmon et al., 2009, 2006; Stanton et al., 2006, 2009; Stewart et al., 2008). The latter approach (a structured manner), as adopted by this thesis, impose specific protocols at both individual agent's level and system level in such a way that mission activity supports SA management across agents' level. In this approach, one might assume that reporting agents would not be continuous, that not all agents will need to know nor be able to process the views of other agents, and that there is a need to 'fill in the gaps' to create a detailed situation picture. The thesis approaches the challenge by developing system protocols (i.e., to effectively maintain the agents' structure) to support system management due to limited resources for the automation agents; efficient agents search mission planning, i.e., UAVs. The thesis approach is novel in the following ways: (i) development of system protocols to support SA, (ii) resource utilisation, and (iii) focus on automation agents (although there will be SMEs guidance). The system protocols aim to consider the agent's distribution constraints in consideration of resources limitation.

In terms of automation, agent SA maintenance becomes complicated if it involves agents with different roles and abilities (Chiappe et al., 2012; Endsley, 2015, 1999; Endsley and Jones, 1996; Kitchin and Baber, 2016b; Lopes et al., 2014; Matthews and Beal, 2002; Nguyen et al., 2019; Pearson et al., 2016; Salmon et al., 2015, 2008, 2006; Stanton et al., 2006; Stanton, 2016). This is as a result of the following reasons:

1. Lack of sufficient cognitive ability to analyse complex situations by automation agents: automation agents rely on the implemented algorithms, which may not be as effective as human solutions.
2. Limited resources: resources are minimal, especially when UAVs are applied. The resources were outlined in Chapters 1 and 3 (i.e., energy, memory, computational capacity, etc.).
3. Coordination of automation agents is challenging: coordination of multiple UAVs is challenging in terms of communication, control, autonomy, and decision-making.

The thesis chose this direction based on the outlined limitations.

2.2 Models of Situation Awareness

There exist different models of SA, such as comprehension-based SA in a dynamic environment (Durso and Sethumadhavan, 2008), system topology SA (Consciousness et al., 1995), and Endsley's model (Endsley, 1995). Endsley's model became popular in literature because of its ability to present the agents' mental model across various situations (Endsley, 2015, 1995; Endsley and Jones, 1996; Jones and Endsley, 1996; Teichmann and Motus, 2021; Walshe et al., 2021). The initial theory of SA by Endsley assumes an individualistic view of SA by each agent and advances further to an unclear team SA, i.e., SA for a team of agents (Endsley, 2000, 1999, 1995, 1988; Endsley and Jones, 1996). Since then, new models have evolved, such as the situated SA (Chiappe et al., 2014), shared SA (Chiappe et al., 2012), sensemaking (a perception and comprehension based SA), and Distributed SA (Stanton et al., 2006).

Shared SA focuses on having several agents with the same SA (Chiappe et al., 2012). This model remains a topic of argument because sharing information does not mean sharing SA, especially when SA is viewed as a product of a process that analyses the received information. As such, unanimous situation understanding could be impossible (Chiappe et al., 2012; Salmon et al., 2008; Stanton et al., 2001). This leads to an argument that perfect shared SA can never be realised in a multiagent system involving diverse agents with varying goals (Stanton et al., 2009), although simple, loosely coupled, and identical modes of information sharing could reduce the challenge. The main reason is that investigation reveals that astronauts using the same language to communicate have a better SA management performance due to their better understanding (Chiappe et al., 2012; Stanton et al., 2009). This will be difficult to achieve in a highly dynamic (e.g., random) situation.

In contrast to shared SA, team SA described how a group of agents could maintain SA towards achieving their common goal within the system SA. In team SA, agents assume to have identical SA towards achieving their goal, although with possible overlap (perhaps with other team members). A clear difference between team SA and DSA is that, DSA comprises distinct agents operating on a common goal with the assignment of individualistic roles and SA management (Stanton et al., 2006). DSA differs from shared SA by making agents to have distinct goals, situations, roles, and views. Thus, shared SA is tightly coupled, whereas DSA is loosely coupled, i.e., DSA assumes loosely coupling at the system level. As such, each agent could have its version of SA compatible with the system SA. At times the concepts overlap. As an arbitrary example of overlapping between team SA and DSA, consider a team of UAVs responsible for forest fire monitoring from the thesis use case (Chapter 1 Section 1.4); the team using similar sensors, say spectrum cameras, could have a team SA. However, from a DSA point of view, each agent will be acting according to its location, current information, and situation, i.e., SA is distributed (perhaps with other agents using different sensors to achieve the same aim). This thesis defines DSA based on the distinct nature of individual agents and collaborative efforts of those agents towards common goal achievement (perhaps with different views). Additionally, the idea is to keep the compatible and transactive nature of achieving the DSA process as simple and effective as possible and suitable for the human-automation team.

2.3 Endsley's Stages of SA: Perception, Comprehension, and Projection

Out of the outlined models of SA, the thesis focuses on DSA involving the human-automation team. Priority is given to automation agents. The human factor aspects were derived from existing work (Burov, 2021; Danczyk et al., 2016; Lee et al., 2007; Salmon et al., 2009; Salmon and Plant, 2022; Stanton et al., 2006, 2009; Stanton et al., 2001; Stefanidi et al., 2022). This remains the current most developing challenge in the field of DSA (Baber et al., 2011; Baek and Lim, 2018; Berger et al., 2021; Bouvry et al., 2016; Cummings and Mitchell, 2008; Heintzman et al., 2021; Kanistras et al., 2013; Lu et al., 2016; Nguyen et al., 2019; Park et al., 2016; Quintin et al., 2017; Salmon and Plant, 2022; Stanton et al., 2017; Zadeh et al., 2021).

This section will discuss Endsley's three-stage model of SA (i.e., perception, comprehension, projection) in line with the current work and thesis aim and objectives.

As a primary layer of SA, perception is the agent's sensed information (i.e., perceived using sensors or derived from the perceived sensors' values) leading to its current situation understanding (Minsky, 1987). This thesis assigned the task of perception to the simple agents, e.g., mini or micro UAVs (as classified in Chapter 1 Section 1.2) mounted with dedicated sensors (on the notion of single-agent per sensor). A clear question is whether the perceived current situation belief (derived from the agent's sensor status) could be Boolean (i.e., 0 or 1) or flexible? If flexible, how could it be measured? This thesis argues the need for a flexible belief measurement, especially when operating in a dynamic search area due to situations variation (remember, a dynamic environment is one in which its phenomena states change over time (Baek and Lim, 2018; Kitchin and Baber, 2017; Norstein et al., 2019; Stanton et al., 2006). Flexible belief (verified information) will be a better approach. For example, a fire-detecting UAV with a camera sensor could have lower reliability during the day due to possible confusion from fire-like objects. An additional essential feature on top of the aforementioned is providing a framework to allow belief learning and handle a missing variable based on the agent spatiotemporal distribution (this will be discussed further in Chapter 7). Existing methods use agents sensor probability, Boolean approach, and situation-based values modelling (Galton and Worboys, 2011; A. Khan et al., 2014; Khan et al., 2015; Lähdesmäki et al., 2006; Lohia et al., 2019; Nebel et al., 2019; Schloss et al., 2014; Schwab et al., 2020; Uma Pavan Kumar Kethavarapu and S. Saraswathi, 2016). This thesis develops a team of agents' based presentation of perceived information using BBN.

Comprehension is the understanding of the search area situation based on the logical organisation and interpretation of the perceived information (Endsley, 2015, 1995; Endsley and Jones, 1996). The logical organisation and interpretation of information require appropriate information assembling based on the current situation and level of the agents (i.e., it needs a practical situation assessment). Thus, the comprehension state presents the SA model. Existing works utilise the use of concepts maps, propositional networks, fuzzy logic and ontologies (Galton and Worboys, 2011; Norstein et al., 2019; Salmon et al., 2006; Stanton et al., 2006; Stewart et al., 2008; Zhang et al., 2021) to model the system SA qualitatively. These approaches are limited by the lack of agents' belief measurement, support for SA projection (prediction and uncertainty handling), multi-state presentation, and adaptability. To address these challenges, this thesis applied a Bayesian Belief Network. Results show an ability to handle uncertainties (in the form of missing variables due to hardware/software issues or soft findings from a faulty sensor using probabilities estimation algorithms), prediction support, nice interface, belief measurement, and adaptability, i.e., flexible situation assessment and reconfiguration (Bari, 2011; Bouckaert, 1995;

Meloni et al., 2009; Park et al., 2016; Pavlin et al., 2010; Scanagatta et al., 2019, 2019; Williamson, 2001; Zhang et al., 2020) as described in Chapters 4,5,7, and 8.

SA projection involves the ability to predict plausible future states of the system under limited time/space and information availability. For example, considering the forest fire spread monitoring use case (Chapter 1), projections of the situation estimate where a fire will likely be in t future time, etc. The effectiveness of the projection states can be measured based on their accuracy and mission support. For example, considering the Mann Gulch incident described in Chapter 1, the projection of Dodge was more effective because of the following reasons.

1. He clearly understood the situation escalation based on the perceived information, e.g., the sound of burning trees, strong wind speed, etc.
2. He effectively applied his previous experience (i.e., through the learning process) to understand the current situation.
3. He predicted the plausible future states (i.e., the fire would be out of control).

Thus, we can see that every agent can project, but the question is, does the projection supports the mission goal? The use of BBN and learning as proposed by this thesis demonstrates effective previous information management, prediction ability, autonomous knowledge presentation (data-driven knowledge presentation), and uncertainty handling as proved by the thesis chapters. The thesis evaluates prediction and uncertainty handling methods such as the time series models (AR, MA, ARMA, ARIMA, and SARIMA models), Gaussian Process and, the expectation-maximisation algorithm (Adhikari and Agrawal, 2013; Bottou, 2010; Dama and Sinoquet, 2021; Dempster et al., 1977; Ganoni and Mukundan, 2017; Hendikawati et al., 2020; Karduni et al., 2021; Mandt and Hoffman, 2017; Papastefanopoulos et al., 2020; Romanycia, 2019; Tandon et al., 2020).

In line with the thesis problem (Chapter 1 Section 1.2), perception, comprehension, and projection could differ based on the agents' type and capacity. For example, perception, comprehension, and projection of the simple agents (from Chapter 1) refers to the ability to perceive the environment (using sensors), understand it (by taking the values, e.g., X^{o}C for temperature sensor), and act accordingly (as projection, e.g., reporting to the nearest PC after rough mapping process) based on the limited view. The case differs in terms of Picture Compilers (PC) and host. PC performs lower-level information organisation and integration (perhaps using the proposed Bayesian Network) to make predictions and handle uncertainties. Perception comes from various individual agents' information, and comprehension is realised by logically organising those information). The host can comprehend larger information (due to abundant resources as described in Chapter 1) including learning and projection tasks. Human Subject Matter Experts (SMEs) comprehension is the ability to understand the presented information, amend it where necessary, and implement the system planning and control.

The perception, comprehension, and projection cycle is incomplete without the decision and action complementary events, which some authors sometimes assume as part of the projection (Endsley, 1995). The perception, comprehension, and projection process requires appropriate decisions and actions. For example, going back to the Mann Gulch fire scenario of Chapter 1, the perception begins with the lookouts' fire detection. Comprehension is followed by merging the information (fire, location, time, weather reports, etc.), and projection estimates the mission time, critical fire level etc. The decision state is an immediate part of projections, e.g., whom to send to the scene, what to use, etc., and action (i.e., execution of decisions, e.g., sending firefighters, firefighting starting location, etc.).

The critiques in (Endsley, 2015) on DSA by Endsley are mostly nothing but a highlight of how the recent works improved the earlier concepts. Despite the harsh feedback by Endsley to other models, e.g., situated SA and team SA, her comments on DSA were good. However, her criticism on the lack of information exchange architecture, agents' effective information presentation technique, and system information management (Endsley, 2015) was correct and is one of this thesis's main aims (Chapter 1). For example, Chapters 4, 5, 6, 7, and 8 tackle the issues of effective SA model (Chapter 4), conflict and interactions management (Chapter 5), SA realisation methods (Chapter 6), autonomous knowledge (data-driven knowledge presentation) presentation (Chapter 8), and uncertainty handling (Chapter 7); while Chapter 3 describes an SA-supportive agents search mission coordination.

2.4 Agents Search Coordination to Support DSA

The system perception task is assigned to simple agents (Chapter 1) under imposed constraints, e.g., limited resources, communication range, etc., as outlined in Chapter 1. Thus, the thesis proposes a structured, efficient, scalable, and predictable search plan generation method to support the system DSA management.

2.4.1 Search Plan Generation

Existing work has focused on fixed patterns (fixed geometric patterns, e.g., parallel track, creeping line, etc.) methods, enabling each agent to compute and adapt its paths (Bevacqua et al., 2015; Jensen-Nau et al., 2021; Kappel et al., 2020). Geometric fixed-pattern approaches follow predefined geometric paths (Cabreira et al., 2018), e.g., expanding square shapes, parallel sweeps, and other patterns, to explore the search area (Bevacqua et al., 2015; Jensen-Nau et al., 2021). A related method, such as sector search, defines angles and edges to control the agents’ paths (Bevacqua et al., 2015; Jensen-Nau et al., 2021). These approaches make it easier to compute paths for each agent but do not support adaptation to changes (utilisation of the method to conduct different tasks, e.g., searching, mapping, etc.) in complex, dynamic domains and struggle to optimise the agents' resources (Cabreira et al., 2019, 2018; Di Franco and Buttazzo, 2016).

There is also work inspired by animal foraging with random waypoint generation (i.e., drawing from suitable distributions) within the search area (Chawla and Duhan, 2018; Sutantyo et al., 2011). Lévy flight and Brownian motion became the most popular pseudo-random method in which waypoints are generated based on certain distributions seeded by random numbers and proved to be the most effective. The critical advantage of pseudorandom methods is the agent’s independent planning, which supports decentralised coordination. At the same time, these methods often suffer from poor agent coordination in complex domains, difficulty in predicting the future activities of the agents, and little consideration of the agent's sensing abilities (Nurzaman et al., 2009), which in general affects the system DSA management process.

Grid-based methods segment the search area into cells and impose some structure on the problem by constraining the random walks (Hackney and Clayton, 2015). The paths followed by each agent and the number of times an agent visits a particular cell are controlled probabilistically, e.g., using computational models inspired by the ant pheromone (Cabreira et al., 2019; Di Franco and Buttazzo, 2016; Koenig and Liu, 2001; Nasirian et al., 2021; Yang et al., 2014). The limitations of such methods include the computational demands of searching for an optimal solution and the difficulty in exchanging information collected by the individual or sub-groups of agents (Demyen, 2006; Koenig and Liu, 2001).

There exist hybrid methods that combine the strengths of fixed-pattern and pseudo-random approaches. The hybrid methods apply protocols to guide the plan generation and maintain flexible and good agents coordination (Bolander et al., 2018; Hasegawa et al., 2012; Kallmann, 2005; Nebel et al., 2019; Ozkan and Kilic, 2022; Quintin et al., 2017; So and Ye, 2005). For example, Voronoi tessellation generates random waypoints and visits the centres of the circumcircle of the Delaunay triangles of the random waypoints. Similarly, the approach of (Bolander et al., 2018; Nebel et al., 2019; Ozkan and Kilic, 2022) proposes using local protocols in the form of an if-then fashion to control agents' search activity with initial random or fixed-pattern-based exploration. Other forms of hybrid approaches are the pseudorandom methods augmented with an artificial potential field, bat algorithm, Fireflight algorithm, cuckoo birds inspired algorithm, ant colony optimisation, etc., (Chawla and Duhan, 2018; Sutantyo et al., 2011). This thesis contributes to the hybrid strategies by aiming to develop a search plan generation algorithm that supports agents' SA management and utilises their resources using global system protocols. This is similar to the work of (Quintin et al., 2017; Vagale et al., 2021), i.e., the idea of path generation to support SA. However, the thesis extends the focus to SA management and resource utilisation. This is achieved by developing a predictable system protocol (Chapter 3).

2.4.2 Agents Interaction

Agent coordination and control strategies follow either centralised, decentralised or partially decentralised methods. A centralised approach utilises an omniscient server responsible for decision making, communication, coordination control, and tasks control. Each agent will be waiting for the central server for any of these actions. A key advantage of the centralised approach is that optimal solutions can be achieved effectively (Cortés and Egerstedt, 2017; Desai et al., 1998; Gage and Murphy, 2004; Vasile and Zuiani, 2011). Critical challenges to centralised method are (Cortés and Egerstedt, 2017; Desai et al., 1998; Turpin et al., 2014; Vasile and Zuiani, 2011):

(i) Communication has to be maintained thoroughly. This is impossible due to the possibility of hardware/software failure, interferences, etc.

(ii) Workload management. The server is undergoing a vast control workload such as coordination, decision-making, etc. This places a significant burden on resource demands such as memory, processing power, communication bandwidth etc. Thus, failure of the server means complete system failure (Desai et al., 1998; Turpin et al., 2014; Vasile and Zuiani, 2011).

In the decentralised method, agents are tasked to act individually. That is, each agent is responsible for coordinating its activity within the team (Bouvry et al., 2016; Gage and Murphy, 2004; Lumelsky and Harinarayan, 1997; Nguyen et al., 2014; Stranders et al., 2009; Turpin et al., 2014; Vásárhelyi et al., 2014; Yan et al., 2011). There exists partially decentralised or sparse interaction or a hybrid method (Bolander et al., 2018; Khan et al., 2015; Kho, 2009; Nebel et al., 2019). In the hybrid method, part of the activities is centralised subject to agents' proximity (i.e., exchange of information when agents are very close). Thus, hybrid strategies could select part of the activities as centralised and others to be decentralised. For instance, initial planning could be decentralised, and subsequent decision-making can be centralised.

In contrast, the decentralised approach offers autonomy because agents act independently based on perceived or shared environment states to achieve their mission. This solves issues bedevilling the use of the central server. However, resource utilisation and optimal coordination cannot be guaranteed due to the partial distributions of the agents. This thesis assumes a hybrid approach by selecting the best possible combination. For instance, simple agents will be reporting their information to the respective picture compilers or hosts based on proximity interaction (e.g., within 2 meters) and engaged with any superior command. A similar interaction approach exists between the picture compilers and the host. Thus, during the interaction between the Picture Compiler (PC) agent and the simple agent, coordination tasks and decision-making will be controlled locally in a decentralised fashion. Commands can be changed during an interaction, e.g., if the PC receives information about fire presence from other agents, then that information will be used in altering the simple agents' activities (i.e., centralised control). Thus, the thesis focuses more on hybrid coordination.

2.5 Distributed Situation Awareness in Forest Fire Monitoring

The author is interested in forest fire monitoring due to its complexity and dynamism, which allows application in most practical domains (Weick, 1995); as such, this use case will be repeated many times in this thesis. The initial task of forest fire monitoring is to look for the fire, which means early detection results in an early control (Fire Lookout History of the Santa Fe National Forest, 2017). Thus, lookout agents (i.e., UAVs in this case) need to spread themselves effectively (e.g., avoiding redundant search, focusing on sensitive locations, etc.) to effectively cover the forest and report information on fire presence. UAVs became the leading methods because they are cheap and manoeuvrable. Unfortunately, coordination, limited resources (i.e., insufficient battery capacity, little computational power, etc.), and limited ability to perceive situations and make decisions are the main issues bedevilling UAVs application which are to be addressed by this thesis.

The UAVs will be mounted with fire detection sensors, e.g., infrared, spectrum camera, temperature sensor, etc. (as described in Chapter 1). The belief of the UAVs (derived from their sensor states) and other agents is measured using probability to quantify the perceived information's level of certainty and uncertainty. For instance, fire detection using a visual camera will not be as reliable as temperature sensors during the daytime because fire-like objects, e.g., dried grasses, could interfere with visual detection. Similarly, during human lookouts in the olden days, couples' lookouts are more reliable than a single person's lookout (*Fire Lookout History of the Santa Fe*

National Forest, 2017), i.e., four eyes are better than two. Thus, this thesis focuses on an effective search coordination algorithm for the UAVs with the main aim of resource utilisation (the key factor of successful forest fire monitoring) and DSA management.

The second challenge is the issue of understanding the fires' situation (comprehension) based on the distributed agents' architecture. This includes information presentation (i.e., about perception), understanding (e.g., where it occurs, when it occurs, where will it spread to, etc., which will be achieved through logical organisation of the information) and projection (e.g., decisions were to start fighting, what assets to be deployed, mission time, etc.). All these should align with the changing nature of the situation, e.g., based on the dynamic search situation. This thesis proposes using BBN to address that challenge, as discussed in Chapter 4.

The projection challenge refers to predicting future states and uncertainty handling. For example, in the aspects of fire spreading forecast, wind speed forecast, etc., as mentioned, the projection state is tightly coupled with the decision-making and actions. This thesis approaches prediction and uncertainty handling (i.e., SA projection) using learning algorithms.

In conclusion, the thesis system DSA is maintained by a number of varying agents. For instance, the simple agents (i.e., the mini or micro UAVs) are tasked to perform the perception task (i.e., by gathering information from their sensors). Thus, the simple agents will coordinate their search activity to ensure the search area's coverage and resource utilisation. This is also in consideration of their varying location, roles (e.g., fire detectors, weather monitors, etc.), and changing nature of the environment (i.e., due to dynamic environmental parameters, e.g., wind speed, wind direction, etc.). The perceived information (e.g., information on fire occurrence within the search) will then be comprehended (perhaps by depicting the contextual logical relations with other information) at the PCs and host level through interactions (i.e., information exchange). The comprehension tool needs to adapt to the perceived information. The projection tasks are characterised by making predictions and uncertainty handling. The prediction task forecast the search area's situation, e.g., where the fire will likely move in t future time, etc., based on the perceived information (e.g., fuel type, wind speed, wind direction, etc.). This allows decision-making and actions by both human and automation agents, e.g., where to place the firefighters, what assets to be deployed, etc.

3 Chapter 3 Efficient Constraint-based Search

In Chapters 1 and 2, I have argued for the need for an efficient, predictable, scalable, and adaptable method to generate agent search plans based on the imposed constraints. Therefore, in this chapter, a solution to this challenge is reported. The proposed approach builds on a Delaunay triangulation of the search area to generate a search plan. The method was implemented on simulation software using a multi-UAV mission for forest fire monitoring as the use case and compared its performance against fixed-pattern and pseudo-random baselines. Results proved a better method. Again, the proposed solution demonstrates easy implementation on real UAVs.

3.1 Introduction

Different constraints can characterise a search path. As described in the thesis use case (Chapter 1), resources such as sensor range, battery capacity, agents interactions, computational power, memory use, and communication range were considered to be limited and thus conform to the challenges of applying UAVs to search problems (Bailon-Ruiz et al., 2022; Bolander et al., 2018; Cabreira et al., 2018; Cortés and Egerstedt, 2017; Jensen-Nau et al., 2021; Kanistras et al., 2013; Merino et al., 2006; Mohd Daud et al., 2022; Ozkan and Kilic, 2022; Revach et al., 2017; Ucgun et al., 2021). Additionally, agents do not know where the targets (e.g., fires for the thesis use case) are located. The location of targets may change due to dynamic search area phenomena, e.g., forest fires in Figure 2 move faster downwind proportional to the wind speed. Thus, the search area is dynamic in addition to the imposed constraints.

The focus of the search plan generation is for the simple agents (mini or micro UAVs as described in Section 1.2 of Chapter 1) with the requirements of allowing easy sensor information collection (between simple UAVs and PCs or host), resource efficiency, scalability, adaptability, and predictability. Each simple agent is responsible for generating its search plan, whether alone or collaborating with other agents at the premission planning stage. PCs are responsible for assigning initial location and search plan generation protocols to control simple agents' waypoints plan generation. Thus, the challenge is developing a search method that allows efficient agent search plan generation (i.e., utilising the agent’s resources parameters in Table 5) and easy coordination despite the imposed constraints and assupions of (Chapter 1).

Existing agents' search methods assign fixed search patterns (geometric paths) to each agent within the team. These agents can operate in a centralised or decentralised manner (as discussed in Chapter 2). The fixed pattern follows specific predefined geometric shapes, e.g., sector search of Figure 7 (sector shape with defined angles and radius), parallel track of Figure 3 (follows horizontal sweeps defined by a track size), etc. The main limitations of these methods are the issue of scalability, adaptability, and resource utilisation (Cabreira et al., 2019, 2018; Jensen-Nau et al., 2021). Alternative to those methods are pseudo-random methods (search plan waypoints generation is based on a random distribution), e.g., the Lévy flight(Chawla and Duhan, 2018; Nurzaman et al., 2009; Sutantyo et al., 2011). These approaches were developed to improve the adaptability and scalability of the search solutions. Unfortunately, pseudorandom methods make it challenging to manage agent coordination due to pseudorandom behaviours. To address these issues, hybrid strategies (combined versions of the fixed-path or pseudorandom method with other techniques were developed (Chawla and Duhan, 2018, 2015; Nurzaman et al., 2009; Ozkan and Kilic, 2022; Yang and Suash Deb, 2009; Yang, 2012)). For example, the work of (Ozkan and Kilic, 2022) and (Sutantyo et al., 2011; Yang, 2010) shows the hybrid versions of fixed-path and pseudorandom strategies, respectively. In these methods, the waypoints path interval is controlled by the sensor information, i.e., a longer path with no targets and shorter ones when the target is close (i.e., target attraction and repulsion protocols). Although the hybrid methods addressed some of the original fixed-pattern or pseudorandom methods' limitations, they still have limited predictability and coordination issues. For example, the attraction and repulsion protocol of (Ozkan and Kilic, 2022; Sutantyo et al., 2011; Yang, 2010) controls the individual agents only, and there is no way to coordinate the whole system agents. Therefore, in this Chapter, we contribute to the hybrid methods as follows:

i. Applied a Delaunay triangulation of the search area to allocate simple agents to particular regions while optimising the outlined resource parameters (e.g., battery power, memory, computational resources, etc., as described in Table 5), despite the imposed constraints. This applied system control protocol development using the Delaunay-triangulations theorems and defined custom protocols rather than the individual agents' protocols.
ii. Generates efficient, adaptable, predictable, and scalable search plans;
iii. Demonstrate easy implementation on real UAVs.

Specifically, the work reported in this chapter focuses on RQ1 of Chapter 1(i.e., *How can we define a constraint-based search method for agents with limited resources operating in dynamic search areas?)*.

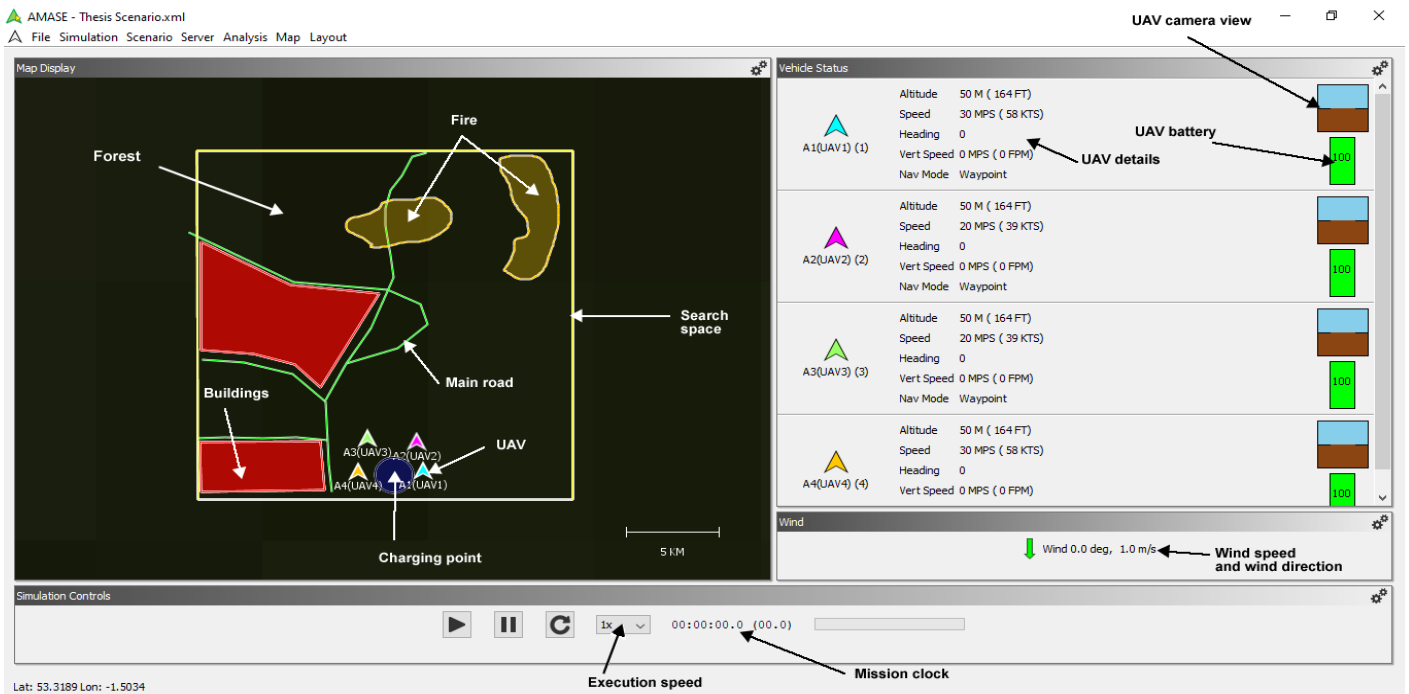


Figure 2: Problem Modelling on AMASE

Figure 2 illustrates the search problem simulation using the Aerospace Multi-agent Simulation Environment (AMASE)[4]. The UAV's mission is to explore the search area (defined by the rectangular area) and report its sensor information while optimising the resources (outlined in Table 5). For this Chapter, the mission activity is to search for the fires.

3.2 Problem Formulation and Model

The central problem of this chapter is to develop a method for search plan generation that efficiently utilise the agents' resources (e.g., energy, memory, computational power, etc., as specified in Table 5) and is scalable, adaptable, and predictable to allow real-world application despite the imposed constraints. I modelled the problem as a finite horizon, proactive, dynamic, and multi-objective distributed constraint optimisation problem, PMO-DCOP. This is described by the tuple D of Equation 1.

$$D = \{A_{i,j}, P, T, W, \lambda, \Upsilon_i, \delta, S_{condition}, K_i, \alpha_i, C, O, S\}$$

Equation 1: Problem Modelling using DCOP

here,

$A_{ij} = \{a_{11}, a_{22}, a_{33}, \ldots, a_{ij}\}$ is the set of agents, i, of type, j i.e, $i \epsilon [1,M], j \epsilon [1,N]$. For instance, fire spread detecting agent of simple agent type. Thus, the agents are heterogeneous in terms of roles and types. For example, the agents can be a team of mini UAVs serving as simple agents and tasked to detect the search area phenomena (fires, wind speed, fuel types, etc.) using variety of sensors and a group of micro UAVs tasked to collect and understand information of the simple agents.

---

[4] AMASE is a simulation-based framework developed by the Aerospace Vehicle Technology Assessment & Simulation Branch of AFRL. AMASE can display mission planning with simulated objects, waypoints path, communication channels, etc. AMASE is available at (*https://github.com/afrl-rq/OpenAMASE*, 2019)

P is the set of agents' (P_A) and mission (P_m) parameters to be optimised with their cost optimisation function C_o parameters (Table 5). These parameters determine the success of the agents' mission. For example, considering the forest fire use case (Chapter 1), exploring the search area with minimum battery (energy), memory use, computational power, number of agents interactions, mission time, redundant search, and higher coverage is most preferable, e.g., covering a forest of $100km^2$ in 5 minutes is better than covering $20km^2$ in the same time.

Table 5: Definition of Parameters

Parameter	Optimisation	Parameter Type
Energy(battery)	Minimisation	Agent
Memory	Minimisation	Agent
Computational power	Minimisation	Agent
Number of agents interactions	Minimisation	Agent
Coverage	Maximisation	Mission
Path divergence	Maximisation	Mission
Redundant search	Minimisation during search	Mission
Mission time	Minimisation	Mission

The justification for selecting these parameters is based on the limited resources of UAVs (Cabreira et al., 2019; Kanistras et al., 2013), as discussed above. In addition, most of the outlined cost functions parameters (Table 5) are widely used in literature (Huang, 2001; Jensen-Nau et al., 2021; Kanistras et al., 2013; Koenig and Liu, 2001; Li et al., 2011; Sauter et al., 2005; Sutantyo et al., 2011; Yoon and Kim, 2013).

T is the set of finite mission time, $T_i = \{t_1, t_2, t_3, \ldots, t_n\}$, n = 1, 2, 3, … n (e.g., t_1 = 5 minutes, t_2 = 10 minutes, etc.). T defines the solutions finite horizon feature (Fioretto et al., 2018; Hoang, 2019; Hoang et al., 2017) and can be measured using the mission clock.

W is the finite set of search plan waypoints to be explored, i.e., $W=\{w_1,w_2,w_3,\ldots,w_n\}$. Thus, $W \in S_i$, where S_i is the search space.

λ is the agents' waypoints assignment function based on the agent's situation, such that $\lambda: W \times \Upsilon_i \times \alpha_i \rightarrow A_i$.

Υ_i is the agent's situation defined by its location and sensor value over a period of time T, i.e. $\Upsilon_i = \{ \Upsilon_1, \Upsilon_2, \Upsilon_3, \ldots, \Upsilon_n\}$. The agent's situation is defined jointly by its current location and belief over time. i.e., $\Upsilon_i \rightarrow \delta_i \times s_i \times T_i$.

δ is the agent's belief (based on the sensor state and location, e.g., information about fire presence in location X). Thus, the probability distribution of δ consists of situation action transition values, i.e., $\delta \rightarrow \alpha_i \times \Upsilon_i$. The value of δ can be initialised using δ = 100%/n, where n is the number of sensor states, e.g., fire presence or absence. The update (increments/decrements) of δ occurs after every agent's sensor poll.

$S_{condition}$ is the search area conditions described by the tuple $S_{condition} = \{ S_v, \Upsilon_i, T_i\}$ where S_v is the set of search area's dynamic parameters (i.e., wind speed, wind direction, fuel type, fuel condition, and terrain nature). That is, the search area is dynamic, and the dynamism depends on the changing parameters S_v. It is assumed that the search is obstacle-free because the agents are aerial. However, the proposed solution should be adaptable enough to incorporate obstacles.

K_i is the constraints $K_i =\{k_1,k_2,k_3,\ldots,k_n\}$ imposed on the agent, i.e., limited resources parameters of Table 5.

α_i is the set of action spaces across agents' situations Υ_i, i.e., $\alpha_i = \{\alpha_1 \times \alpha_2 \times \alpha_3 \times \ldots \times \alpha_i\}$ is factored across each agent at every situation Υ_i. For example, if a simple UAV spots a fire, the action could be to make a different search plan to understand the fire's spread etc.

$O = \{o_1, o_2, o_3, ..., o_n\}$ is the set of targets. The target's movement is subjected to the search area conditions $S_{condition}$ parameters. This can be modelled using a dynamic function that takes the changing parameters e.g., search area condition, fuel type, etc. An example of this functions has been described in Equation 13

C is a real-valued cost function defined by $C: \lambda_i \rightarrow \mathbb{R}^+$. Every agent's waypoint assignment is measured using an assigned real value e.g., two waypoints separated by a distance of 2KM have higher coverage than the ones with a distance of 1KM. Similar real values will be applied for other parameters. The real values are assumed to be assigned by a Subject Matter Expert (SME).

S is the search area defined by the cells set $S = \{s_1, s_2, s_3, ..., s_n\}$, e.g., a bound forest with n number of cell segmentations. That is, each cell s_i uniquely identifies a portion of the search area.

Thus, all of the outlined variables of Equation 1 fit the described forest fire use case of Chapter 1. For example, the varying agents (A_{ij}) are tasked to generate the search area's phenomena belief, δ, (verifiable perceived sensor information) based on the agents' situations (Υ_i), actions(α_i), waypoints plan (W), changing nature of the search area ($S_{condition}$), and evolving nature of the targets (O) under the imposed constraints K_i and over the time t and mission costs C measured using the agents and mission parameters P. The cost utilisation function performs its actions through an effective waypoints assignment function (λ) based on the agent's belief. For instance, considering the Mann Gulch incident discussed in Chapter 1, the lookout agents (which are UAVs in the thesis use case) undergo various situations, Υ_i (e.g., fire presence, wind speed change, etc.) and actions, α_i, (e.g., interactions with PCs, changing search paths to map fire, etc.) while exploring their assigned waypoints (W) to detect the dynamic targets (O_t), i.e., fires. The operating search area is changing based on its dynamic parameters, $S_{condition}$, e.g., wind speed, wind direction, fuel type, etc. (as can be seen by the rapid escalation of the Mann Gulch fire discussed in Chapter 1), which requires various actions given varying situations. All the agents' actions need to utilise the cost values C measured using the agents and mission parameters of Table 5 (e.g., minimising energy, mission time, etc.).

Therefore, the DCOP challenge is to find the set of search plans π' such that the mission utility function U^{π} is efficient, i.e., $\pi' \in \mathrm{argmin/max}_{\pi \in \Pi^i} U^{\pi}$ where $\pi' = (\pi_1, \pi_2, \pi_3, \ldots, \pi_i)$ is the set of search plan that utilise agents' and mission parameters. U is the best parameter cost-utility function given agents' waypoints assignment, situation, and action defined by Equation 2. For example, during the search mission, redundant search needs to be avoided (based on the specifications of Table 5). The function becomes the maxima/minima function based on the passed parameter's target optimisation, i.e., minimisation or maximisation. In other words, the DCOP problem for the agents (e.g., UAVs) tasked to search the area (forest) is to generate search plans that utilise their resources, e.g., by avoiding redundant search, maximising coverage, etc.

$$U(C, P, \lambda) = argmin/max_{\ C}[\textstyle\sum_{t=0}^{T} \sum_{ci}(\vec{C}_i(\lambda_i \backslash P_i))]^T$$

Equation 2: Problem Utility Function

Thus, Equation 2 measures the waypoints' mission utility based on the waypoints' assignment, agents' situations, and parameters, e.g., highly separated waypoints need to be assigned when exploring the search area and is measured based on the separation distance.

3.3 Unique Features of the Existing Solutions

The specific features of each method need to be considered during the experiment development. For instance, the fixed pattern methods vary based on their path structures in terms of coverage, redundant search, scalability, and adaptability. The creeping line (Figure 3) and parallel track (Figure 4) methods have similar path structures, and redundant search can be controlled by balancing the sensing range between tracks (inter-sweeps interval, note that the path structure can be traced using the directional arrows). This is different from other methods, such as expanding square (Figure 6), Zamboni search (Figure 5), and sector search (Figure 7). For example, the angle difference of a sector search can be used to control its scalability. Again, this is different from the Zamboni search (Figure 5) and the expanding square search (Figure 6), of which redundant search and path control require edge adjustment only.

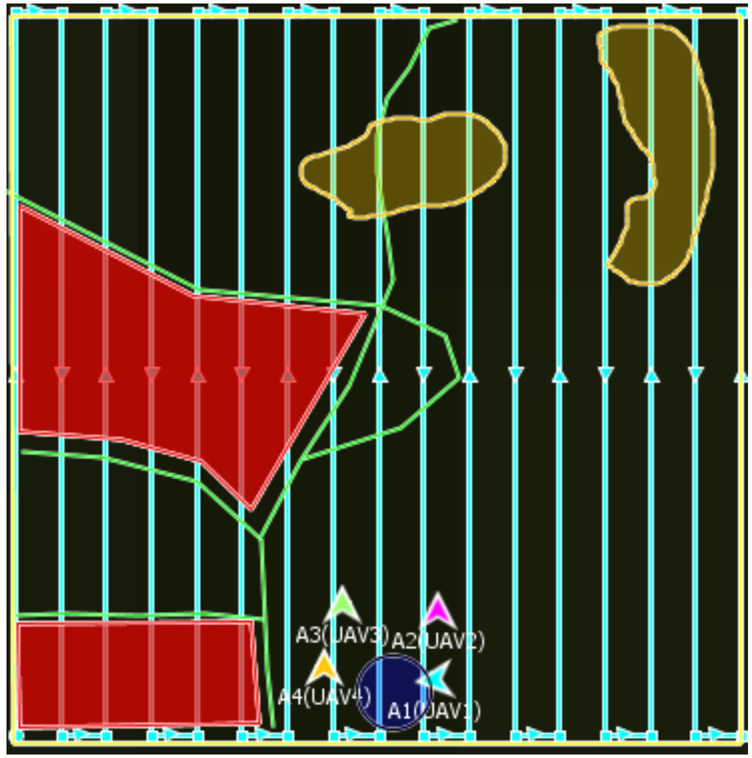


Figure 3:AMASE Implementation of Creeping Line Search

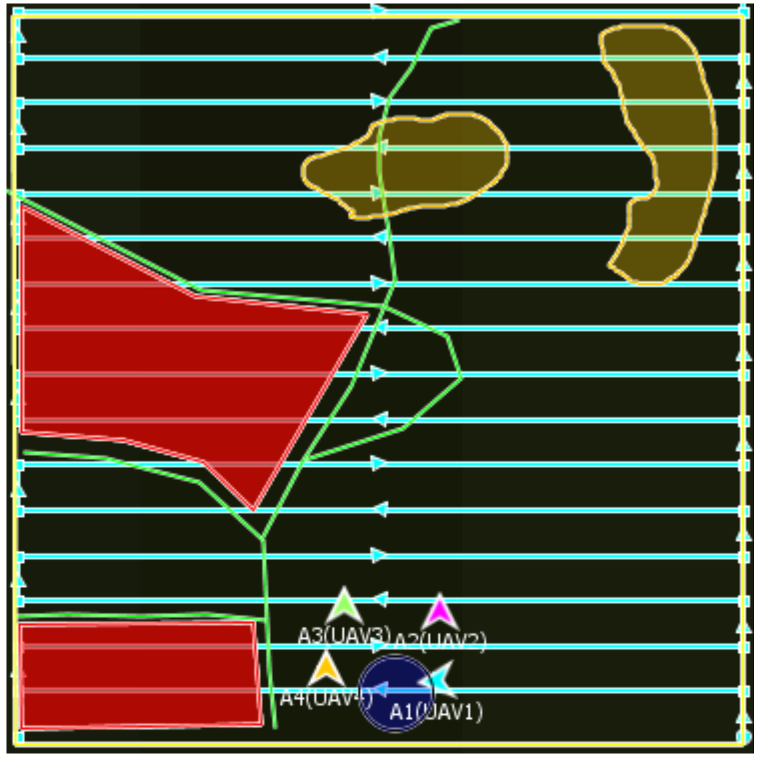


Figure 4:AMASE Implementation of Parallel Track Search

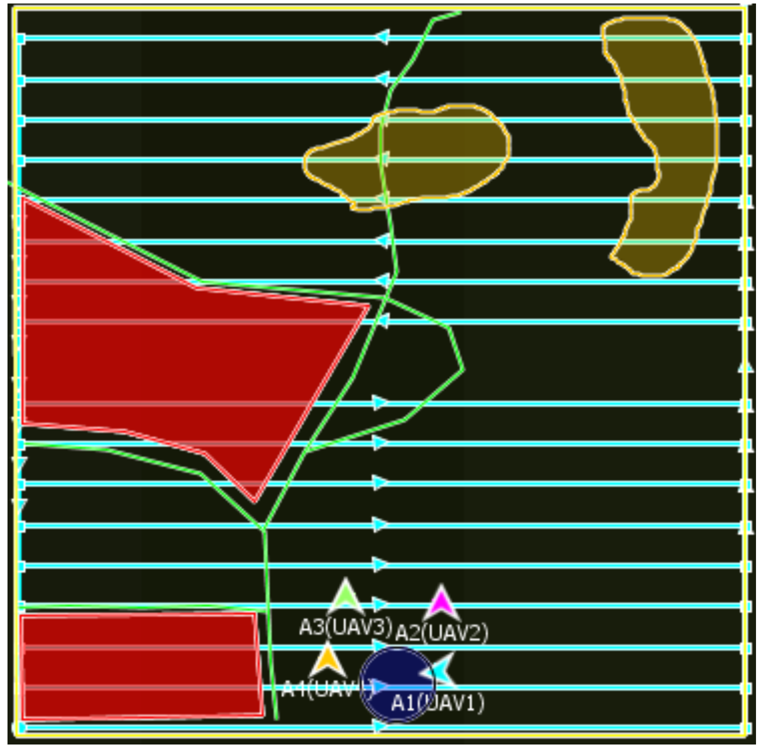


Figure 5:AMASE Implementation of Zamboni Search

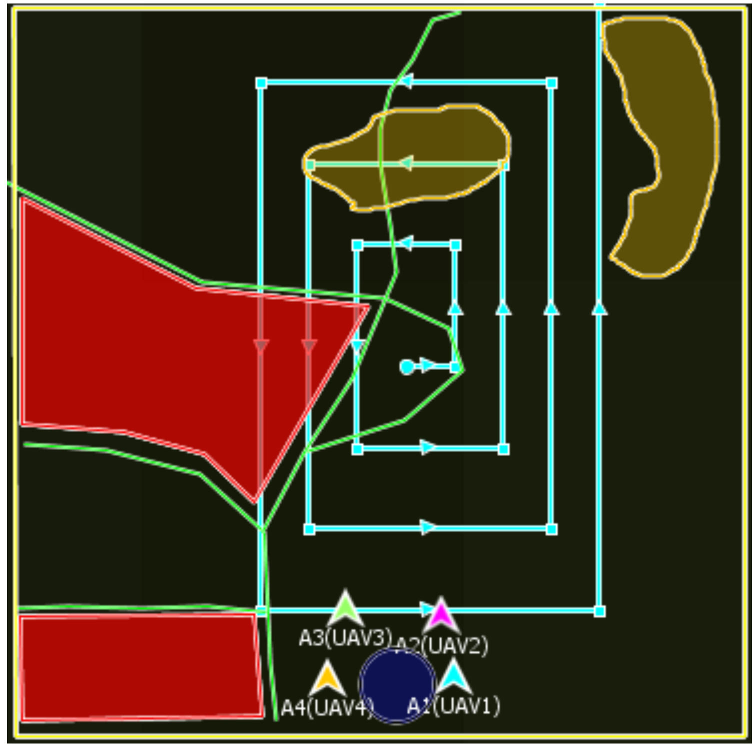


Figure 6:AMASE Implementation of Expanding Square Search

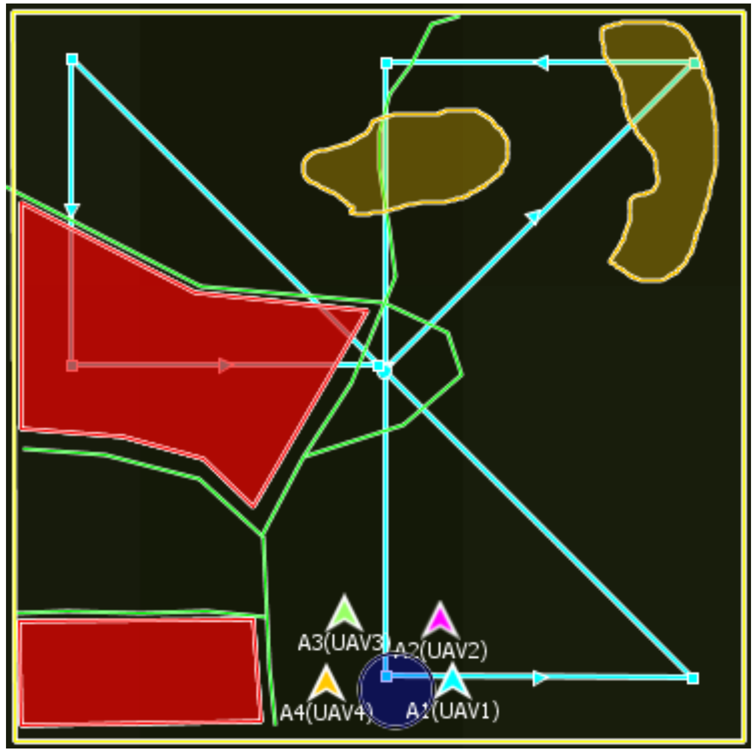


Figure 7:AMASE Implementation of Sector Search

The pseudo-random strategies' structure depends on the pseudo-random number generation distribution. Waypoints are generated randomly using any suitable random number generator. This chapter used the implementation of Lévy flight in (Chawla and Duhan, 2018) as the candidate for the pseudorandom methods. The linear congruent approach for random number generation (Knuth, 1997) was adopted as the random number source due to its popularity.

3.4 The Proposed Solution

The proposed solution evolved by taking the Delaunay-triangulation of systematically selected seed waypoints (seed waypoints can be selected using defined protocols, e.g., the longest non-cross and opposing path from the current or chosen location, or sample from a predictable and non-redundant distribution). Each centre of the Delaunay triangle is taken as a waypoint. This version of the algorithm (as described in Figure 8 for the problem in Figure 2) is the Delaunay-centric algorithm.

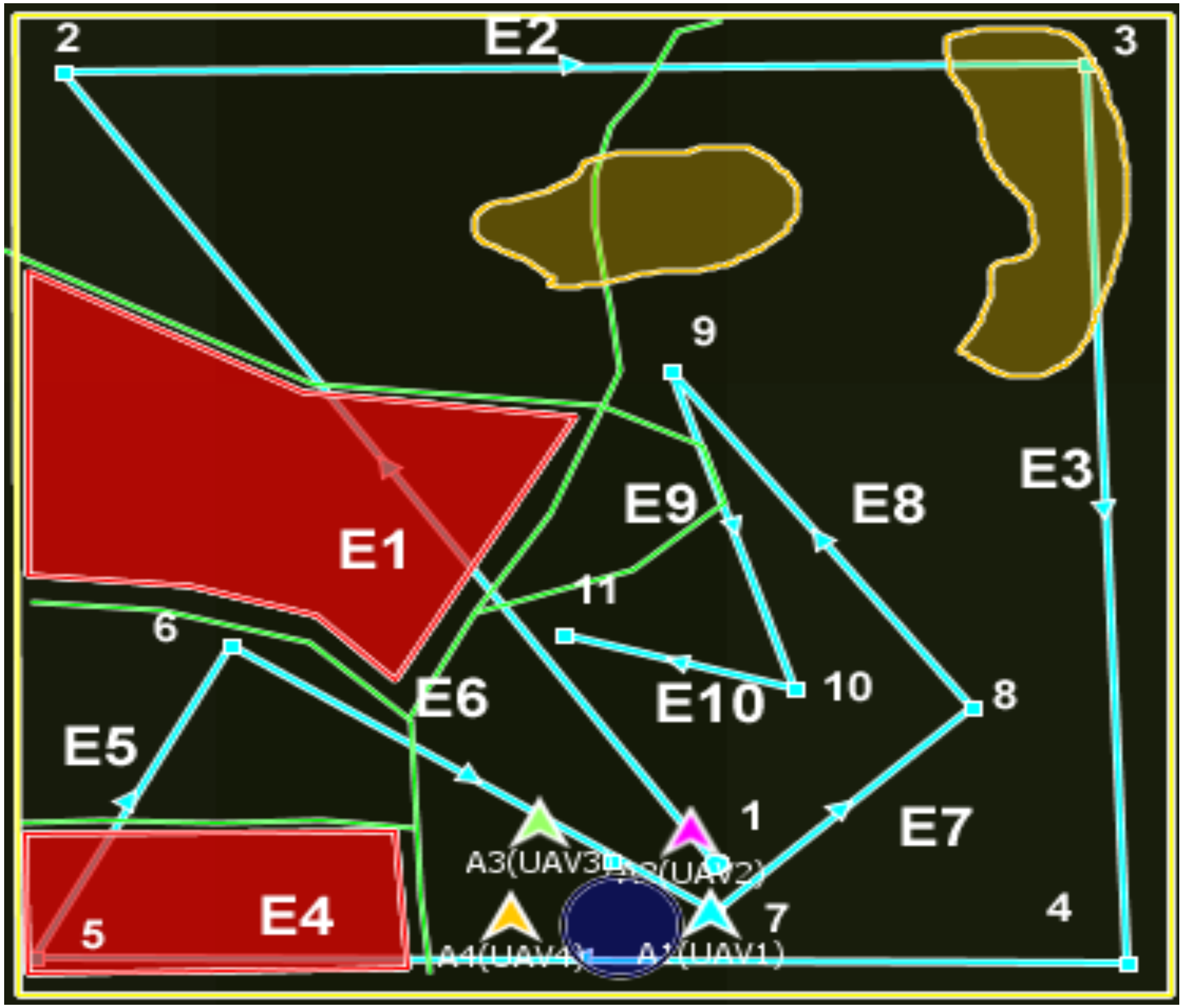


Figure 8:Implementation of the Initial Algorithm on AMASE

From Figure 8, the waypoints labelled 1, 2, 3, 4, and 5 are the longest non-crossed (non-overlapping path detected through visual observation or line intersections) paths from the agent's (UAV1) current location and serve as the first-layer waypoints. The second layer waypoints are labelled 6, 7, 8 and 9 and are the centres of the Delaunay triangles of the first layer (see Definition 3.1). Similarly, waypoints 10 and 11 are the centre of the second layer Delaunay triangles, the last layer's waypoints (waypoints number =2). Note that a similar method is a Voronoi diagram in which centre of the circumcircles is visited (Koenig and Liu, 2001; So and Ye, 2005; Yu et al., 2014). The plan in Figure 8 and the Voronoi diagram are predictable because agents' future locations can be predicted if the initial waypoint, speed, and the waypoints generation protocols (i.e., Delaunay-triangulation theorems), etc., are known. Transformation of this version of the algorithm to a more efficient, adaptable, scalable, and predictable performance was described in Section 3.4.1 as a Delaunay-Inspired Multi-agent Search Strategy (DIMASS) solution.

Definition 3.1. Layer (τ_i) refers to the set of waypoints at the same level (hierarchy) of the plan, $\tau_i : W_x \rightarrow A_{ij}$, such that, $W_x = \{w_1, w_2, w_3, \ldots, w_m\}, \forall\ W_x \in W,$ and $\exists\ \tau_j = W_y \rightarrow A_{ij},$ where $W_y = \{w_1, w_2, w_3, \ldots, w_n\}$, and $W_x \cap W_y = \{\} \forall\ W_x, W_y \in W$. Waypoints in every layer are characterised by having similar edges configuration, quadrants, and angles. For example, from Figure 8, the first layer waypoints are waypoints 1 to 5 and the second layer waypoints are 6-9, and finally, the third layer waypoints are waypoints 10 and 11. Each layer's waypoints are characterised by having similar edge length, quadrant patterns, and angles; e.g., from Figure 8, first layer waypoints have the highest edges, quadrants, and angles. This is different from other (layer 2 and layer 3) layers' values.

3.4.1 Delaunay-Inspired Multi-agent Search Strategy (DIMASS) Solution

The rule for generating seed waypoints is similar to the Delaunay-centric method (i.e., the proposed solution). However, Algorithm 1 (DIMASS) uses angles, quadrants, and edges length to control the generation of subsequent waypoints. For example, waypoints 6, 7, 8, and 9 in Figure 9 were obtained by projecting towards angle $\Theta = 180^o/n,$ where n is the number of upper-layer waypoints computed using the Delaunay-triangulation theorem (below). This provides a system protocol for better predictability. The quadrant projection sequences depend on the number of agents and unique paths needed. For instance, UAV 1 could use the quadrant sequences first, third, second and fourth, while other UAVs could have the third, fourth, second and first quadrants, etc. The ability to control the edge (e_i) length for $i = 1,2,3...,n$, projection angles, and quadrants allows the generation of a predictable solution. For example, the second layer edges (edges for waypoints 6,7,8 and 9) of Figure 9 were selected as half of the opposing longer edges, i.e., the path (edge) 5 to 6 is half of edge 4 to 5 *(i.e.,* $e_5=e_4/2$*),* etc. Projection angles, quadrants, and edge length depends on how the waypoints utilise the agents' and mission's cost values (Table 5). The number of waypoints at every layer is computed using the Delaunay-triangulation number of waypoints theorem. The number of triangles and edges of a Delaunay triangulation process is $2n-2-k$ and $3n-3-k$, respectively, where n is the total number of waypoints and k is the number of convex waypoints (Perera and Barnes, 2011). Therefore, the solution to the DCOP problem in Equation 1 is based on quadrants, angles, and edge length values adjustment to utilise the resources in Table 5. Finding the best combination of curves, quadrants, and edges for the agents (i.e., the solution to Equation 1) is computationally cheap and simple because the highest number of quadrants is only four.

Similarly, edges and angles can be controlled by discretising the values into ranges. The best plan produces the best utilisation of the agents' and mission parameters considering the imposed constraints. Algorithm 1 describes the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) solution, and Figure 9 shows the solution for the problem in Figure 2.

In terms of agents collision avoidance in DIMASS, waypoints altitudes variation (i.e., varying the colliding agents' altitudes e.g., 30m and 20m, etc.), waiting technique (stopping agents to wait for the passing of other co-agents), and waypoints adjustment (e.g., quadrant, angles, or edges variation) can be applied. Note that, collision among agents can be detected by considering the agents' generated plan and operating speed or collision detection sensors.

Algorithm 1: Delaunay-Inspired Multi-agent Search Strategy (DIMASS) Algorithm

1	Input: seed waypoint (W), U, control protocols (derived from the Delaunay-triangulation theorems)
2	Output: Search plan waypoints plan(π)
3	Initialise W (i.e., the first layer waypoints), for instance, the longest non-crossed jumps of
4	Figure 8 etc. i.e.,: $W_i \rightarrow \tau_i$, for i=1. For all $a_{ij} \in A$ do Find $\pi_i \in \Pi$, such that $\pi_i \in \text{argmin/max}_{\pi \in \Pi^i} U^{\pi}$ using
5	While (count($\tau_i \leq 2$) do Use the Delaunay triangulation theorem to generate the number of waypoints for each layer and repeat the process until the number of waypoints is less than or equal to 2.
6	Find the best angle, quadrants, and edge lengths allocation based on the control protocols, imposed constraints, parameters cost trade-offs and current situation. For all w_i

	$\in W$ allocate $U_{best} : w_i \times \lambda \rightarrow A_i, w_i, \forall w_i \in W$. // by adjusting waypoints angles, quadrants, and edges size (note that, to be done by picture compiler)
7	$w_i \rightarrow \tau_i$ {Add waypoint(s) to layer}
	Endwhile
8	$\tau_i \rightarrow \pi_i$ {Add the layers to search plan}
9	Return π_i

The agents' coordination and scalability can be controlled by generating protocols to control the waypoints generation; as such, Definitions 3.2 and 3.3 could help in monitoring the agents' coordination.

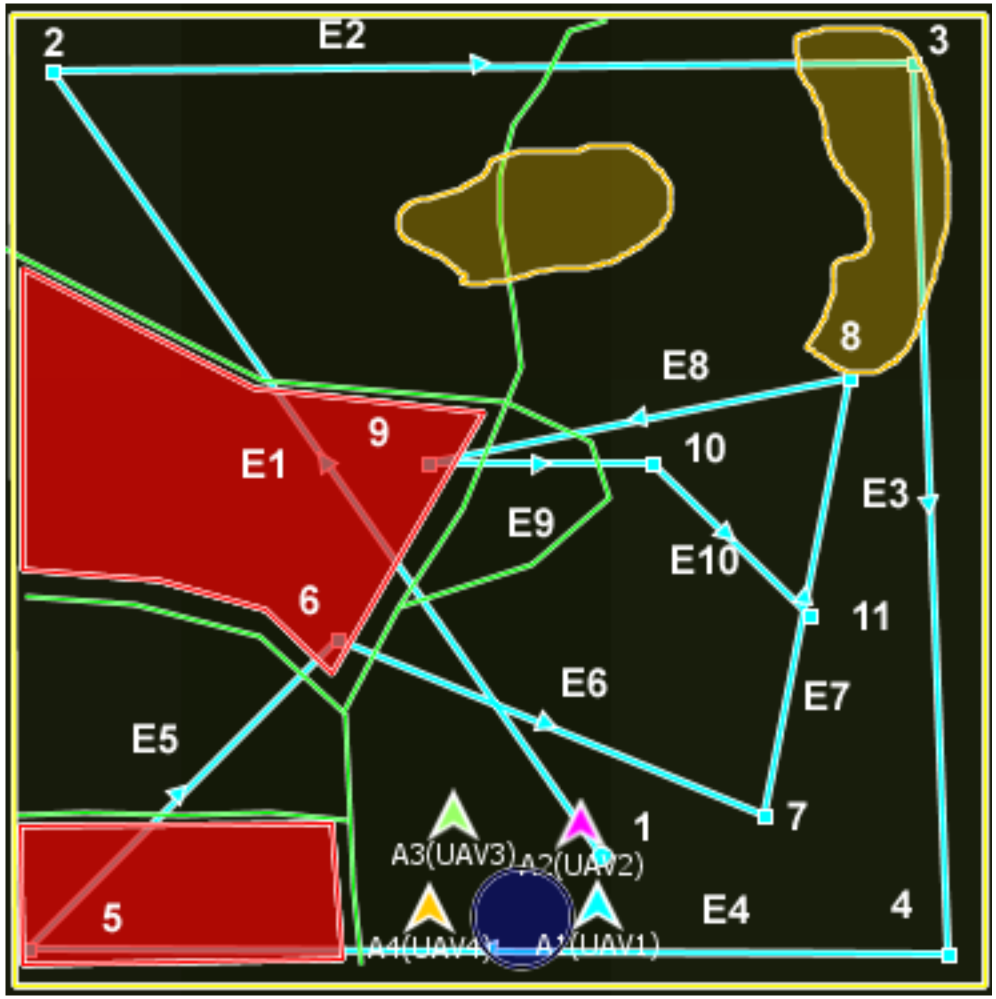


Figure 9: Example of Delaunay-Inspired Multi-agent Search Strategy (DIMASS) Solution

Thus, the application of Algorithm 1 follows these simple steps:

1. Seed waypoints selection protocols generation, e.g., the longest non-crossed paths

2. Control protocols applications i.e., based on the Delaunay-triangulation and layering
3. Quadrants, angles, and edge length generations control protocols applications

In terms of the AMASE simulation, this can simply be generated by passing the coordinates of the seed's waypoints to the plan generation Java method in the form of array. Each entry of the array contains the longitude and latitude entry of the seed waypoints. Subsequent layers' waypoints are then generated based on the control protocols using a dedicated Java method that take in as a parameter the upper layer values and control protocols. The edge length can be computed using the Haversine formula (for longitudes and latitudes) or Euclidean distance (for planar coordinates). For example, the protocols will then impose that $E_{i+1} = E_i/2$, i.e., edges from lower layer are half of the upper layer's ones, e.g., E5 = E4/2 from Figure 9. The angles and quadrants will be derived in a similar way as described above.

Definition 3.2: Reflection Two waypoints, X_{ij} and Y_{ij}, in two-dimensional space, i, j, with upper and lower search area boundaries, M_{ij}, N_{ij} where i, j ∈ $\mathbb{R}^d$, within a plan π, are said to be reflected if and only if the distance computation in Equation 3 exists.

$$Y_i = N_i-(X_i - M_i) \text{ or } Y_j = N_j - (X_j - N_j) \text{ where } i,j \in \mathbb{R}^d$$

Equation 3: Waypoints Reflection

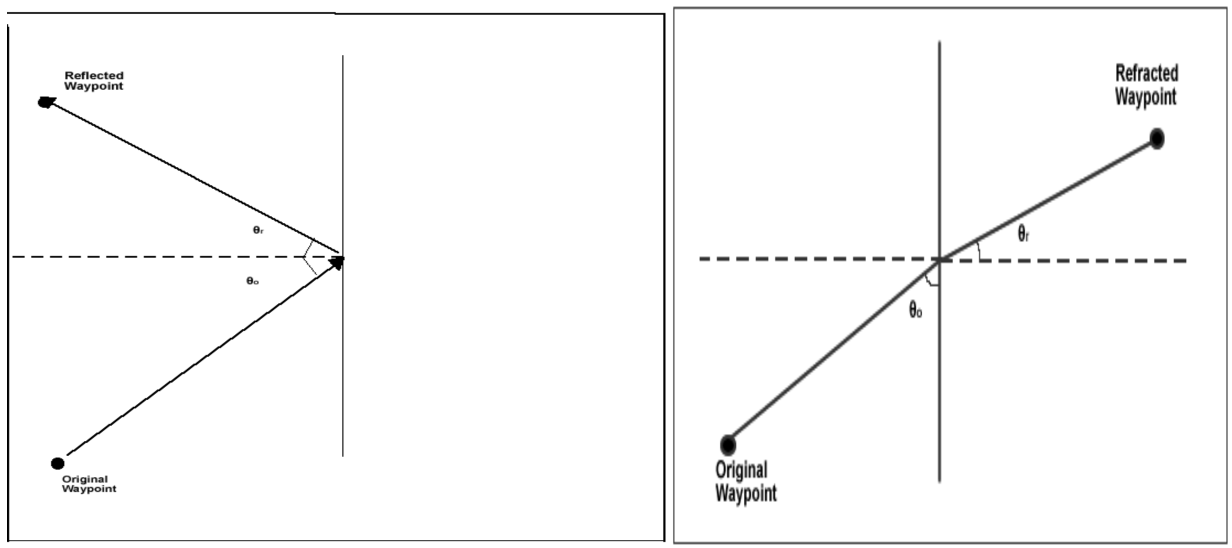
where i,j ϵd, d is the dimension of the search area. In other words, a reflected waypoint maintains the same distance as the original waypoint from the direct opposing side of the search space as described in Figure 10a.

Definition 3.3: Refraction Two waypoints X_{ij} and Y_{ij} in a search area with boundaries, M_{ij}, N_{ij}, where i,j ∈ $\mathbb{R}^d$ are said to be refracted waypoints if and only if the distance computation in Equation 4 exists.

$$Y_{ij} = N_{ij}-(X_{ij}-M_{ij}) \text{ where } i,j \in \mathbb{R}^d$$

Equation 4:Waypoints Refraction

In other words, a refracted waypoint maintains the same position as the original waypoint but from the opposing angle as described in Figure 10b.



(a)Reflected Waypoints (b) Refracted Waypoint

Figure 10: Example of Waypoints Reflection and Refraction

The concept of waypoints reflection and refraction can be used to control the algorithm's scalability (by reducing the number of interactions). For instance, Figure 11 shows the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) algorithm applied to four UAVs (from the problem in Figure 2) using the concept of waypoints reflection and refraction. UAVs labelled A1 and A4 have refracted initial waypoints s1 and s4, and agents A2 and A3 have reflected initial waypoints s2 and s3. Each agent has a unique path (shown by colour and directional arrows) because they use different initial waypoints and control protocols. Note that, this is one of the possible solutions; the best solution can be obtained by adjusting the edge length, angles, and quadrants of waypoints to conform to the parameters cost utilisation described in Table 5. One of the advantages of the concepts of waypoints reflection and refraction is that a larger number of agents can be controlled with fewer interactions. For example, considering the plan in Figure 11, it is obvious that the reflected or refracted version of the seed waypoints guarantees non-redundant waypoints across layers as far as the edge is greater than the diagonal of the sensor range. Interestingly, this implicitly coordinates the visits, e.g., the overlapping edges of Figure 11 will be explored by different agents at different time. Overall, this reduces the number of agent interactions and improves agents' coordination scalability.

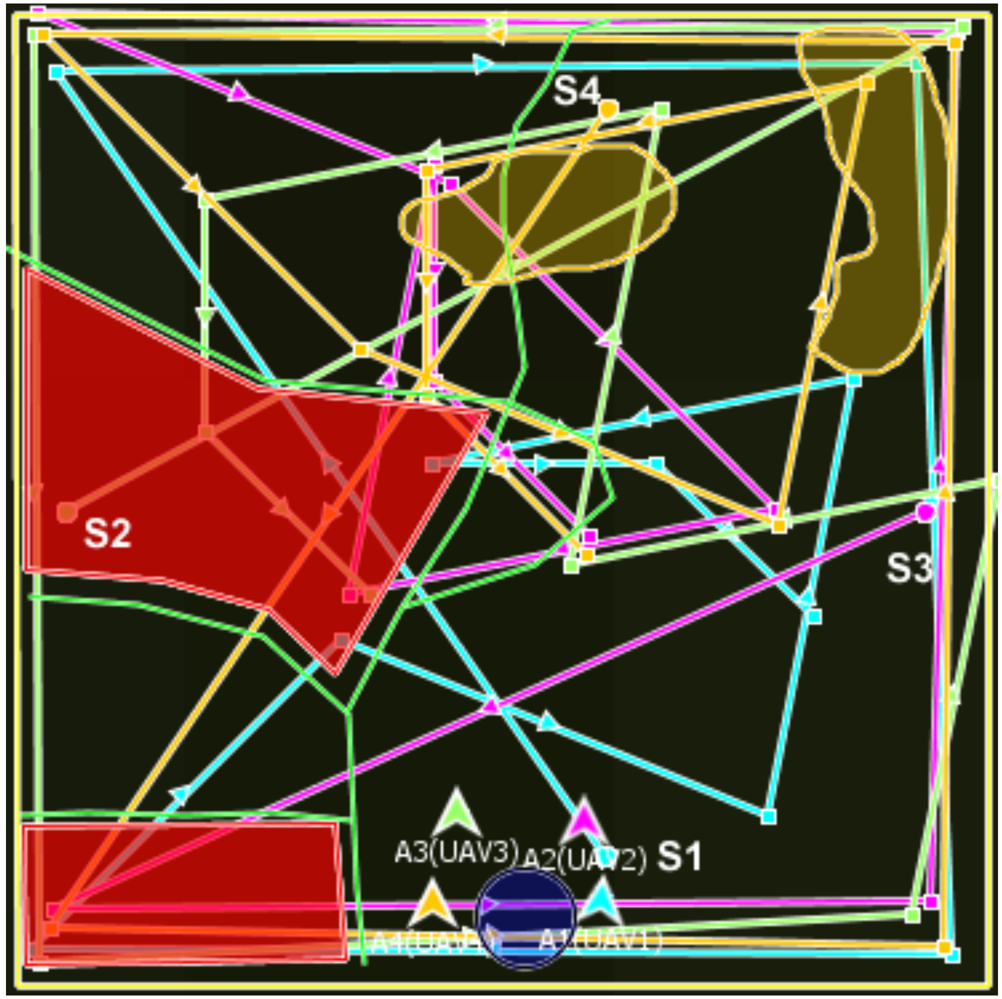


Figure 11: Multiple Agents Plan Generation

Interestingly, proposition and theorems could be applied to control the algorithm's waypoints generations. For example, proposition 1 proves that path divergence depends on the angle configuration of the search plan.

Proposition 1: A convex plan has more path divergence and coverage than its concave counterpart with the same edges.

Proof: Let n be the number of waypoints, and the convex of n be Conv(n). If m out of n edges of a concave plan has concave edges, then the convex of the concave plan is Conv(n-m), i.e. using Graham's Scan algorithm. From Euler's formula (number of triangles +number of waypoints = edges + 2), the number of triangles formed for the convex plan Δ_{convex} *will be:* $\Delta_{convex} = 2 + E - n$*, while for a concave plan, it will be* $\Delta_{concave} = 2 + E - (n-m)$*. Therefore,* $\Delta_{convex} > \Delta_{concave}$ *because* $n > n-m$ *for* $m > 0$ *and the edges are of the same size.*

Thus, Proposition 1 encourages convex plans more than concave ones in terms of path divergence. Therefore, during forest fire search plan generations, convex plans will have higher priority than concave ones when number of waypoints is higher. This is due to the path separation and lower redundant search offered by the convex paths.

3.2.2 Similar Strategies

Based on observation and analysis, most fixed-pattern methods follow specific geometric shapes with poor adaptability and scalability (Jensen-Nau et al., 2021; Kappel et al., 2020; Koenig and Liu, 2001). For example, parallel track, creeping line, expanding square, Zamboni, and sector search in Figure 3 to Figure 7 all follow a fixed structure and explore the search area sequentially. Sector search can be similar to Algorithm 1 in terms of adaptability. For example, in Figure 7, the angle and projection edges can be controlled to incorporate multiple agents. Despite the potential for adaptability, sector search lacks the following features compared to the proposed Algorithm 1 (DIMASS) algorithm.

i. Poor adaptability: sector search follows shapes that make them inappropriate for narrow space exploration, e.g., road mapping, whereas, using DIMASS, protocols are used to achieve adaptability by controlling angles, edges, and quadrants. Thus, adaptability is limited in sector search (as described in Table 10).
ii. Presence of large gaps (see Figure 7).

iii. Poor scalability: even though multiple agents can be handled by changing the angle and edges difference, controlling many agents can be challenging because of the nature of the path (sectors shapes).

Similarly, in the Voronoi method, agents visit the centre of the circumcircles of the Delaunay triangles (Hasegawa et al., 2012; McLain et al., 2001; So and Ye, 2005). This method resembles the Delaunay-centric and DIMASS algorithms with the absence of layering. Existing hybrid strategies act based on local protocols, which limits their system control and agent coordination (Chawla and Duhan, 2018, 2018, 2015; Ozkan and Kilic, 2022; Yang and Suash Deb, 2009; Yang et al., 2014). The proposed solution suggests both system (e.g., the concepts of reflections, refractions, seeds waypoints, waypoints layers structuring, etc.) and local protocols (e.g., sensor information based actions). In conclusion, different strategies have their unique styles. However, sector search and Voronoi method are more similar to the proposed DIMASS algorithm.

3.5 Performance Comparison

The performance of the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) algorithm is compared against popular strategies. For the fixed-pattern methods, parallel track (Bevacqua et al., 2015; Cabreira et al., 2019), creeping line (Bevacqua et al., 2015; Cabreira et al., 2019), Zamboni (João, 2012), expanding square (Bevacqua et al., 2015; Cabreira et al., 2019), and sector search (Bevacqua et al., 2015; Cabreira et al., 2019) are selected; and Lévy flight(Chawla and Duhan, 2018) is chosen as the pseudo-random methods candidate. The selection of the comparing candidates is based on their popularity and efficiency (Bevacqua et al., 2015; Cabreira et al., 2019; Chawla and Duhan, 2018; Jensen-Nau et al., 2021; Koenig and Liu, 2001). Comparison against the hybrids method was ignored due to being the combined versions of fixed-path and pseudorandom approaches, and the performance depends on the search area's structure e.g., making small jumps when target is detected. The evaluation was performed on the forest fire AMASE simulation scenario in Figure 2. UAVs team was tasked to search for the forest fires (as described in Figure 2 and the use case of Chapter 1).

3.6 Hypothesis

The experiment will test the following hypotheses:

a. Search methods perform differently on the outlined resources measures based on the mission constraints and their structure.
b. The adaptability of a search plan depends on its paths control elements.
c. An efficient, scalable, predictable, and adaptable search method under the outlined constraints is feasible.

The hypotheses will be tested against the outlined performance measures of Section 3.8.

3.7 Evaluation task

The evaluation task is to search the simulated area described in Figure 2. The agents and search area mimic their real presentation based on the physical experiment described in Chapter 6. The AMASE simulation entities e.g., fires, search area, UAVs, etc. were designed using their corresponding eXtensible Markup Language (XML) values. For example, fire has a tag of <Hazard> with corresponding characteristics such as spread rate, translation rate, etc. Note that, dynamic variables control e.g., fire spread rate value in consideration of wind speed can be handled by the corresponding Java controller class. Detailed process of AMASE simulation was described in Chapter 6.

Search methods were given the same resources and constraints. Due to the pseudorandom behaviour of the Lévy flight, mean and standard deviation of 15 experiments were taken. The number 15 is to justify the angle selection of Algorithm 1. That is, the angle change (i.e., from 0^o to 360^o) can be discretised into trenches of 24^0 (i.e., 360/15), which is equivalent as the lowest angle of the DIMASS solution . Therefore, for a fair comparison, 15 experiments of Lévy flight could be approximately equivalent to 15 angles, edges, and quadrants searching for Algorithm 1. The performance metrics were assessed across all methods. For example, coverage is measured by tasking agents to apply each method to explore the area within the assigned mission time. After the mission, the proportion of the cells covered is taken as the coverage value (i.e., as described in Section 3.8.1).

In terms of AMASE approaches implementation, the fixed-pattern approaches are implemented using a Java method that generates all waypoints based on a selected starting waypoint. For example, considering the creeping line implementation of Figure 3, the vertical and horizontal edge lengths have the same length within the search plan. That is, starting from the bottom-left waypoint (as initial waypoint), the next waypoint was obtained by adding the vertical edge (longer edge). Again, the third waypoint was obtained by adding the horizontal edge (short edge). As such, the waypoints generations implement a loop that constantly adds the vertical and horizontal edge length as far as the edge is within the search area. Note that, the value of the horizontal edges increases at every value whereas the vertical edge remains constant. As such, the end of the search area is determined by the vertical edge. The parallel track is a direct opposite of the creeping line. That is, the vertical edge keeps increasing while the horizontal edge remains constant. The expanding square method selects the initial waypoint at the middle of the search area instead of the extreme edge contrary to the parallel track, creeping line, and Zamboni approaches. Expanding square method increases the edge values across all directions for all opposing edges (see Figure 6). Similarly, Zamboni search starts from the extreme end of the search space and reduces the track length across each opposing edge (see Figure 5). The end of the waypoint is detected by when the deducting edge cannot be reduced further within the search space. Sector search varies angles and quadrants to mimic sector shapes as described in Figure 7. For the Lévy flight method, the waypoints generation follows Equation 5 and Equation 6.

$$P(\lambda) = \frac{1}{\pi}\int_0^{\infty} \cos(\lambda t).e^{-t\lambda^c} \quad 0<c\leq 2$$

Equation 5 : Lévy Distribution

where c is the constant value which ranges from 0 to 2. λ is the step size computed using Equation 6, and t is the time between two successive step sizes.

$$\lambda = \frac{U}{V^{\frac{1}{c}}}$$

Equation 6: Lévy Flight Waypoints Step Size

U and V are generated using a random number of generation function. That is, each waypoint is generated using Equation 6 after every waypoint visit. A sample code for the implementation of the approaches can be found in the supplemental documents.

3.8 Performance Measurement

The author measures the success of the generated plans using the following qualitative and quantitative measures.

3.8.1 Quantitative Performance Metrics

Quantitative measures of performance grade the agent's and mission's parameters utilisation based on functional requirements (i.e., the outlined parameters in Table 5). That is, maximising and minimising parameters based on their target optimisation assigned in Table 5. For instance, coverage needs to be maximised during exploration tasks, and redundant search needs to be minimised. Thus, if an agent X used search method A to cover 20Km^2 by spending 50% of its resources (e.g., energy, etc.); and covered 15Km^2 using search method B with the same resources, then A is more successful (efficient) than B in terms of coverage. Other parameters will be quantified using a similar real-valued cost function. The metrics are:

- **Energy**

Each of the agents' mission tasks (e.g., cruising, descending, ascending, etc.) consumes energy, measured as a percentage per second (%/s). For example, mini UAV ascending flight mode can consume 0.049%/s. This is very different from the other flight modes such as descending, loitering, etc. Thus, energy is measured as the percentage of energy consumed given a particular time interval and mission tasks. As described in Table 5, the lower the energy consumed relevant to the area covered, the better the search method. The energy consumption values for the simulation UAVs are derived from a physical drone experiment (detail of the experiment is described in Chapter 6).

- **Mission Time**

Mission time is the time taken to execute the mission. This is measured using the mission clock as described in Figure 1.

- **Memory Use**

This is measured using the algorithm space complexity. That is the amount of memory space needed to store the algorithm variables. This includes any variable exchange among agents.

- **Coverage**

This is a proportion of search area covered, i.e., cells s_i with path and sensing. Coverage is measured by segmenting the search area into a set of cells s_i of equal sizes and counting the proportion of cells with path. For example, in Figure 12, the coverage is 0.92 (i.e., 23/25, i.e., the uncovered cells were C11 and C19 marked with x out of the 25 total cells). Thus, coverage is the measure: $\sum_{i=1}^{i=n} S_i$ such that, $\forall s_i \ \exists \ w_i$x $r_v \epsilon s_i$, $\forall s_i \epsilon \ S_i$ where r_v is the sensor range and equal to the cells sizes i.e., $r_v = s_i$.

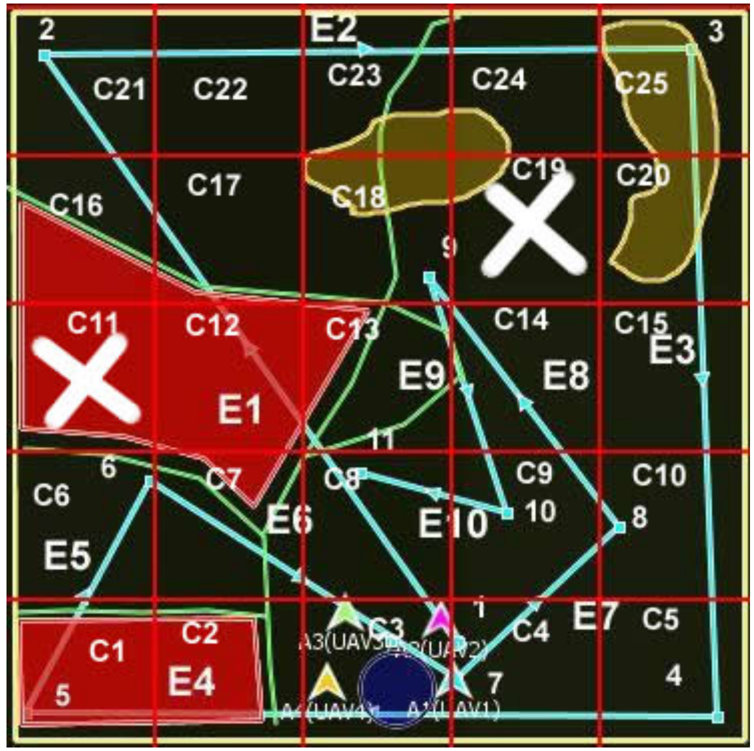


Figure 12: Example of Coverage Measurement

One of the limitations of measuring coverage this way is the partial visit. The partial visit is when a cell s_i has no agent's path but receives a partial visit (i.e., based on the sensor coverage). For example, consider the uncovered cells C11 and C19 of Figure 12, they were not counted as covered but they got a partial sensor coverage based on their neighbouring cells' paths. Thus, partial coverage is the measure: $\sum_{i=1}^{i=n} S_i$ such that, $\forall s_i \ \exists \ w_i \notin S_i$, and $r_i \epsilon s_i$, $\forall s_i \epsilon S_i$, and r_i is the sensor coverage corresponding to S_i. Due to partial visit limitation, I introduce the path divergence.

- **Path Divergence**

Path divergence measures how the search path spreads across the search area. This is measured as the summation of the Delaunay triangles of the search plan waypoints $\sum_{i=1}^{i=n} areaDelaunay(\Delta_i)$. The function $areaDelaunay(\Delta_i)$, i ϵ[1,N] computes the area of each Delaunay triangle. Again, one of the issues of measuring coverage using path divergence is that larger triangles tend to have more larger values. As such, balancing the number of triangles and the calculated area could guide in understanding the area covered. For example, if four triangles give 200kilometers square and ten triangles provide the same amount, then probably the later value has a higher number of uncovered cells (based on the search methods paths structure); this can be balanced further by creating closer waypoints. Thus, the path divergence and coverage can measure the portion of the search area covered.

- **Redundant Search**

Redundant search measures how search method performs a repetitive search. This can be measured by counting the number of subsequent overlapping waypoints within a search plan. Redundant search can be categorised into intra-agent and inter-agent redundant search. Intra-agent redundant search refers to overlapping waypoints within a particular UAV plan, and the inter-agent redundant search is the overlapping waypoint with other agents' waypoints.

- **Agents Interactions**

This is measured as the number of times required to exchange information among n agents given inputs waypoints w to maintain resource utilisation. For example, given two agents using pseudorandom search method, avoiding redundant search requires *n-1* at least interactions at each waypoint generation w_i. Thus, the number of interactions is $O(nN_w)$ where N_w is the number of waypoints. Thus, the total number of interactions are n-1 x N_w.

- **Time Complexity**

Time complexity measures the computational operations needed to implement the algorithm for a team of n agents. This is quantified by the run time complexity required given n waypoints inputs. For example, if the search method's computational operations are directly proportional to the input waypoints n, the outcome is O(n). Thus, the time complexity can be used to measure the computational power needed. One of the limitations of time complexity is the omission of implementational effort. I propose using McCabe's cyclomatic complexity (McCabe, 1976) to address the challenge.

- **McCabe Cyclomatic Complexity**

The McCabe cyclomatic complexity measures the implementation effort requires. It counts the number of loops, conditional statements, and methods as cyclomatic units. I used the eclipse metric plugin[5] to measure the cyclomatic complexity. Note that the result depends on the implementation, and I did my best to minimise unnecessary loops, conditional statements, and methods.

3.8.2 Qualitative Performance Measures

Qualitative measures grade the search plan in terms of non-functional features, and these are scalability, predictability, and adaptability.

- **Scalability**

Scalability is the ability to address multiple agents' plans with stable resources. I measure scalability using number of interactions, time, space, and cyclomatic complexities measures. Therefore, scalability can be in terms of coordination (measured using number of agents interactions), memory use (measured using space complexity), computational power (measured using time complexity) and implementational (measured using cyclomatic complexity). Each of these types of scalabilities has different implications on the agents. For example, implementation on small agents with very low capacity would prioritise memory use, computational power, and

[5] https://marketplace.eclipse.org/content/codecity

implementations scalability over coordination scalability. Similarly, coordination scalability will have a higher priority when monitoring many agents. Thus, a preferable outcome is a low space, time, and implementational complexities.

- **Predictability**

Predictability is the ability to estimate agents' location given specific parameters. For instance, if the distance between the waypoints 2 to 3 of Figure 12 is 3KM, then assuming a stable weather report (e.g., upwind/downwind flights), a UAV with a speed of 30m/s will be 1.8Kilometers (30m/s x 60s /1000) away from waypoint 1 at its first minute. Therefore, a search method will be considered predictable if it will allow the agent's location prediction across n waypoints with a certain level of accuracy.

- **Adaptability**

Adaptability measures how the search method can be utilised to perform different tasks, e.g., mapping, searching, etc. I measure adaptability by counting the number of controllable elements of the search plan paths, i.e., angles, quadrants, and edges.

3.8.3 Results

Table 6 to Table 9 describe quantitative metrics evaluation results for the agents and mission parameters in Table 5. Table 6 shows the agents' coverage and path divergence measures across different search methods.

Table 6: Quantitative Measures Result- Coverage Measures

S/N	Approach	Coverage	Path Divergence (meter square)
1	Delaunay-centric algorithm	1	834.83
2	Delaunay-Inspired Multi-agent Search Strategy (DIMASS)	1	962.33
3	Lévy flight (Chawla and Duhan, 2018)	0.64+/-(0.28)	434.17+/-(281.20)
4	Parallel Track (Bevacqua et al., 2015; Cabreira et al., 2019)	1	128.03
5	Creeping Sleep (Bevacqua et al., 2015; Cabreira et al., 2019)	1	252.64
6	Sector Search (Bevacqua et al., 2015; Cabreira et al., 2019)	0.69	588.55
7	Expanding Square (Bevacqua et al., 2015; Cabreira et al., 2019)	0.75	172.28
8	Zamboni Search (João, 2012)	1	518.06

Table 6 shows the coverage and path divergence performance of the methods. Most of the fixed-path and the proposed method shows good performance in terms of coverage better than the pseudorandom method. However, the fixed-path method shows poor performance in terms of path divergence due to their poor flexibility (i.e., as a result of poor flexibility which affects the Delaunay triangles area). The Delaunay-Inspired Multi-agent Search Strategy (DIMASS) method shows good performance due to its flexibility (i.e., it produces well-separated and well-spread waypoints). Table 7 describes the complexity metrics evaluation.

Table 7: Quantitative Measures Result- Complexities Measures

S/N	Search Method	Cyclomatic Complexity for n agents	Time Complexity	Space Complexity	Least number of Agents Interactions to avoid redundant search
1	Delaunay-centric algorithm	19	$O(n^2 \log n)$	$O(n^2 w_z)$ where w_z is the size of the waypoints.	$O(n_l a_n)$ where n_l is the number of waypoints with length less than the diagonal of the sensor range and a_n is the number of agents.
2	Delaunay-Inspired Multi-agent Search Strategy (DIMASS)	2	$O(n)$	$O(nw_z)$	$O(n_l a_n)$where n_l is the number of waypoints with length less than the diagonal of the sensor range and a_n is the number of agents.
3	Lévy flight(Chawla and Duhan, 2018)	3	$O(n)$	$O(nw_z r)$ where r is the memory needed to store the random number seeds of Equation 6.	$O(na_n)$

4	Parallel Track(Bevacqua et al., 2015; Cabreira et al., 2019)	7	$O(n^2)$	O(n)	$O(n_l a_n)$ where n_l is the number of waypoints that are less than the width of the sensor range.
5	Creeping Sleep(Bevacqua et al., 2015; Cabreira et al., 2019)	9	$O(n^2)$	O(n)	$O(n_l a_n)$ where n_l is the number of waypoints that are less than the width of the sensor

Table 7 shows the algorithms complexities performance. In terms of time complexity, the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) and Levy flights show good performance due to independent plan generation. The fixed patterns have a higher value due to vertical and horizontal edge loops addition. The Delaunay-centric algorithm performs poorly because of the Delaunay-triangulation processes.

Table 8: Quantitative Measures Result- Agents and Mission-Based Featured

#	Search Method	Energy Used	Time Spent(seconds)	Number of waypoints
1	Delaunay-centric algorithm	9%	3909.6	11
2	Delaunay-Inspired Multi-agent Search Strategy (DIMASS)	9%	4095.3	11
3	Lévy flight(Chawla and Duhan, 2018)	19.13 +/- (1.54%)	4095.3+/-3	10.73 +/- 7.13

4	Parallel Track(Bevacqua et al., 2015; Cabreira et al., 2019)	14%	2758.9	10
5	Creeping Sleep(Bevacqua et al., 2015; Cabreira et al., 2019)	18%	3562.8	10
6	Sector Search(Bevacqua et al., 2015; Cabreira et al., 2019)	16%	3109.0	9
7	Expanding Square(Bevacqua et al., 2015; Cabreira et al., 2019)	19%	3774.44	15
8	Zamboni Search(João, 2012)	87%	17693.1	33

Table 8 describes the energy and mission time performance of the algorithms. The energy is measured by stopping the simulation scene (using the pause/play button of Figure 2) and taking the value of the energy consumed. The energy consumption considers various flight modes e.g., descending, ascending, etc., and their respective consumptions rate as derived from a physical experiment (Chapter 6). Similarly, the mission time is recorded together with the energy value. The redundant search measurement of Table 9 considers different sensing ranges to evaluate the performance. The evaluation is based on the two UAVs' missions as describe in Figure 9. The sensing range is measured as the percentage of the search area.

Table 9: Quantitative Measures Result- Redundant Search Result

#	r_v	Delaunay-centric algorithm	Delaunay-Inspired Multi-agent Search Strategy (DIMASS)	Lévy flight(Chawla and Duhan, 2018)

		Intra-agent redundancy(n/11)	Inter-agent redundancy(n/11)	Intra-agent redundancy(n/11)	Inter-agent redundancy(n/11)	Intra-agent redundancy μ(σ) 10.34(7.13)	Inter-agent redundancy μ(σ) 10.34(7.13)
1	5%	0	0	0	1	4 (2)	4.7 (2.11)
2	10%	4	1	2	2	9 (4.76)	9.8(4.80)
3	15%	5	2	3	3	7.9 (3.75)	9.8 (4.13)
4	20%	5	3	4	4	9.4 (3.37)	10.9 (3.11)
5	25%	5	3	4	4	11.2 (2.48)	13.3 (2.54)
6	30%	6	3	5	4	11.5 (3.84)	14.2 (4.09)
7	35%	6	4	6	4	14.9 (6.89)	18.2 (8.04)
8	40%	6	5	6	4	12.6 (3.17)	15.3 (3.89)
9	45%	6	5	6	5	17.6 (3.60)	18.26 (2.77)
10	50%	6	6	6	7	18 (2.83)	21.6 (3.89)

The results in Table 9 show the redundant search performance (based on the number of redundant waypoints) for the flexible methods (i.e., the Delaunay-centric, DIMASS, and the Lévy flight methods). The intra-agents redundancy counts the number of overlapping waypoints among agent's waypoints. For example, Table 12 #2 shows the number of overlapping waypoints of an agent using the

Delaunay-centric algorithm when the sensor range is 5% of the search area is 4 number of waypoints. This is less effective than the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) counterpart with 2 as the number of redundant waypoints. The Lévy flight shows higher redundant waypoints within the agent's mission. As expected, increasing the sensor range keeps increasing the number of redundant waypoints. Similar results are expected when the search area is kept constant, and the number of agents is increased (i.e., based on the results in Table 9). Thus, this remains the main reason for the author's use of two UAVs.

Table 6 to Table 9 describes the results for various quantitative metrics across varying parameters. For example, #1 row across the tables describes the performance of the Delaunay-centric algorithm. In terms of coverage from Table 6, all the cells of the search area were covered (i.e., each of the cells of the search has at least a visit) by the agents. Remember, the coverage is measured by segmenting the search into cells of equal size with the sensor range. The numbers of waypoints generated were 11 which are the same as the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) approach because they utilise similar protocols. The path divergence was 834.83km^2 (measured as the sum of the area of the 11 waypoints Delaunay triangles). The cyclomatic complexity reported in is based on the author's implementation and the value is 19 (measured using the eclipse metric plugin[6]). The experiments for the Lévy flights were repeated 15 times due to their randomness; and the mean and standard deviation were reported accordingly. The results from Table 6 to Table 9 show good performance of the proposed method across various parameters.

- **Qualitative Metric: Scalability**

Definition 3.2 (waypoints reflection), Definition 3.3 (waypoints refraction), and protocols provide a way of controlling the higher number of agents (i.e., coordination scalability based on the low number of agents interactions). The low value of cyclomatic, time, space, and number of agents interactions complexities shows better performance for the implementational, computational power, memory use, and coordination scalabilities of the proposed DIMASS solution due to the low complexities values.

[6] https://marketplace.eclipse.org/content/codecity

- **Qualitative Metric: Adaptability**

Table 10 describes the adaptability results based on the number of controllable path elements of the search methods. The controllable path elements selected are angles, quadrants, and edge lengths.

Table 10: Qualitative Measures Result- Algorithms Adaptability Comparison

Algorithm	Number of Controllable Path Elements	Controllable Path Elements: {Quadrants, angles, edges}	Comments
Delaunay-Inspired Multi-agent Search Strategy (DIMASS)	3	{Quadrants, angles, edges}	All controllable path elements can be controlled
Lévy flight (Chawla and Duhan, 2018)	0	¬{Quadrants, angles, edges}	None of the controllable path elements can be controlled
Parallel track (Bevacqua et al., 2015; Jensen-Nau et al., 2021)	1	{¬Quadrants, ¬ angles, edges}	Edges can be controlled, whereas angles and quadrants are fixed because angle has to be either 90^{o} or 180^{o}.

Creep lining(Bevacqua et al., 2015; Jensen-Nau et al., 2021)	1	{¬Quadrants, ¬ angles, edges}	Edges can be controlled, whereas angles and quadrants are fixed because angle has to be either 90^{o} or 180^{o}.
Sector search(Bevacqua et al., 2015; Cabreira et al., 2019)	2	{¬Quadrants, angles, edges}	Changing quadrants configuration of sector search will make it not to be in sector form anymore.
Expanding squares (Bevacqua et al., 2015; Cabreira et al., 2019)	1	{¬Quadrants, ¬ angles, edges}	Edges can be controlled, whereas angles and quadrants are fixed because angle has to be either 90^{o} or 180^{o}.
Zamboni Search(João, 2012)	1	{¬Quadrants, ¬ angles, edges}	Edges can be controlled, whereas angles and quadrants are fixed because angle has to be either 90^{o} or 180^{o}.

Based on the result in Table 10, the proposed solution offers the most adaptable solution. This means it can be applied to perform many other tasks e.g., mapping, searching, etc. For example, roads within the search area can be tracked using the proposed solution by changing the angles, quadrants, and edges lengths. Probability value can be assigned to interesting k-previous waypoints (past waypoints), e.g., junctions. When the agent finishes its current task, it could then return to the important k-previous location and continue from there. The locations priority probabilities marks could be stored in the agent's short term, medium or long term memory based on the saliency of the waypoint. This process resembles the operation of simulated annealing in terms of waypoints storage in short-term, medium-term, and long-term memory based on importance; and smart Rapidly-exploring Random Tree (RRT) in terms of waypoints radius assignment (Nasir et al., 2013; Varty, 2017) i.e., during roads tracking. Thus, the proposed solution can be utilised for many other tasks.

- **Qualitative Metric: Predictability**

From the fixed-pattern methods of Figure 3 to Figure 7 and the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) methods (Algorithms 1), agents' location can be estimated based on their speed and control protocols. For example, from Figure 9 (solution of the proposed DIMASS), if a UAV starting from waypoint 1 has a speed of 30m/s, assume the length between waypoints 1 and 2 is 5.6KM; then, at the second minute of the UAV mission, it is expected to be 3.6KM away from the initial point (i.e., 2 x 60 x 30/1000). Other reports could be incorporated, e.g., upwind and downwind accelerations and retardations. The predictability can be graded further to consider n number of waypoints. For example, the prediction ability can span across all waypoints for the fixed pattern and the proposed solution (because waypoints are structured). This is not possible in the case of the Lévy flight. Thus, the predictability feature could help the PCs' data collection (by arranging rendezvous). Lévy flight shows that it is purely pseudo-random; as such, predicting agents' future location will be very difficult or even impossible. One could argue that limiting the Lévy flight's seed waypoint range i.e., U and V of Equation 6 could help in improving predictability. Interestingly, this does not affect predictability, as described by the results in Table 11.

Table 11: Limiting the Random Number Seed Range for Levy Flight

#	Random Number Seed Range for U and V of Equation 6	Avg. Path Divergence Square Kilometre (KM^2) μ(σ)	Avg. Number of Waypoints(σ)
1	1-5	273.58(73.34)	7.6(1.16)
2	1-10	288.39(61.75)	10.8(1.90)
3	1-15	260.34(94.42)	6.2(1.46)
4	1-20	275.86(119.35)	7.4(1.58)
5	1-25	193.87(43.64)	5.1(0.64)
6	1-30	293.29(83.3)	7.8(1.64)

The Levy flight result in Table 11 was obtained similarly to the result in Table 8. The path divergence is measured by summing the area of the Delaunay triangles of the waypoints plan. Table 11 shows that limiting the range of the random number seeds for the Lévy flight method does not affect its predictability. For example, the path divergence in #1 with range (1-5) has a higher path divergence better than #5 and #6 with ranges (1-25) and (1-30). This shows low structure and difficulty in predicting agents' activities when Lévy flight is applied.

3.4 Discussion and Conclusion

Based on the results in Table 6 to Table 11, the proposed solution utilises the agent's resources better than the existing methods, although in some cases, some of the existing techniques produce a good performance (e.g., the zero redundant search performance of Zamboni and Sector Search despite the 5% sensor range). The performance of the proposed method is based on the applied fixed protocol (i.e., shorter edges are half of the opposing longer edges). Thus, the best solution can be obtained by adjusting the angles, quadrants, and edge configurations. This can be done in two ways: (i) by imposing a rejection protocol on any edge that is less than the diagonal of the sensor range, or (ii) agents need to interact with other agents to share information on waypoints (i.e., waypoints with edges less than the diagonal of the sensor range). The first option reduces the number of agents' interactions and hence improves the coordination scalability, although this will only solve the intra-agent redundancy. Inter-agent redundancy in this approach can be avoided by adjusting angles, quadrants, and edges at the premission state or using the later method. For example, based on the seed waypoints, layers inter-agent redundant waypoints can be detected and avoided, especially when plan generations among agents are in sequential order. That is, agent 1 will not consider adjusting its waypoints and agent 2 will avoid redundant search with agent 1's waypoints based on the defined seed waypoints and system protocols etc. The second option can resolve both intra-agent and inter-agent redundancy with the overhead of additional agents' interactions for at least 1 interaction across all agents per each redundant waypoint (as described in Table 7).

Summarily, the proposed solution proved higher performance across the quantitative and qualitative measures. The clear difference between the proposed method and the existing hybrid methods is that, the existing hybrid methods imposed their protocols on local agents instead of the whole team. The proposed solution considers system control protocols that coordinate the agent's search activity and utilise their resources throughout the agents mission. The protocols also consider agents' activities efficiency, e.g., avoiding redundant search instead of focusing on coordination. Thus, the proposed hybrid solution focuses more attention on the efficiency of the system protocols. The discussion chapter (Chapter 9) describes how the proposed method can easily be applied on real UAVs.

3.4.2 Further Thoughts on the Agents Search Algorithms

The primary concern of this chapter is to obtain a resource-efficient search method based on the imposed constraints. Results proved the efficiency of the proposed solution on the outlined measuring parameters. Further investigation on more control theorems and propositions is marked as future research. Finally, for the method to be more valuable and supportive to the search mission, the question, "*How can we manage the Situation Awareness of distributed agents?*" is essential. Therefore, the objective of the following chapter is to address this challenge.

4 Chapter 4: Towards Agents Distributed Situation Awareness Modelling

In this Chapter, the concept of DSA is applied to a team of distributed agents from Chapter 1 and extends the idea using formal properties of the Bayesian Belief Network (BBN). In particular, the aim is to show how BBN can define DSA and how such a network can be utilised to handle various DSA challenges, e.g., flexible belief measurement, prediction, uncertainty handling, DSA schemata (phenotype and genotype) activities etc. The outcome shows a better result than the existing methods of concept map, propositional network, fuzzy logic, ontology, etc.

4.1 Introduction

Distributed Situation Awareness (DSA) originated from Hutchins (Hutchins, 1995) concept of Distributed Cognition. This involves SA management from a team of distributed agents accessing various system information and playing varying roles. For example, determining an aircraft's speed (e.g., in terms of an approach to landing) involves several agents (e.g., sensors in the plane, instrument displays in the cockpit, flight manuals, 'speed bugs', pilot and co-pilot, Air Traffic Control etc.). Hutchins proposed that no single agent knows the speed; instead, knowledge is distributed across agents. Following this, Distributed Situation Awareness (DSA) proposes that one needs to take a systems-level perspective on SA (Kitchin and Baber, 2017; Rosário et al., 2021; Salmon et al., 2008). Since then, DSA employs concept maps (e.g., as in Figure 13), ontology, fuzzy logic, proposition networks, etc. (Nguyen et al., 2019; Salmon et al., 2009; Stanton et al., 2006; Stefanidi et al., 2022; Stewart et al., 2008; Suhail et al., 2022) to model the system-level SA (e.g., as described in Figure 13). In this Chapter, Bayesian Belief Network was applied to address the existing methods' challenges, such as the issue of the good interface, real-time system SA update, belief measurement, possibility for prediction and uncertainty handling etc. As such, this chapter is aimed to answer the subquestions "how could agents' sensor states be transformed to manage SA in a distributed team?" and the "how to best model the SA of the distributed agents?" as part of the main question: "RQ2. How can we manage the Situation Awareness of distributed agents?".

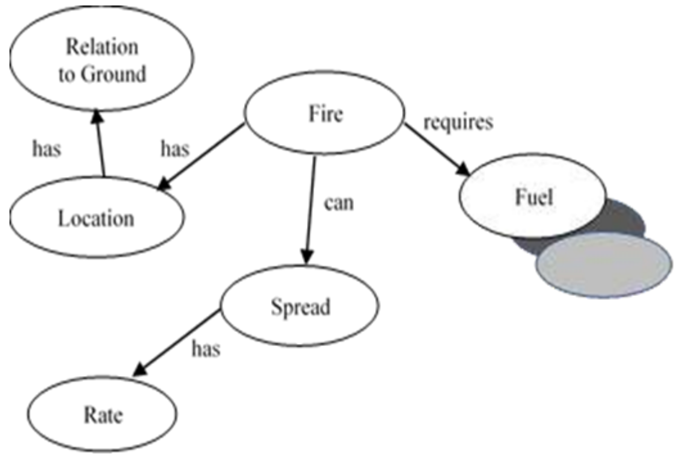


Figure 13: Example of Concept Map of Fire Scenario

One can annotate concept maps to indicate which entities use a concept, e.g., in Figure 13, fire node is linked up with the concept of 'fuel' and use by different other information. However, the agents might define the concept differently. For instance, the concept "Fuel" could have different meanings and importance among agents, e.g., the fire spread agent would prioritise the "Fuel" concept better than evacuation agents (i.e., in the case when roads are available). These differences in meaning can be reflected by the context in which the concept is used, i.e., how it is related to other concepts. Existing methods (concepts maps, propositional network, fuzzy logic, etc.) provide qualitative descriptions of the 'system SA'; created *a priori* (from expert knowledge or Standard Operating Procedures, SOP) or *a posteriori* (from analysis of agents' interaction logs). Consequently, these are limited by having issues with real-time system SA update (being post-hoc), poor or absence of prediction supports and uncertainty handling, lack of belief measurement, and lack of agents interaction analysis, which this Chapter aims to address by applying Bayesian Belief Network.

A Bayesian Belief Network (BBN) is a directed acyclic graph G(N, E), with relations between nodes, N, and directional edges, E, which can be captured using adjacency matrix, C_{ij}. Each node of the BBN (e.g., 'Fire' node of Figure 15) has a different set of states, (S_i) i∊[1, N], (e.g., present or absent for fire node) with assigned probability values $P(S_i)$. The state's probability values are specified using a prior generation algorithm, $P(S_i)$, which can be derived from sensor values (e.g., as described in Section 4.8) or Subject Matters Experts (SMEs) as described in Section 4.7. The BBN node states probabilities update defines the local SA, e.g., fire presence or absence based on the state's probabilities. As the situation unfolds, the BBN must be updated either by introducing new nodes (this is addressed in Chapter 8), by altering links between nodes (discussed in Chapter 8 as well) or by changing the probabilities of the states to reflect the current system SA (Section 4.8). Thus, the chapter describes how BBN can be formalised to present the system DSA.

4.2 An Example of the System DSA using BBN

The approach is illustrated using the thesis use case of responding to forest fires discussed in Chapters 1 and 3. Fire has been spotted at two locations (the polygons in the top right of Figure 14). The agents' mission is to find the fires, monitor them (as discussed in Chapter 3) and manage the system SA. In this instance, the 'Situation' is defined by the phenomena to which the network is responding, the location (S_i) in which these phenomena occur, and the activities that can be performed in response to the phenomena. In other words, at the system level, the situation can be defined in terms of Search Area Phenomena and Activities. This thesis assumes that elements of the situation will be known at this high level, even if their parameters are not, which will be collected by the agents (micro or mini UAVs of Chapter 3) using their sensors. For example, 'fire' might be defined in terms of {location, spread, fuel type, fuel condition etc., as described in Figure 15}. Updates of the states of the nodes of the BBN in Figure 15 will be done by the agents' sensors. Therefore, the main challenge to be addressed in this Chapter is how these Search Area Phenomena (in the form of BBN nodes from Figure 15) and their corresponding Activities can be presented so that agent perception models DSA in a comprehensible and predictable manner. For example, Figure 15 describes a BBN to present the forest fire spread behaviour. This BBN is assumed to reside at the Picture Compiler (PC) level and is updated by various agents. Thus, the challenge of maintaining DSA within the system needs to consider the agents' capabilities and the changing nature of the search area phenomena.

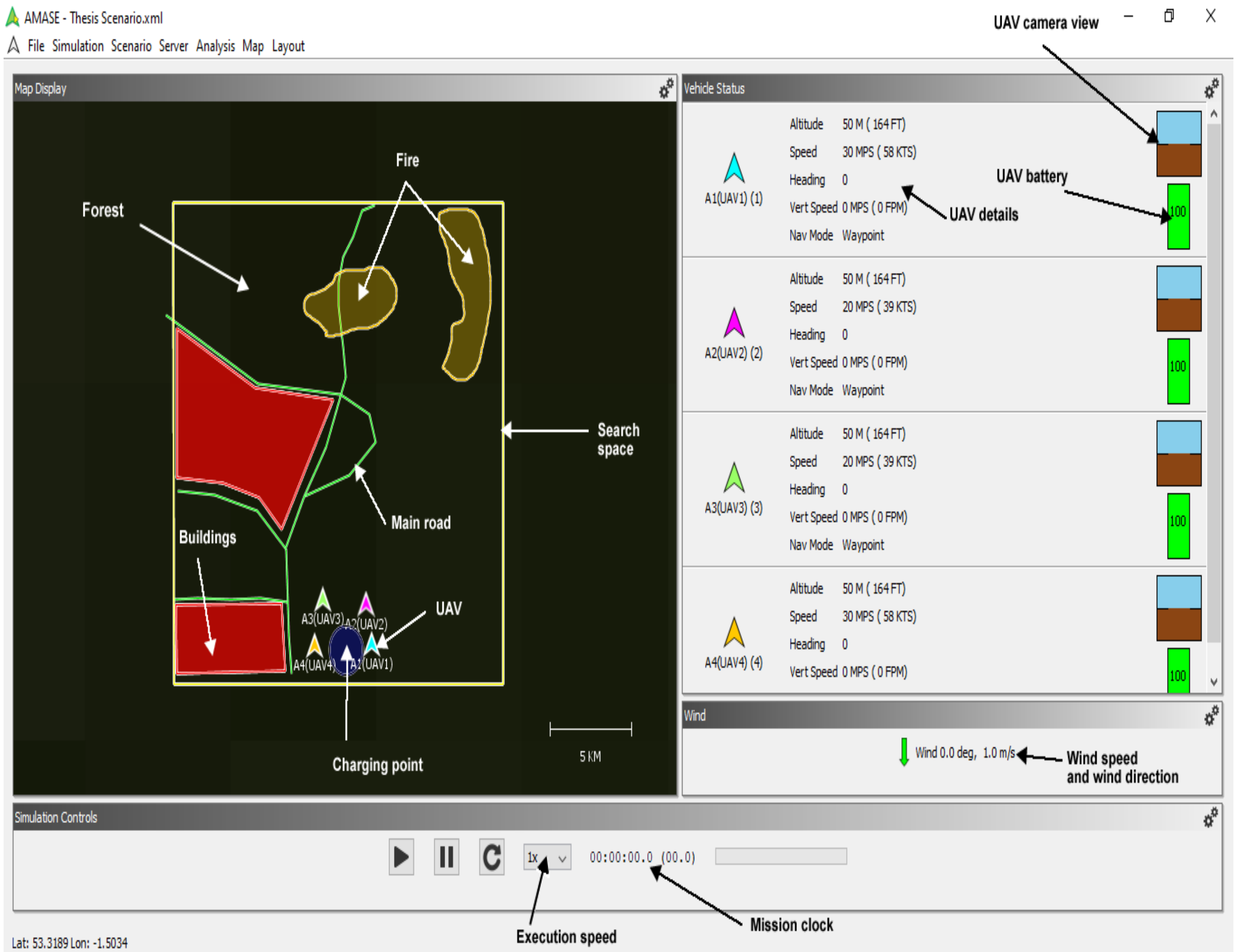


Figure 14: AMASE Example of Search Area Situation (Replica of Figure 2 of Chapter 3)

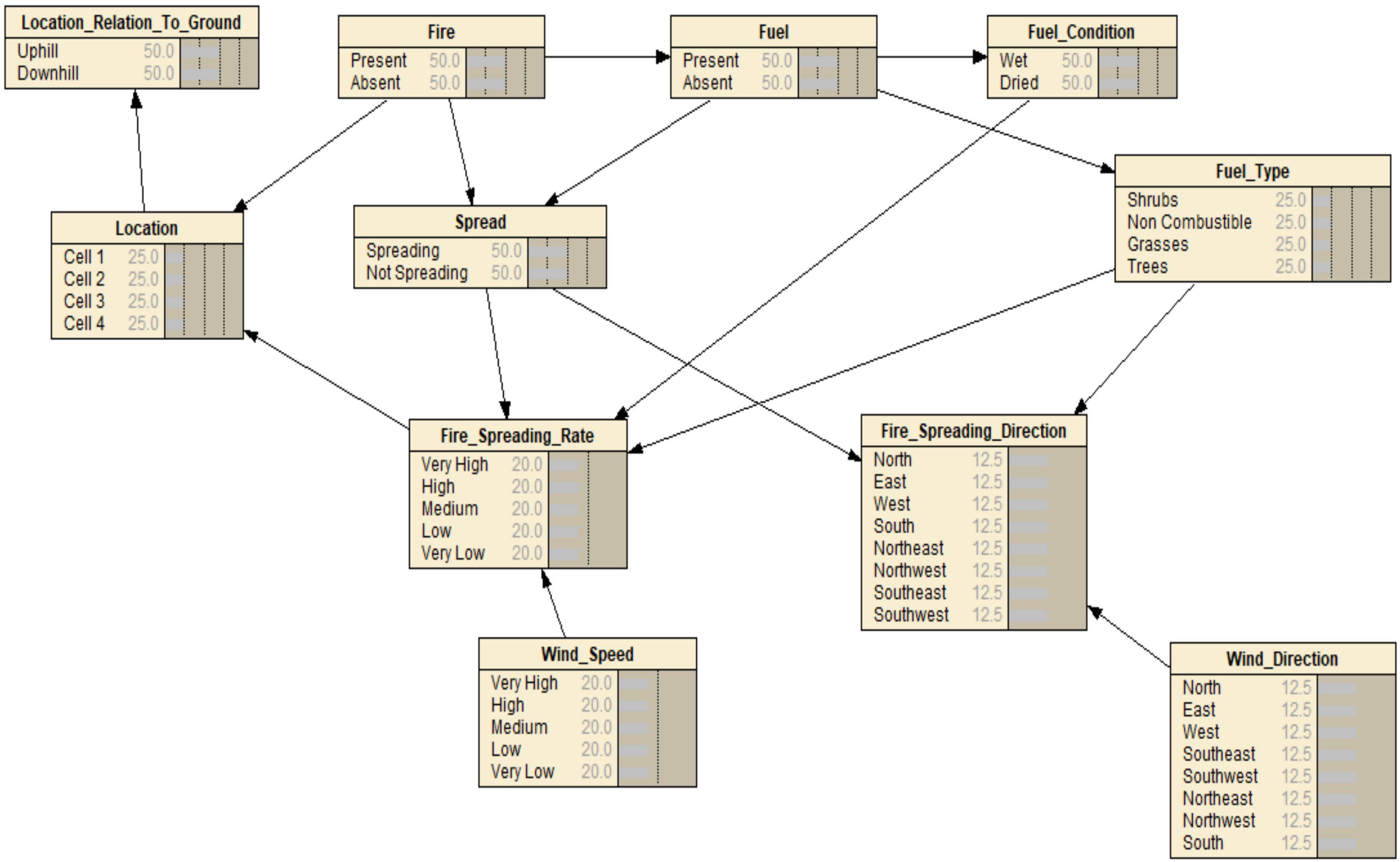


Figure 15:Initial BBN (produced in Netica) for Forest Fire Spread Monitoring

4.3 Performance Measures

The performance measures were based on metrics and features that support DSA as outlined in literature. The following metrics are selected:

i. Belief measurement: based on the configuration of the agents' sensors in Chapter 1, there is a need for presenting different Search Area Phenomena (in form of nodes). For example, informing the model about the presence of fire not only in a Boolean form (e.g., fire presence or absence) but also reflects sensor reliability. This mode of belief measurement is essential due to multiple sensors and search area dynamism. For example, detecting fire using visual sensors during night-time is more reliable than during daytime due to possible confusion from fire-like objects. Thus, there is no single sensor with an absolute priority in all situations. This is measured by how the SA modelling tool allows non-Boolean information presentation (i.e., not presenting information as present or absent only but in a flexible manner).
ii. Prediction of plausible phenomena: the ability to incorporate different system information to predict future Search Area Phenomena. This is measured using the prediction error rate after model training.
iii. Adaptable and up-to-date presentation: this is the ability to reconfigure DSA to the current situation of the search area. This can be measured as the ability to be reconfigured given various situations.
iv. Uncertainty Handling: Like the prediction task, the estimation of missing or lost information relating to phenomena states due to hardware/software problems, sensor failure, etc., needs to be addressed. This is measured using the error rate of the estimation model.
v. Knowledge reusability: the model needs to have features for incorporating existing knowledge from both human and automation agents. Human understanding of Standard Operating Procedure (SOP) is beneficial to the mission operation. Although this thesis has no focus on the human aspects, values from documented SOP will be used. Thus, SOP knowledge

from the Subject Matter Experts (SMEs) is essential to the mission planning. This is measured as a qualitative feature (i.e., ability to incorporate the previous system information to improve the SA management).

vi. Presentation of multiple states of a search area phenomenon: search area phenomena have different states over a period based on their dynamism. For example, considering the fuel type concept in Figure 15, it is possible states could be grasses, shrubs, trees, and a combination of different types based on the location. Thus, the model needs to present alternative interpretations of the current situations based on the perceived information. This is also measured as a qualitative feature.

vii. Heterogeneity handling: the system configuration describes a variety of sources of information from agents to cover different phenomena in different search area situations. Contextual understanding of the information is critical to the DSA (Hutchins, 1995; Merino et al., 2006; Stanton, 2016). This feature is measured by how the DSA modelling tool addresses heterogeneous agents conflicts.

Therefore, the outlined features and metrics will serve as the evaluation performance measures for comparing the proposed Bayesian Belief Network with the outlined existing strategies. This chapter describes only the method and modelling algorithms, the evaluation of the metric was discussed thoroughly in Chapters 5, 7, and 8.

4.4 Hypothesis

It is hypothesised that an effective SA modelling tool exhibits the outlined metrics (Section 4.3) management. The experiments task is to prove this claim based on the use case discussed in Chapter 1 and the agents' mission of Chapter 3.

4.5 Experiment Design and Task

The experiment configuration remains the same as the one in Chapter 3. The task is to develop a BBN model that describes the forest fire phenomena and their logical relations to reflect the system DSA. The experiment is set up by modelling the scenario (e.g., elements of Figure 2, i.e., fires, houses, etc., using a BBN of Figure 15) and other dynamic search area phenomena such as wind speed, wind directions, fuel types, etc., for each location (detailed description of the modelled as described in Chapter 6).

4.6 BBN for DSA Modelling

The simple UAVs will be submitting their sensor values as text keywords. For example, whenever a fire detecting UAV detects a fire, it will update a .csv (comma-separated values) onboard memory file with aligned time and location values; and submit this update to the respective PC or host. The update can be achieved simply using file read/write methods (e.g., using file handling library of Java). Therefore, the PC update involves simple UAVs updating their respective columns (nodes labels) of the .csv file.

The experiment modelled the BBN using NETICA Java API[7] and integrated it with AMASE using Eclipse IDE (Integrated Development Environment). The BBN states update follows these steps:

i. AMASE UAVs update of sensor state using text keywords e.g., fire present or absent
ii. The received text recording in a .csv memory file in form of columns and rows (detail description was done in Chapter 6)
iii. The .csv file is used to update the priors of the situation BBN using sensor keywords text of (i)

This chapter proposes Bayesian Belief Network (BBN) to model the system DSA. Visually this has similarities to the concept map, propositional network, and ontology (in that each node represents Search Area Phenomena, and the relationship between phenomena is managed by the directional edge). However, BBN assumes a hierarchical dependency (which imposes a different logic from the 'grammar' that defines concept maps, ontology, or propositional network).

Each system mission will have a high-level BBN, perhaps residing at the PC or host level. This BBN will be an abstract version of 'situations-like-this' and can be constructed *a priori* from SME's knowledge of SOP or learned from the previous mission data. Thus, an initial version of the BBN would involve phenomena characteristic of similar situations based on the agents' goals. As the mission

[7] https://www.norsys.com/netica_api.html

progresses, the BBN becomes specific to the developing situation based on the simple agents' sensor information which updates the nodes' states probabilities. The basic process for constructing the BBN is as follows:

1. Define Mission type, e.g., 'response to a forest fire monitoring' as described in Figure 15.
2. Define the BBN Phenomena (nodes) based on the SOP. In this step, there is a need for defining the information needed to understand various mission activities, e.g., fire spread, etc. For this chapter, the mission goal is the understanding of forest fire spreading behaviour. From the simulation in Figure 14, the fire moves in response to wind direction, wind speed, location relation to ground (uphill or downhill) because fire moves faster uphill, based on fuel type and fuel condition (e.g., dried grass, wet grass). Therefore, the following Situation nodes are needed: {Fire, Fire Spreading Rate, Fire Spreading Direction, Wind Speed, Wind Direction, Location, Location Relation to Ground, Fuel, Fuel Condition, Fuel Type} in Figure 15.
3. Specify the phenomena states. The state of each phenomenon needs to be identified. Again, continuous variables need to be discretised to allow state structuring. For example, nodes of Figure 15 can have the following states:

   {Wind Direction: North, East, South, West, Northeast, Northwest, Southeast, Southwest}

   {Fire: Present, Absent}

   {Wind speed: Very high(greater than 8m/s), High(greater than 6m/s to 8m/s), Medium (greater than 4m/s to 6m/s), Low(greater than 2 to 4m/s), Very Low(0-2m/s)} {Location: Cell_1, Cell_2, Cell_3, Cell_4} i.e., the environment is segmented into four cells.

   {Locations Relation to Ground: Uphill, Downhill}

   {Fuel: Shrubs, Trees, Grasses}

   {Fuel Condition: Dried, Wet}

   {Spread: Spreading, Not Spreading}

{Fire Spreading Rate: Very high (greater than 8m/s), High(greater than 6m/s to 8m/s), Medium (greater than 4m/s to 6m/s), Low(greater than 2 to 4m/s), Very Low(0-2m/s)}

{Fire Spreading Direction: North, East, South, West, Northeast, Northwest, Southeast, Southwest}

4. Initialize concepts relations i.e., the definition of the initial BBN links configurations.
5. Specify the Conditional Probability Table (CPT) for the SOP-updating nodes (e.g., as describe in Table 12). NETICA allows that by selecting the node's table option (i.e., right click on a node and selecting table option). This will automatically select all the dependent nodes states as the entry of the parent node. Table 12 describes an example of a CPT for "Spread" of Figure 15.

Table 12: An Example of Node CPT Entries

#	Fire	Fuel	Spread Node CPT	
			Spreading Probability	Not Spreading Probability
1	Present	Present	100%	0%
2	present	Absent	0%	100%
3	Absent	Present	0%	100%
4	Absent	Absent	0%	100%

Table 12 describes the simple forest fire spread CPT definition. That is, fire spreads only if fuel is present (Breejen et al., 1998b; Peter Hirschberger, 2016b). Thus, the probability values of the CPT are responsible for defining the situation of the nodes.

4.7 Initialising Priors of the Phenomena State in BBN

Every mission is assumed to start at the initial state level (i.e., as initialised by the SMEs or equal values for nodes in a state of ignorance). The agents sensor information will be used in updating the states nodes. Therefore, the priors used in the BBN can be initialised in two ways. One involves a simple cold-start, and the other requires elicitation from SMEs values. This chapter considers the former, and the latter is explored in Chapter 5.

For the cold-start, probabilities are initialised simply as *1/n* x *100%*, where *n* is the number of states for a given node. Thus, in the absence of any information, each node state, e.g., the 'Fire' node of Figure 15 has a starting probability of ½ x 100% = 50%. Assume that a UAV carrying an infrared sensor detects fire at a location. In this case, the probability of 'fire = present' increases (and the probability of 'fire=absent' correspondingly decreases). To derive the probability of each node from SME's judgement, the chapter proposes the use of Thurstone's paired comparison (Allen, 1994). This should be an activity performed when SOP is written rather than a precursor to each situation. In this approach, an SME is presented with pairs of element states in a BBN and asked to grade the relative criticality of each state. There will be n(n-1)/2 pairs for a given set of states, where *n* is the number of states. The main limitation is that the number of pairs can quickly become exhausting, so it might be appropriate to decompose the BBN into sections and have groups of SMEs score small sections based on the specific SME experience, which can then be integrated.

To illustrate this method, Table 13 shows the SMEs' frequency for the ‘Fuel type’ node of Figure 15 evaluated by 10 SMEs respondents with experience in forest fire physical operations[8]. The number of pairs to compare is 6. Each SME has presented with a single pair at a time and asked to indicate which pair is more critical (by assigning 1 to this type).

Table 13: Example of SMEs Paired Comparison Frequency Table

	shrubs	non-combustibles	grasses	trees
shrubs		10	8	2
non-combustibles	0		1	0
grasses	2	9		2
trees	8	10	8	

Once a count of values has been made, this is converted to a proportion of the number of respondents described in Table 14. For example, shrubs and non-combustibles proportion values are 1 (i.e., 10/10), and shrubs and grasses will be 0.8 (8/10). Note that entries across the diagonal of Table 14 are replaced with 0.5 because BBN does not allow a cycle. The mean for each column is calculated, e.g., the ‘shrubs’ column = 0.5+0.5+0.2+0.8 / 4 = 0.5, and the corresponding z-score obtained[9], i.e., 0.69. The z-score is then normalised to 1 to define the final weighted value as described in Table 15 (i.e., suits the CPTs probability values). In Table 15, the elements have been re-ordered to reflect increasing SME values.

[8] As derived from the physical forest fire experiment conducted by the author in Nigeria (the experiment result was discussed in Chapter 6).

[9] This can be done from a Standard Normal Distribution Table (z-table) or uses NORM.DIST in Microsoft Excel (with the values of a From population mean, standard deviation and cumulative being 0,1,1 and x being the calculated value of mean in table IV).

Table 14: Example of SMEs Paired Comparison Frequency Table

	Shrubs	Non-combustibles	Grasses	Trees	Mean across rows
shrubs	0.5	1.0	0.8	0.2	0.625
non-combustibles	0	0.5	1.0	0	0.375
grasses	0.2	0.9	0.5	0.2	0.45
trees	0.8	1.0	0.8	0.5	0.775

Table 15:Frequencies Mean and Z-Score

	Non-combustibles	Grasses	Shrubs	Trees
Z	.8	.71	.69	.63
Mean	.85	.55	.5	.35
Weight	-.05	.16	.19	.28

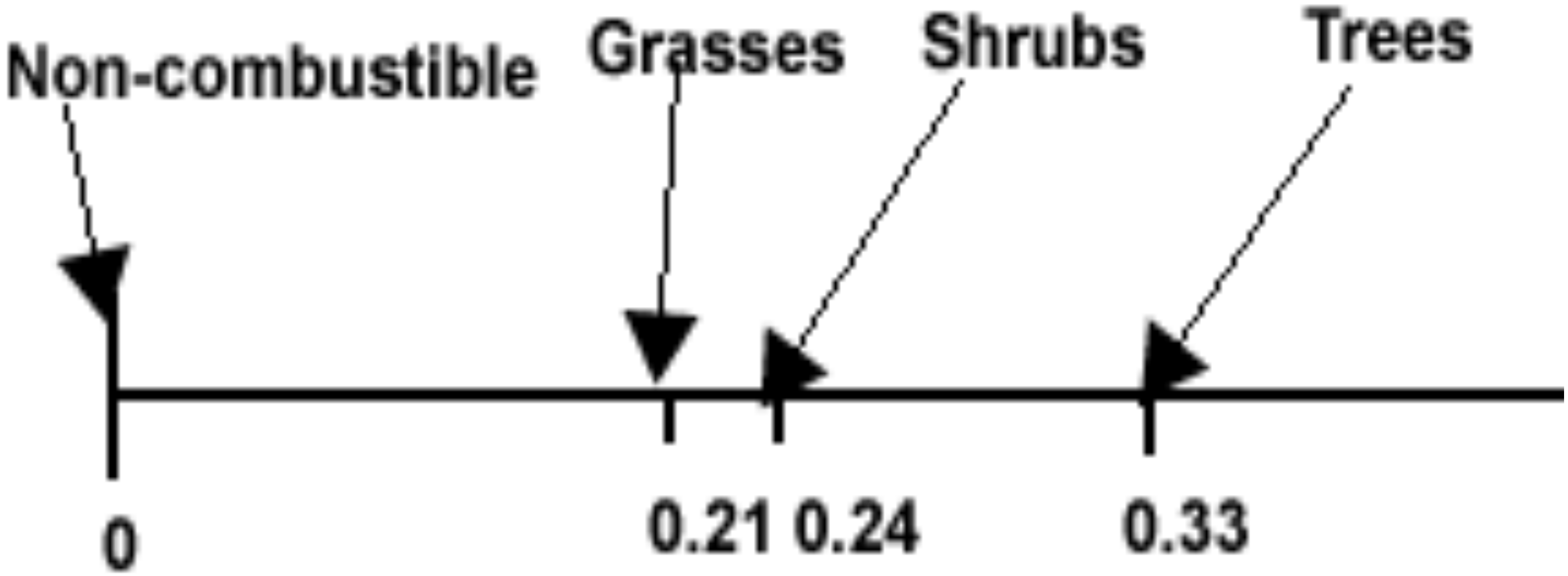


Figure 16: Final Weighted Values Number Line

From Figure 16, the trees got a higher weight. This shows that having trees as fuel type, fire will spread faster than having other fuel types, e.g., grasses, shrubs, and non-combustibles. Similarly, non-combustible got the lowest weight; this means the fire will not spread given the non-combustible fuel type.

Therefore, Thurstone's paired comparison method serves as a way of merging different SMEs' contributions to define nodes' states probabilities. The outcome can be applied to update the BBN through the CPTs as described in Table 12.

4.8 Updating Phenomena States Priors of the BBN

Having established a BBN for the search area, the next step is to apply this to a mission. The thesis suggests a simple algorithm that manages the change in state probabilities due to incoming agents' information. To illustrate this, assume a UAV has a temperature sensor that defines fire presence (as 1 for 'present' or 0 if the information does not exceed the temperature threshold for 'present,' i.e., absent). On initiation, the UAV will have a belief, B_1, for its 'fire' element. The initial information state, I_1, has $B_1 = 0.5$. Assume that the sensor reports new information to update its belief, B_{new}. Assume that on the first report, the sensor indicates fire present. From this, p(fire = present) increases from 0.5 to 0.75, i.e., p(fire = present) = B_1 x I_1 + 1 / I_2 (I_1 + 1) = 0.5 x 1+1 / 1x(1+1) = 1.5 / 2 = 0.75. On the second report, its sensors make another positive report. In this case, p(fire = present) increases; (0.75 x 2 +1) / 3 = 0.83. Note that this gradual update is for unreliable sensors. Reliable information needs to be updated directly to a value of 1. For example, if a fire is detected from a reliable source (say, an experienced human lookout ranger or a reliable sensor), the belief will rapidly increase to 1. The sensor reliability weights can be assigned using Thurstone's paired comparison method described in Section 4.7. An intensifying factor, k, serving as a measure of reliability (derived from the SMEs' judgement) can be applied. Thus, the detected states' belief (probability value) increases while the non-occurring states decrease (e.g., Figure 17). This allows active DSA modelling within the system (a higher probability identifies, i.e., winning states and as such, the current situation perception).

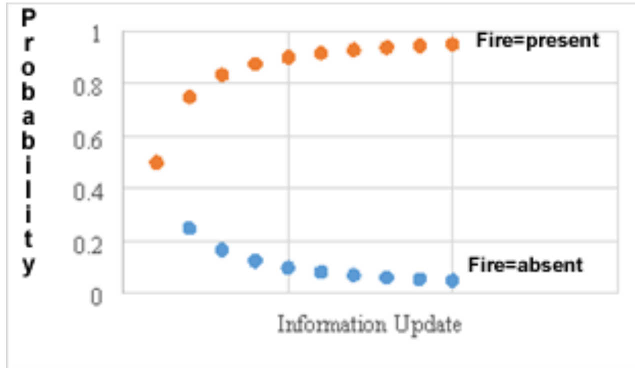


Figure 17: Example of Belief Update Transition.

Table 16 demonstrates an example of how the "Wind Speed" node of Figure 15 changes its probability values over various sensor updates. This begins with no report at epoch 0 so each state {very low, low, medium, high, very high} is equally probable i.e., $P(S_i) = 100\%/n$ *i.e.,* $100\%/5$. At each epoch, new information is received (indicated by the grey cells background and bold value in Table 16). From the formula presented, the probability at #1 is calculated as *p*(WindSpeed = Low) = (0.2x1+1) / 2 = 0.6 and the probability of any other value for windspeed will be (1 - 0.6)/ 4 = .1. Likewise, at #2 *p*(WindSpeed = High) = (0.1x2+1) / 3 = 0.4 (windspeed High will update to 0.4, while windspeed= Low will be 0.4,i.e., 0.6x2/3), and the other non-occurring states will be 0.066, i.e., .1x2/3. The process continue at every sensor value reception.

Table 16: Estimated Probabilities for Wind Speed Data

#	Very Low	Low	Medium	High	Very High
0	.20	.20	.20	.20	.20

1	.10	**.60**	.10	.10	.10
2	.067	.40	.067	**.40**	.066
3	.05	.30	**.30**	.30	.05
4	.04	**.44**	.24	.24	.04

4.9 Nodes Relations Measures

The structure of the BBN describes the causal relations among different nodes. This allows the ability to measure the connections among BBN nodes using conditional independence measures. That is the depiction of "who affects who the most?" among the nodes of the DSA system. Depicting the relations have many benefits to the DSA system. For instance, this can be used as the basis for the network structural learning (as described in Chapter 8) and the identification of the most critical nodes of the system. Measures such as entropy, Pearson's correlation, and probability variance can be used (Neapolitan, 1990; Pearl, 1978). For example, Equation 7 measures the likelihood of S_s situation increment/decrement given findings at related entries R_r, with respective states s,r ϵ [1, N].

$$\lambda(S_s \mid R_r) = \sum_{r=1}^{r=n} \sum_{s=1}^{s=n} P(S_s, R_r)[\, P(S_s|R_r) - P(S_s)]^2$$

Equation 7:Nodes Relation

where Ss is the querying situation node with several states s, Rr is the related nodes with their number of states r.

The degree of relevance $\lambda(S_s \mid R_r)$ is the likelihood of change in the probability of querying node S_s given finding in a related node R_r (Equation 8). Values range from 0 to 1, with 0 meaning the lowest degree of relations and 1 means a high degree of connections. For example, assume the degree of the relation of the 'Fire' node with respect 'Location' node at when the initial update values are {fire present:0.75, location_Cell_2:0.625}, then $\lambda(Fire_s \mid Location_r)$ is :

$$\lambda(Fire_s \mid Location_r) = \sum_{r=1}^{r=n} \sum_{s=1}^{s=n} P(Fire_s, Location_r)[\, P(Fire_s|Location_r) - P(Fire_s)]^2$$

Equation 8: BBN Nodes Relevance Computation

Expanding this, gives:

$$\lambda(Fire_s \mid Location_r) = \sum_{r=1}^{r=n}(P(Fire = Present,\; Location_r)[P(Fire = Present|Location) - P(Fire = Present)]^2) + P(Fire = Absent, Location_r)[P(Fire = Absent|Location) - P(Fire = Absent)]^2$$

$$\lambda(Fire_s \mid Location_r) = P(Fire = Present, Location = C1)[P(Fire = Present|Location = C1) - P(Fire = Present)]^2 + P(Fire = Absent, Location = C1)[P(Fire = Absent|Location = C1) - P(Fire = Absent)]^2 +$$

$$P(Fire = Present, Location = C2)[P(Fire = Present|Location = C2) - P(Fire = Present)]^2 + P(Fire = Absent, Location = C2)[P(Fire = Absent|Location = C2) - P(Fire = Absent)]^2 +$$

$P(Fire = Present, Location = C3)[P(Fire = Present|Location = C3) - P(Fire = Present)]^2 + P(Fire = Absent, Location = C3)[P(Fire = Absent|Location = C3) - P(Fire = Absent)]^2$

+

$P(Fire = Present, Location = C4)[P(Fire = Present|Location = C4) - P(Fire = Present)]^2 + P(Fire = Absent, Location = C4)[P(Fire = Absent|Location = C4) - P(Fire = Absent)]^2$

From Table 16 #1, the probability values are:

P(Fire = Present) = 0.75, P(Fire = Absent) =0.25,

P (Fire=Present, Location = C1) = 0.75x0.125 =0.0937,

P (Fire=Present, Location = C2) = 0.75x0.0.625 = 0.468

P (Fire=Present, Location = C3) = 0.75x0.125 =0.0937

P (Fire=Present, Location = C4) = 0.75x0.125 =0.0937

P(Fire=Absent, Location = C1) = 0.25x0.125 = 0.0312

P(Fire=Absent, Location = C2) = 0.25x0.625 = 0.156.

P(Fire=Absent, Location = C3) = 0.25x0.125 = 0.0312

P(Fire=Absent, Location = C4) = 0.25x0.125 = 0.0312

From Bayes rule,

P(Fire =Present|Location =C1) = P(Fire=Present, Location=C1)/P(Location=C1) = 0.0937/0.125 = 0.0117.

P(Fire =Present|Location =C2) = P(Fire=Present, Location=C2)/P(Location=C2) = 0.468/0.625 = 0.748.

P(Fire =Present|Location =C3) = P(Fire=Present, Location=C3)/P(Location=C3) = 0.0937/0.125 = 0.0117.

P(Fire =Present|Location =C4) = P(Fire=Present, Location=C4)/P(Location=C4) = 0.0937/0.125 = 0.0117.

P(Fire =Absent|Location =C1) = P(Fire=Absent, Location=C1)/P(Location=C1) = 0.0312/0.125 = 0.249 .

P(Fire = Absent |Location =C2) = P(Fire= Absent, Location=C2)/P(Location=C2) = 0.156/0.625 = 0.249

P(Fire =Absent|Location =C3) = P(Fire=Absent, Location=C3)/P(Location=C3) = 0.0312/0.125 = 0.249 .

P(Fire = Absent |Location =C3) = P(Fire= Absent, Location=C3)/P(Location=C3) = 0.156/0.625 = 0.249

P(Fire = Absent |Location =C4) = P(Fire= Absent, Location=C4)/P(Location=C4) = 0.156/0.625 = 0.249

Now, substituting the values:

$\lambda(\text{Fire}_s \mid \text{Location}_r) = 0.0937(0.0117-0.75)^2 + 0.0312(0.249-0.25)^2 + 0.468(0.748-0.75)^2 + 0.156(0.249-0.25)^2 + 0.0937(0.0117-0.750)^2 + 0.0312(0.249-0.25)^2 + 0.0937(0.0117-0.750)^2 + 0.0312(0.249-0.25)^2$

$\lambda(\text{Fire}_s \mid \text{Location}_r) = (0.0937 \times 0.545 + 0.0312 \times 0) + (0.468 \times 0 + 0.156 \times 0) + (0.0937 \times 0.545 + 0.0312 \times 0) + (0.0937 \times 0.545 + 0.0312 \times 0)$

$\lambda(\text{Fire}_s \mid \text{Location}_r) = 0.15$ (15%).

From this, the chances of change in the belief of ‘Fire’ node given any information on location is 15%. Thus, the measure of relevance depicts the degree of relevance among different concepts.

4.10 Discussion and Conclusion

In this Chapter, the concept of how sensor information from distributed agents can be presented and managed using BBN to reflect DSA has been described. The aim is to illustrate how the proposed BBN could reflect DSA at a system level. The use of BBN for DSA offers the following beneficial features in contrast to the existing concept maps, propositional networks, fuzzy logic, and ontologies:

i. Belief quantification: BBN quantifies belief using probabilities to present phenomena within the DSA system. Increasing or reducing probabilities based on the agents' perceived sensor information makes the BBN adaptable to dynamic situations. Section 4.8 describes the update methods and algorithm. This is very different from the concept maps, propositional networks, or ontologies (Nguyen et al., 2019; Salmon et al., 2009; Stanton et al., 2006; Stefanidi et al., 2022; Stewart et al., 2008; Suhail et al., 2022). In those approaches, belief occurrence is Boolean, i.e., either occurred or not and is presented textually. Overall, the concept of belief measurements paves the way for controlling agents' activities transition using probabilities and sensor reliability allocation. For example, Bayesian learning can be applied to support prediction and uncertainty handling (as described in Chapter 7), agents interaction analysis (Chapter 5), and adaptable knowledge prediction (Chapter 8), which makes it distinguishable from other methods.

ii. Projection of future Situation: the BBN can predict situations using Bayes conditional probability rule, expectation-maximisation, or gradient descent algorithms (Bottou, 2010; Dempster et al., 1977; Lee et al., 2007; Mandt and Hoffman, 2017; Romanycia, 2019). This supports Endsley’s Situation Awareness projection stage by estimating the future most likely values of the Search Area Phenomena. A detailed description of the process has been described in Chapter 7.

iii. Uncertainty handling: uncertainties due to sensor faults (soft findings) can be handled by adjusting the probabilities of the uncertain states. Again, learning algorithms such as the Bayes rule, expectation-maximisation, gradient descent, etc., can be applied to estimate missing values (e.g., due to hardware/software issues, etc.). The metric evaluation has been conducted in Chapter 7.

iv. Adaptability: learning in dynamic environments requires a flexible BBN configuration in both parametric (value-based) and structural updates based on the conditional independence measures (node relations) of Section 4.9. The algorithm and methods described in Section 4.8 and Section allow flexible presentation of various situations based on the received information. Again, exceptional cases can be identified by human SMEs and treated accordingly. For instance, from Figure 15., a fire could not spread when absent. Therefore, the joint probability p(Fire:Absent, Spread:Spreading) = 0. Normal joint probabilities can be derived simply as the product of their independent priors, e.g., for the update at #2 of Table 12 (assumed spread node value is "spreading"), p(fire:present, spread:spreading) = 0.75x0.75 $\approx$0.56. In terms of causal relations, we can say that from the BBN in Figure 15 the link between 'Fire' and 'Spread' nodes could exist only if 'Fire=Present'. That is, fire will spread only if it occurs. Similarly, Chapter 8 describes adaptability of BBN in terms of the BBN structural learning.

v. Infomration reusability: knowledge can be reused within the BBN models. For example, exchanging information between two picture compilers is possible by revising the assigned SMEs weights to conform to the receiver's goal. To illustrate this, assume a fire detecting PC exchanges fire information with evacuation PC, the recipient will then revise the SMEs weight of the fire node to be in line with its mission goal and consider it for its future activities.

vi. Multiple situations state presentation: as described in Figure 15 all possible states of a situation can be presented using a single BBN node. The winning states differ only by having higher probabilities. This is very different in the case of propositional networks, concept maps, or ontology, which involves reconstructing a new model.

vii. Agents heterogeneity handling: both automation and SMEs agents have a unique way of presenting their information to reflect SA within the BBN. For example, Section 4.7 and Section 4.8 describe how SMEs can assign probability weights to states of the BBN and how UAVs information can be transformed to the BBN and reflect the system SA.

The above features demonstrate the potential of applying BBN toward good DSA modelling in a team of agents with varying capabilities. The use of BBN shows a certain level of superiority above the existing concept maps, propositional networks, fuzzy logic, ontology, etc., in terms of the above-outlined features. For example, prediction and uncertainty handling (SA projection) can be handled using Bayes rule or learning algorithms such as the expectation-maximisation algorithm, gradient descent algorithm, etc. (Bottou, 2010; Dempster et al., 1977; Pearl, 1988; Romanycia, 2019) as described in Chapter 7. Summarily, the use of the Bayesian Belief Network to describe Distributed Situation Awareness in the team of agents provides viable solutions to the quantification of beliefs, uncertainty handling, prediction, and adaptability better than the outlined existing methods. More results to prove this claim was developed in Chapters 5, 7, and 8.

5 Chapter 5: Agents DSA Comprehension and Interaction

The agents' resources utilisations aspect of the search problem is modelled as a Distributed Constraint Optimisation Problem (DCOP) in Chapter 3; in this Chapter, I extend the formalisation of DCOP to the Distributed Situation Awareness (DSA) by considering the SA modelling discussed in Chapter 4. The aim is to take a step towards merging the concept of DSA modelling and area coverage(search) problems by considering agents' interactions and the applied BBN of Chapter 4.

5.1 Introduction

As discussed in Chapter 4, DSA agents have different views or perceived information based on their goal of which the combination of agents' current beliefs and understanding defines Situation Awareness at the system levels (Salmon et al., 2009, 2008, 2006; Stanton et al., 2006, 2009). This can happen only through agent interactions, i.e., information exchange (Stanton et al., 2006, 2009; Stanton, 2016). The interaction here is referred to as the simple agents exchanging information with the PCs or host. For instance, if the temperature sensor detects fire using temperature value threshold (e.g., when the temperature rises above 89^{o}C, this indicates fire presence) while the visual camera and infrared sensor carrying agents use colours detection; these information needs to be contextually interpreted by the corresponding Picture Compiler (PC) to know which sensor to trust in a particular situation. This interaction process needs to be monitored to avoid unnecessary interactions, infinite message exchange, or deadlock. Remember, the decision time for most DSA systems in a dynamic search area is critical. For example, fire occurrence information needs to be interpreted within 5minutes of the detection (Ingle et al., 2011). Thus, achieving DSA effectively with minimal resources could help in tackling the time/space constraints. Thus, the aim of this chapter is to address the following research questions: "how can we ensure a good interaction among DSA agents?" and "how do agents interactions support both search plan and SA management?" as a subquestions to the main question "*RQ3. How could agents' search plan support SA management?*" of Chapter 1. Good interaction here means a resource utilised interaction and is modelled as a Distributed Constraint Optimisation (DCOP) Problem (i.e., similar to Chapter 3).

DCOP refers to the assignment of variables to optimise (minimise/maximise) parameters' cost functions by considering the imposed constraints (Fioretto et al., 2018; Hoang et al., 2017, 2016). For example, considering the use-case of multi-UAV mission for forest fire monitoring in Chapter 3, the UAVs are tasked to explore an environment to monitor the fires. Therefore, the DCOP refer to the assignment of waypoints, i.e., places to visit (waypoints), to minimise/maximise the UAV/mission cost functions, e.g., UAVs battery life (minimise), computational demand (minimise), mission coverage (maximise), redundant search (minimise), etc., under the imposed constraints as outlined in Chapter 1. This can only be achieved through good interaction among agents.

DCOP exists in many forms such as the classical, dynamic (i.e., with changing cost functions and variables), multi-objective (i.e., DCOP with different cost value target optimisation, different minimisation and maximisation cost values), etc., (Fioretto et al., 2018, 2017, 2015; Hoang et al., 2017, 2016). The choice of the application domain determines the best form to model the problem. Most real-world applications in dynamic environments utilises the proactive dynamic DCOP (PD-DCOP)(Hoang et al., 2017, 2016, Hoang, 2019). PD-DCOP is a form of DCOP in which the agents are subjected to solving a series of DCOP problems in changing forms and dynamic search area (Hoang, 2019; Hoang et al., 2017). Although PD-DCOP algorithms respond to changes within the agents' search area, it fails to handle the analysis of agents' interactions (Fioretto et al., 2018; Hoang et al., 2016), SMEs contribution integrations, success measurement, and SA modelling. Therefore, these issues will be addressed in this chapter.

5.1.1 DCOP Agent in DSA System

The role of an agent in DCOP is to select a variable in a cost-effective manner despite the imposed constraints (outlined in Chapter 1). To do this effectively, the agent needs to understand its current situation (e.g., based on the sensor information and location) and act accordingly. This conforms to DSA perception (through sensors), comprehension (through the logical organisation of the perceived information as discussed in Chapters 1 and 2), and prediction (forecast of future plausible states and actions). Additionally, for adequate DSA formalisation with DCOP, agents' roles, cognitive abilities, constraints, and resource parameters must be considered. This chapter maintains the agents' roles and constraints configuration discussed in Chapter 1, Section 1.2. Thus, DCOP in the DSA system is role and goal-based in a dynamic, uncertain (because agents' information could be unreliable due to sensor, hardware, software, and the search area dynamism issues), and multi-objective form. For example, understanding forest fire spread for the simple agents means acting on the sensed information only, e.g., by generating specific waypoints around the fire detection area to get the rough shape of the fire. This is very different at the PCs' level due to the ability to join information from different simple UAVs. Thus, it involves the logical organisation and interpretation of information submitted by other simple agents, e.g., fuel type, weather report, etc. Therefore, the DCOP optimisation in DSA system is based on the agent's situation, role, and mission goal.

5.1.2 Dynamism and Multi-objectivity of DCOP in DSA System

This Chapter adopts the definition of dynamic DCOP problem as a finite horizon, proactive, dynamic, and multi-objective PMO-DCOP (as in Chapter 3). As such, every set of agent situations has a particular set of optimising parameters and their respective target optimisation (i.e., either minimisation or maximisation), as outlined in Chapter 3, Table 5. For example, coverage needs to be maximised during the agents' searching task by minimising the redundant search. Similarly, coverage will not be the targeted optimising parameter during the fire mapping task. Therefore, finding a DCOP solution in a DSA system within the limited time/space frame signifies that the problem is finite-horizon (time-based) and in multi-objective form within a time limit t. DCOP finite horizon problem needs to be solved within the assigned time limit (Hoang et al., 2017), which conforms to the limited time/space demand for decision-making while managing SA. Therefore, in most DSA systems, finite horizon DCOPs will be more rampant than infinite ones.

In terms of DCOP proactivity (prediction of future states using previous mission data) and reactivity (acting solely on current information), the DSA system is mostly proactive for the thesis problem. However, the problem could be reactive sometimes if the variables are rapidly changing in such a way that the history of perceptions has no significant effect on the current situation perception. For example, considering the case of generating a searching waypoint randomly using the Lévy distribution (Chawla and Duhan, 2018) as discussed in Chapter 3, previously generated variables have little or zero impact on future variables, i.e., based on the results in Table

7, Table 8, Table 9, and Table 11 of Chapter 3). In this situation, agents' previous information will not help in predicting plausible future situations; as such, the version of this DCOP is reactive.

5.1.3 The Model

To capture the dynamism in DCOP and DSA, I modelled the problem as a finite horizon proactive, dynamic, uncertain, and multi-objective DCOP (PUDM-DCOP) similar to the description in Chapter 3. The slight difference is that this chapter considers measuring each of the agents' interactions as a way of calculating the overall resources utilisation. In contrast, Chapter 3 focuses on resources utilisation only.

$$D = \{A_i, P, T, V, \Upsilon_i, S_{condition}, W, K_i, \alpha_i, \lambda, \delta, \beta, C, I\}$$

Equation 9:Agents Interaction DCOP Model

where,

$A_{ij} = \{a_{11}, a_{22,} a_{33}, \dots, a_{ij}\}$ is the set of agents *i* of type *j* , $i \in [1,M]$, $j \in [1,N]$ e.g, a UAV of fire detecting type.

P is the set of agents' (P_A) and mission's (P_m) parameters (i.e., from Chapter 3 Table 5), and their target cost optimisation function C_o.

T is the mission time space $T_i = \{t_1, t_2, t_3, \dots, t_n\}$, n = 1, 2, 3, … n (e.g., t_1 = 5 minutes, t_2 = 10 minutes, etc.).

V is the set of variables i.e., $V=\{v_1, v_2, v_3, \dots, v_n\}$. I assume V is the set of searching waypoints for the agents for the chosen use case.

δ is the situation's probability priors measure, such that $\delta: \Upsilon_i \rightarrow \mathbb{R}$. The value of δ is going to be updated at every agent sensor poll.

Υ_i is the set of agent's states over time period T, defined by the Markov chain $\Upsilon_i = \{\Upsilon_1 \times \Upsilon_2 \times \Upsilon_3 \times \dots \times \Upsilon_n\}$. The prediction of agent's states from time t, can be defined by the Markov transition of probabilities $P(\Upsilon_t = a \mid \Upsilon_{t-i}, \dots, \Upsilon_{t=2,} \Upsilon_{t=1,} \Upsilon_{t=0}) = p(\Upsilon_{t-i}, \dots, \Upsilon_{t=2,} \Upsilon_{t=1,} \Upsilon_{t=0} \mid \Upsilon_t$

$= a)$ x $p(\Upsilon_t = a)/p(\Upsilon_{t-n},..., \Upsilon_{t=2}, \Upsilon_{t=1}, \Upsilon_{t=0})$. In other words, current situation of the agent is determined by its previous transition measured using conditional probabilities (i.e., to comply with probabilistic measures δ).

$S_{condition}$ is the set of dynamic environmental conditions that trigger target cost function switching based on the agent current information described by the tuple $S_{condition}=\{\Upsilon_i, S_v, t_i\}$, where S_v is the set of search area's dynamic variables (i.e., wind speed, wind direction, fuel type, fuel condition, and terrain nature). The essence of the search area condition is to describe the changing nature of the operating environment (environmental dynamism). For instance, when an agent detects a fire, the searching pattern and cost optimisation parameters need to be updated to comply with the current detection.

W is the finite set of search plan waypoints to be explored, i.e., $W=\{w_1,w_2,w_3,...,w_n\}$. Thus, $W \in S_i$, where S_i is the search space.

K_i is the finite set of constraints $K_i=\{k_1,k_2,k_3,...,k_n\}$ imposed on the agents, e.g., limited energy, limited interaction, limited sensor range, uncertain target destination, and communication range.

α_i is the set of action spaces across agents, such that $\alpha_i = \{\alpha_1 \times \alpha_2 \times \alpha_3 \times \dots \times \alpha_i\}$ is factored across each agent. That is the set of agents' actions given a particular state. For example, if a fire is spotted by a simple UAV (i.e., a state), the action could make a circular pattern to understand its shape. The actions are defined jointly by the agent's current state and constraints, i.e., α_i: Υ_i x k. Other agents, e.g., the PCs, have various actions given their respective situations.

β is the probability of selecting a successful variable at time t_{n+1} given situation and variable transitions at time t_1, t_2, ..., t_n. i.e., $\beta = P(V_{i+1}| V_1,V_2, V_3,...V_i) = P(V_1,V_2, V_3,...V_i| V_{i+1})P(V_{i+1})/P(V_1,V_2, V_3,...V_i)$. Therefore, β can be used to measure uncertainty in forecasting future variables assignment. The higher the probability of selecting variable V_i at time t+1, the higher the chances of optimising the cost functions. The priors of the variables (situations) will be derived from previous entries.

λ is the variable assignment function defined by $\lambda: V \rightarrow A_i$.

C is the real-valued cost function defined jointly by agents' actions in every situation and cost implication, i.e., C: α_i x Υ_i x P$\rightarrow\mathbb{R}^+$ (as described in Chapter 3). Every agent's action in a particular situation is mapped to a positive real number cost value measured using SMEs allocation. For example, a waypoint space distance within 1Km^2, 2Km^2, and 3Km^2 cost 30, 20, and 10 cost values in optimising coverage, computational power, memory, etc.

I is the agent's interactions (data exchange through communication), such that the interaction between agent a_i and the tuple defines a_{i+1} I = {a_i, a_{i+1}, k_c, M_{ij}}, where k_c is the communication range constraint, and M_{ij} is the required information for interacting agents i and j. Note that, I ∈ K_i.

Considering the thesis use case of forest fire monitoring, the agents (A_{ij}) select waypoints (V) to optimize the mission costs (C) based on the defined parameters P during their search mission. For example, redundant waypoints need to be avoided by considering the agent situation (Υ_i) derived from the sensor information (δ), location (W), and acting accordingly (e.g., generating a new waypoint based on the action, which is related to the current mission goal, α_i). The action on waypoints assignment considers the situation and assigns waypoints using lambda that optimises the cost values C. Inter-agents interactions (I) monitor the waypoints selection considering the system constraints (K). The search area situations will be predicted (e.g., when a fire will reach critical areas such as buildings) using β based on previous mission data and search area dynamic parameters $S_{condition}$. Note that β will be utilising the Bayes rule and the BBN Conditional Probability Tables (CPTs). Thus, the model describes how the agents' mission and interactions could solve the mission goal with an optimised set of resources.

5.1.4 Hypothesis

It is hypothesised that

i. an efficient interaction among agents within a DSA and DCOP system always utilises the outlined resources in Table 5 in Chapter 3. That is, it maximises the parameters that are needed to be maximised and minimises ones with minimum agent interaction.
ii. The success of agents' interaction in solving DCOP within a DSA system depends on the search plan structure.
iii. A larger number of agents interaction neither improves DCOP solution nor DSA management.

5.1.5 Metrics

The key metrics to determine the success of the proposed approach are entropy and costs (agents' and mission parameters of Chapter 3 Table 5). For entropy, the lower the entropy measure, the higher the effectiveness of the interaction.

5.2 Proposed DSA-based DCOP Solution

The main focus of the proposed solution is to describe how DCOP could be solved effectively through efficient SA modelling (Chapter 4) and agents' interactions based on the agent's search plan. Therefore, this Chapter proposed the following tools and algorithms to address the outlined challenges.

5.2.1 DSA Modelling for DCOP Solution

As mentioned earlier, DCOP in most real-world systems is a dynamic (Hoang et al., 2017), in which agents are tasked with switching between prioritising parameters. This has been described in many DCOP algorithms such as D-DCOP (dynamic DCOP algorithms), PD-DCOPs (finite-horizon dynamic DCOP), IPD-DCOPs (infinite horizon dynamic DCOP) algorithms etc., (Choxi, 2007; Fioretto et al., 2018; Fransman et al., 2019; Hoang, 2019; Hoang et al., 2017; Kluegel et al., 2017; Le et al., 2016; Yeoh et al., 2011). However, a clear model of how the agent actions solve the problem (i.e., changing cost function to be optimised based on the situation), agents interaction analysis, and the incorporation of SMEs inputs have not been described. In this chapter, the proposed BBN of Chapter 4 will be utilised to address these issues. Because agents' interactions are at the PC or host level, a bigger BBN is used to describe the solution, as illustrated in Figure 18.

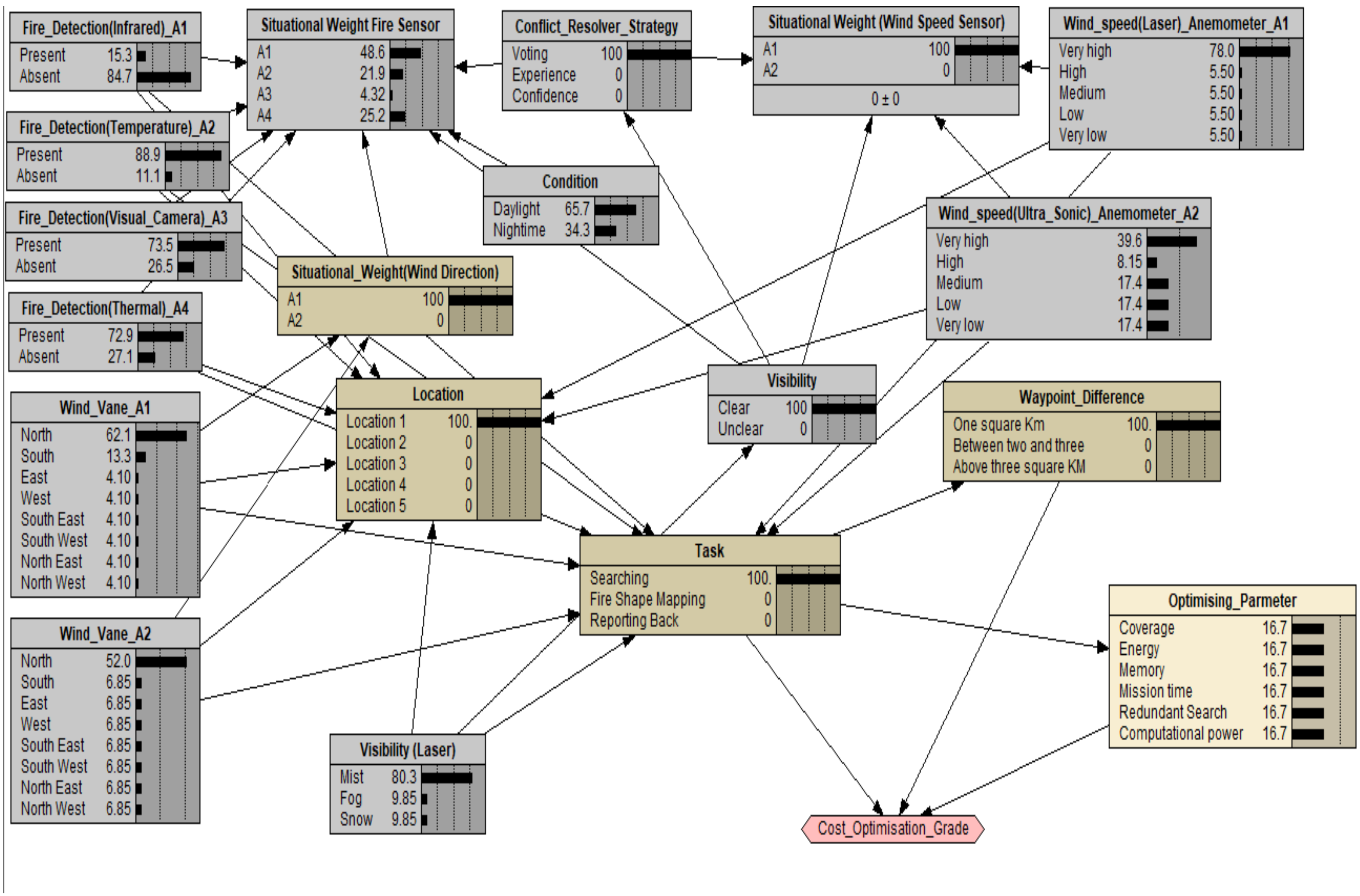

Fire_Detection(Infrared)_A1
Present 15.3
Absent 84.7
Situational Weight Fire Sensor
A1 48.6
A2 21.9
A3 4.32
A4 25.2
Conflict_Resolver_Strategy
Voting 100
Experience 0
Confidence 0
Situational Weight (Wind Speed Sensor)
A1 100
A2 0
0 ± 0
Wind_speed(Laser)_Anemometer_A1
Very high 78.0
High 5.50
Medium 5.50
Low 5.50
Very low 5.50
Fire_Detection(Temperature)_A2
Present 88.9
Absent 11.1
Condition
Daylight 65.7
Nightime 34.3
Fire_Detection(Visual_Camera)_A3
Present 73.5
Absent 26.5
Wind_speed(Ultra_Sonic)_Anemometer_A2
Very high 39.6
High 8.15
Medium 17.4
Low 17.4
Very low 17.4
Situational_Weight(Wind Direction)
A1 100
A2 0
Fire_Detection(Thermal)_A4
Present 72.9
Absent 27.1
Location
Location 1 100.
Location 2 0
Location 3 0
Location 4 0
Location 5 0
Visibility
Clear 100
Unclear 0
Waypoint_Difference
One square Km 100.
Between two and three 0
Above three square KM 0
Wind_Vane_A1
North 62.1
South 13.3
East 4.10
West 4.10
South East 4.10
South West 4.10
North East 4.10
North West 4.10
Task
Searching 100.
Fire Shape Mapping 0
Reporting Back 0
Optimising_Parmeter
Coverage 16.7
Energy 16.7
Memory 16.7
Mission time 16.7
Redundant Search 16.7
Computational power 16.7
Wind_Vane_A2
North 52.0
South 6.85
East 6.85
West 6.85
South East 6.85
South West 6.85
North East 6.85
North West 6.85
Visibility (Laser)
Mist 80.3
Fog 9.85
Snow 9.85
Cost_Optimisation_Grade

Figure 18:BBN Model for Fire Spread Concepts Presentation (NETICA Software)

Figure 18 describes a BBN for fire monitoring at the host level. The agents' perception (belief) of each concept (node) state is calculated using the prior computation algorithm described in Chapter 4, Section 4.8. The probability of the parent node states is maintained using the Conditional Probability Table (CPT). The CPT rows contain dependent nodes' states combination and their final probabilities at the parent slot, which could monitor agents' decisions on parameter optimisation and sensor conflict(condradiction) resolution in both DCOP and DSA systems. An expert can assign the CPT entries (as described in Chapter 4) or learn from the acquired mission data. For example, from Figure 18, the visual sensor's visibility is maintained using the "Visibility" node. Table 18 describes the "Visibility" node CPT entries (based on the assumption that it was received from SOP or learned from previous mission data).

Conflict among sensor information nodes can also be resolved using their parent node's CPT. Conflict resolution nodes (e.g., situation weight assigners nodes) and parameters cost nodes (e.g., the cost optimisation grade Table 17) CPTs will be updated by the SMEs as described in Chapter 4 Section 4.8. The BBN nodes were categorised into cost (utility), situation (perception), and awareness (understanding nodes) for a better DCOP and DSA formalisation.

a. Cost (utility) nodes: defines the cost implication of agents' action. For example, Table 17 describes the CPT of the utility node (Cost Optimisation Grade node) of Figure 18.

Table 17: Example of Cost Optimisation Grade Node CPT Entries

#	Agent Task	Waypoint Area Difference (% of the total searching space)	Optimising Parameter	Cost Optimisation Grade (Optimisation Weight Assigned by SMEs as describe Equation 11)
1	Searching	Between 0% and 10%	Coverage	Grade 3 (50%)
2	Searching	Between 0% and 10%	Energy	Grade 3(50%)

3	Searching	Between 0% and 10%	Memory	Grade 3(50%)
4	Searching	Between 0% and 10%	Mission Time	Grade 3(50%)
5	Searching	Between 0% and 10%	Redundant Search	Grade l(100%)
6	Searching	Between 0% and 10%	Computational Power	Grade 3 (50%)
7	Searching	Between 10% and 20%	Coverage	Grade 2(70%)
8	Searching	Between 10% and 20%	Energy	Grade 2(70%)
9	Searching	Between 10% and 20%	Memory	Grade 2(70%)
10	Searching	Between 10% and 20%	Mission Time	Grade 2(70%)
11	Searching	Between 10% and 20%	Redundant Search	Grade 2(70%)
12	Searching	Between 10% and 20%	Computational Power	Grade 3(50%)
…	…	…	…	…
90	Waypoint Generation	Above 30%	Computational Power	Grade 1(100%)

From Figure 18, the "Cost Optimisation Grade" node has three dependents' nodes (i.e., waypoint area difference, task, and optimising parameters nodes). These dependent nodes determine the success of the waypoints assignment in terms of cost utilisation (solving the DCOP problem in Equation 9). For example, Table 17 #1 indicates that during a searching task, if the waypoint area separation difference (difference between two waypoints meant for the same task) is between 0% and 10% of the searching space (i.e., calibrated using the sensor range), then coverage optimisation grade is level 3.

b. Awareness nodes: show how various nodes states could be joined to understand a particular mission concept. For example, consider the “Visibility Node” of Figure 18; its aim is to understand the clarity of the search area view using visual sensors grading. Table 18 describes an example of “Visibility Node” CPT entries.

Table 18: Example of BBN Awareness Node

#	Fog(Laser)	Mist(Laser)	Operation Time	Visibility	
				Clear	Unclear
1	Present	Present	Night Time	0%	100%
2	Present	Present	Day Time	20%	80%
3	Present	Absent	Night Time	10%	90%
4	Present	Absent	Day Time	10%	90%
5	Absent	Present	Night Time	2%	90%
6	Absent	Present	Day Time	20%	80%
7	Absent	Absent	Night Time	30%	70%
8	Absent	Absent	Day Time	100%	0%

Note that CPT entries for the awareness nodes can be obtained using SME judgment (as describe in Chapter 4 Section 4.7) or learned from previous mission data (as described in Chapter 7). This can be assigned based on the search area situation. For example, the #4 and #6 entries of Table 18 show that fog obstructs visual sensors' views better than the mist (i.e., because they have a larger probability values).

Similarly, sensor information conflict resolving nodes are a particular type of awareness nodes that consider the search area situation in prioritising sensor information. For instance, consider Figure 18, the nodes with the keywords "situational weight" in their titles handles sensor conflicts. The "situational weight fire sensor" node CPT resolves the conflict of the fire detection using an infrared sensor (fire detecting UAV with infrared sensor A1), temperature sensor (fire detecting UAV with temperature sensor A2), visual camera (fire detecting UAV visual camera sensor A3), and spectrum camera (fire detecting spectrum camera A4) in combination with "Visibility" and "Conflict resolving strategy" node. For example, if the visibility node's active state (a state with higher probability) is "clear" (e.g., during a sunny day), then the temperature sensor could be selected as having higher priority, i.e., A2 (this can be done by assigning higher probability value at the CPT entry of the sensor type). Thus, the sensor conflict will be resolved by assigning a higher probability to active states given various situations.

c. Situation node: represents the ordinary perception node belief. The probabilities of the situation node could be updated using the algorithm described in Chapter 4 Section 4.8 upon reception of agent entries.

Thus, based on the nodes categorisation, the transition of the agent's SA states of perception, comprehension, and projection state in every situation while solving the DCOP in the DSA system could be maintained using the BBN nodes and their CPTs. Specifically, DSA comprehension can be achieved by updating the BBN states priors and CPTs.

5.2.2 Variable Generation

As described in Figure 18, waypoints distance difference and current task determine both agent's and mission parameters' optimisation grade. The waypoint generation methods will still utilise the methods discussed in Chapter 3. Although the pseudo-random method performs poorly in resource utilisation and agent coordination (based on the results of Chapter 3), this Chapter applied it to showcase how dynamic situation probabilities can be treated. That is, a structured approach such as the proposed Delaunay-Inspired Multi-agent Search Strategy (DIMASS) algorithm can simply be predicted, and agents' interactions can be monitored based on its structure (i.e., because it is structured, agents' interaction can be controlled easily). Thus, the pseudorandom approach produces a dynamic behaviour, and as such, this chapter uses this approach.

5.2.3 Agents Interaction Success Measurement

Effective DSA and DCOP solutions could be achieved only through agent interaction, i.e., information exchange (Fioretto et al., 2018). Therefore, the efficacy of the interaction needs to be measured to separate between useful and useless interactions. To tackle this issue, I propose using Shannon entropy (Shannon, 1959) as applied in (Kitchin and Baber, 2017; Wiltshire et al., 2018) in combination with the proposed BBN CPTs entries. In this technique, agents' interaction on a particular task (coherence interactions) and overall information exchange (interaction) is monitored using the Shannon entropy of Equation 10.

$$E = -\sum_{i=1}^{i=n} P(S_i)\log_x (P(S_i)$$

Equation 10:Shannon's Entropy

where $P(S_i)$ is the probability measure of the success of an interaction (derived from the respective nodes' CPTs), and i is the interaction type identified using a unique identification code, e.g., i =1,2,3, …, n, and x is the logarithm base which determines the units of the entropy, e.g., if x=10, then the unit is Shannons, and bits for x=2, etc. For example, suppose three UAVs (e.g., temperature sensor, spectrum camera, and visual camera carrying UAVs) were tasked to search for forest fires in Figure 2 of Chapter 3 (i.e., the simulated search area). In that case, waypoint selection interactions among these UAVs could be assigned a unique identification, e.g., #2. Thus, the probability of code #2 P(interaction =2) will be updated at every agent's interaction using the algorithm described in Chapter 4 Section 4.8. Thus, the interaction success can be measured using Equation 11.

$$E = -\sum_{i=1}^{i=n} w_i P\left(C_{best}^{i}\right) \log_x w_i P\left(C_{best}^{i}\right)$$

Equation 11:Agents Interaction Measuring

where C_{best}^{i} is the optimisation grading probability for each interaction as described in Table 17, and w_i is the normalised (i.e., 0 to 1) weight for each optimisation grade as assigned by SMEs based on individual mission goal and parameter priority, e.g. grade 1= 1, grade 2 = 0.7, and grade 3 = 0.5. Therefore, from Equation 11 the lower the value E, the higher the optimisation during the interaction. Again, because too much agents interaction would not guarantee an effective DSA management (Foushee and Helmreich, 1988), the proposed Delaunay-Inspired Multi-agent Search Strategy (DIMASS) structured waypoints (Chapter 4) method or a number of interactions thresholds can be used to avoid deadlock and useless agents interactions while managing the system DSA during search plan generation.

5.3 The Proposed Agents Interaction Analysis Application Procedure

The following steps describe the agents' interaction measuring algorithm application procedure:

Step 1: define agents' situation-tasks transitions. This could be defined in BBN nodes CPTs, which are to be initialised by SMEs (as described in Chapter 4 Section 4.7).

Step 2: Define nodes and CPTs updates method: define how each state's probabilities could be updated either using the algorithm described in Chapter 4 or by receiving inputs from SMEs.

Step 3: Define how agents' interactions can be measured, e.g., the entropy-based approach (Equation 11).

Step 4: Given the limited time frame, generate the best possible waypoint.

Step 5: Exchange information with other agents and decide on the final best variable within the assigned time limit.

Step 6: Grade the success of the prediction strategies using the "Cost Optimisation Grade" node and Equation 11.

Step 7: Apply the CPTs entries to switch between tasks and optimise every agent's situation.

The following steps describe an example of the waypoint generation process for step 6 of the above-outlined procedures.

Algorithm 2: The Proposed DCOP Algorithm

i. Define the agent level

ii. Define the initial BBN entries

iii. For each agent's situation

Solve for $C_{best}(C, P_i, \lambda_i) = \mathrm{argmin/max}_c[\sum_{t=0}^{T} \sum_{ci}(\vec{C}_i(\lambda_i \backslash P_i))]^T$, i.e., from Chapter 3 DCOP problem formulation (Section 4.1)

iv. Generate $\vec{C} = \{ C_{best}^{t=0}, C_{best}^{t=2}, C_{best}^{t=3}, \ldots, C_{best}^{t=n}\}$, i.e., using best variables, $V = \{v_1, v_2, v_3, \ldots, v_n\}$ predictions, where n is the horizon time limit, using $p(C_{best}^{t} | C_{best}^{t=i})$ where i = t-1, t-2, t-3,..., t=0. i.e., maximising the likelihood $p(C_{best}^{t=i} | C_{best}^{t})$.

v. Interact with other agents

vi. Go to step (vii) when waypoint satisfies the cost threshold otherwise go to step iv if the time limit is yet to be reached

vii. Add variable (waypoint) to plan π if found to be the best or the number of interaction threshold has been reached.

5.4 Evaluation

The main aim of the experiment is to investigate how agents' interaction can be analysed using the mission described in Chapter 3. The use case of four UAVs tasked to conduct forest fire search (Chapter 1) was applied. The agents' task is to generate the best set of waypoints that efficiently utilise the UAV's resources (e.g., battery, memory, computation power, etc., outlined in Table 5 of Chapter

3). I assume that each agent carries a different sensor for detecting the target (fire) and waypoints' decision will happen after agents' interaction. For example, UAV1 is assumed to be mounted with a temperature sensor (i.e., detecting fire based on temperature rise, e.g., above 89^0C), UAV2 is carrying infrared sensor, UAV3 has visual camera sensor, and UAV4 carries a spectrum camera. The essence of the sensor variation (heterogeneity) is to describe the unique contribution of each agent towards DSA management based on varying search area situations (e.g., night-time, daytime, cloudy, foggy, etc.).

The experiment task is to assign non-redundant waypoints to UAVs, i.e., A1(UAV1) to A4(UAV4) of Figure 2 Chapter 3 (i.e., the thesis problem). The total size of the search area is the same as the one in Chapter 3, and the redundant search area constraint is $r_v = 10\%$ of the search area. UAV's waypoint will be accepted if it is at least 10% of the search area apart from the other UAVs' waypoints. Thus, its separation from other agents assesses the optimisation grade of the generated waypoint. For example, if the waypoint difference is between $r_v=10\%$ to $r_v=20\%$ of the search area, it is graded at level 3 (grade 3 with weight w=0.5 of Equation 11). Grade 2 and 1 has the threshold difference of $r_v=20\%$ to $r_v=30\%$ (w=0.7) and are more effective than $r_v=30\%$ (w=1 most preferable) of the search space, respectively. That is a waypoint that satisfied grade level 1 offers the most resources. From Table 17, the optimisation grade probability of waypoints from grade levels 1, 2, and 3 are 1,0.7 and 0.5, respectively (an assumed SME assigned values based on the use of probability values to grade optimisation level, i.e., higher probability values for a better optimisation). The interaction of the agents involves the exchange of the generated waypoints and a redundant waypoints check. Agents contacted in the middle of their waypoint generations were assumed to respond after waypoint generation. The number of interactions rises when the generated waypoints fail the redundant search checks. Thus, an efficient waypoint is one with a lower number of interactions and is far away from other UAVs' waypoints. Figure 19 describes an excerpts of the agents' interactions from the AMASE transcript.

```
1   Sagir Server Connected to localhost:5555
2   Redundant Search between UAV4 and UAV2
3   Time :00:02:03(123410) Rejected Waypoint has been Detected by UAV 4
4   Time :00:02:03(123410) Waypoint Generated Successfully by agent 4
5   UAV4-UAV1 score: Waypoint generated successfully generated with grade 3 score 50
6   UAV4-UAV2 score: Waypoint generated successfully generated with grade 3 score 50
7   UAV4-UAV3 score: Waypoint generated successfully generated with grade 1 score 100
8   Time :00:02:52(172169) Waypoint Generated Successfully by agent 1
9   UAV1-UAV2 score: Waypoint generated successfully generated with grade 3 score 50
10  UAV1-UAV3 score: Waypoint generated successfully generated with grade 3 score 50
11  UAV1-UAV4 score: Waypoint generated successfully generated with grade 1 score 100
12  Time :00:05:28(328439) Redundant Search between UAV4 and UAV1
13  Time :00:05:28(328439) Rejected Waypoint has been Detected by UAV 4
14  Time :00:05:28(328439) Rejected Waypoint has been Detected by UAV 4
15  Time :00:05:28(328439) Rejected Waypoint has been Detected by UAV 4
16  Time :00:05:28(328439) Waypoint Generated Successfully by agent 4
17  UAV4-UAV1 score: Waypoint generated successfully generated with grade 3 score 50
18  UAV4-UAV2 score: Waypoint generated successfully generated with grade 2 score 70
19  UAV4-UAV3 score: Waypoint generated successfully generated with grade 1 score 100
20  Redundant Search between UAV4 and UAV2
21  Time :00:09:46(586039) Rejected Waypoint has been Detected by UAV 4
22  Time :00:09:46(586039) Waypoint Generated Successfully by agent 4
23  UAV4-UAV1 score: Waypoint generated successfully generated with grade 1 score 100
24  UAV4-UAV2 score: Waypoint generated successfully generated with grade 1 score 100
25  UAV4-UAV3 score: Waypoint generated successfully generated with grade 2 score 70
26  Redundant Search between UAV2 and UAV1
27  Redundant Search between UAV2 and UAV3
28  Time :00:10:03(603109) Rejected Waypoint has been Detected by UAV 2
29  Time :00:10:03(603109) Rejected Waypoint has been Detected by UAV 2
30  Time :00:10:03(603109) Waypoint Generated Successfully by agent 2
31  UAV2-UAV1 score: Waypoint generated successfully generated with grade 3 score 50
32  UAV2-UAV3 score: Waypoint generated successfully generated with grade 1 score 100
33  UAV2-UAV4 score: Waypoint generated successfully generated with grade 1 score 100
```

Figure 19: Sample of Agents Interaction Transcripts Excerpts from AMASE

From Figure 19 #2, UAV4 share its generated waypoint with UAV1, UAV2, and UAV3, and a redundant search is detected with UAV4 and UAV2 waypoints, respectively. This results in the regeneration of a new waypoint. At #4, the regeneration process generates a waypoint that satisfies the redundant search area constraint (i.e., at least a distance of 10% of the search space away from other co-UAVs waypoints). Therefore, the number of interactions is measured to be 2 (i.e., #2 and #3). Figure 20 to Figure 27 show the UAVs' optimisation entropy and the number of interactions for each UAV across all the generated waypoints of the experiment.

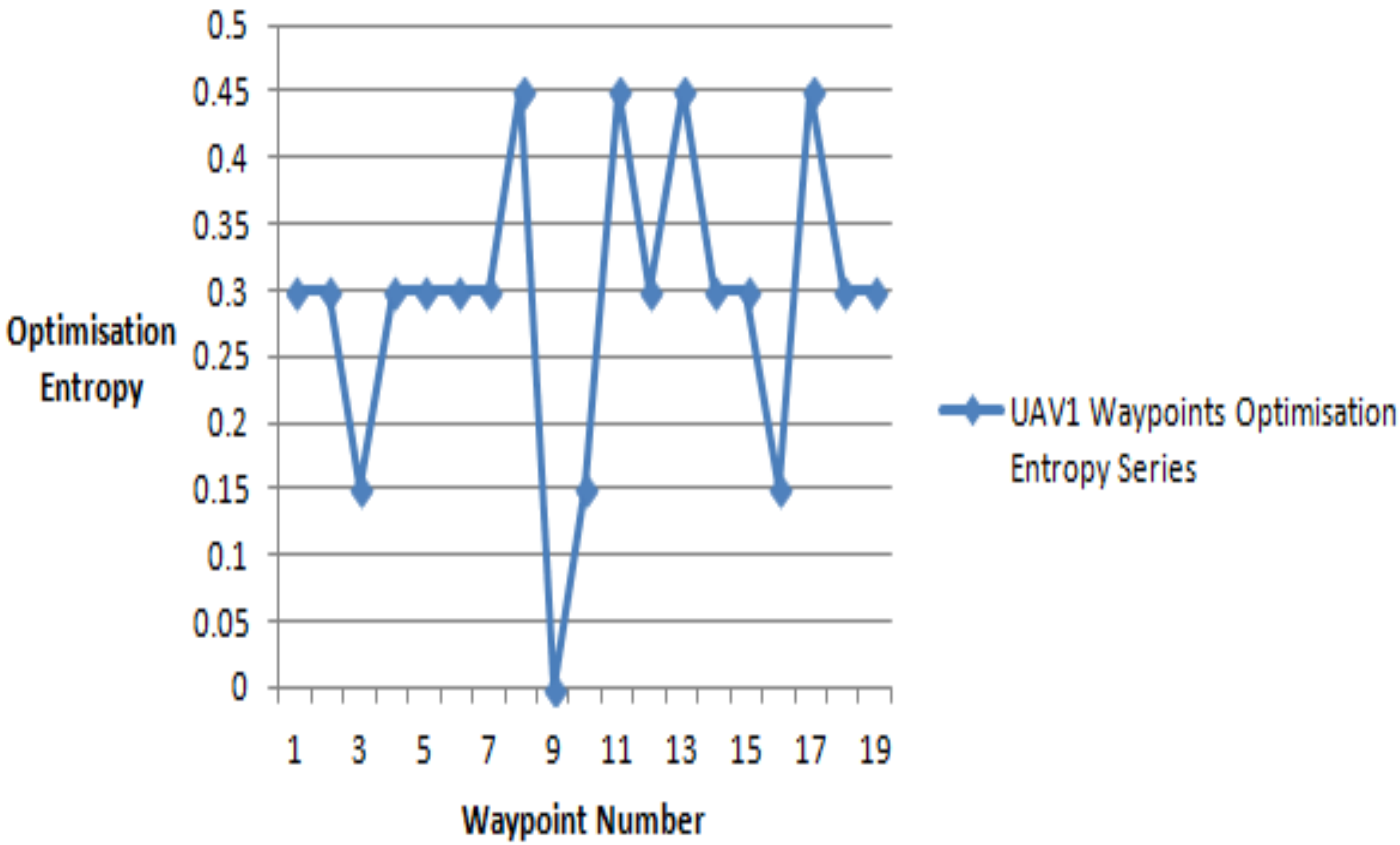


Figure 20: UAV1 Waypoints Optimisation Entropy

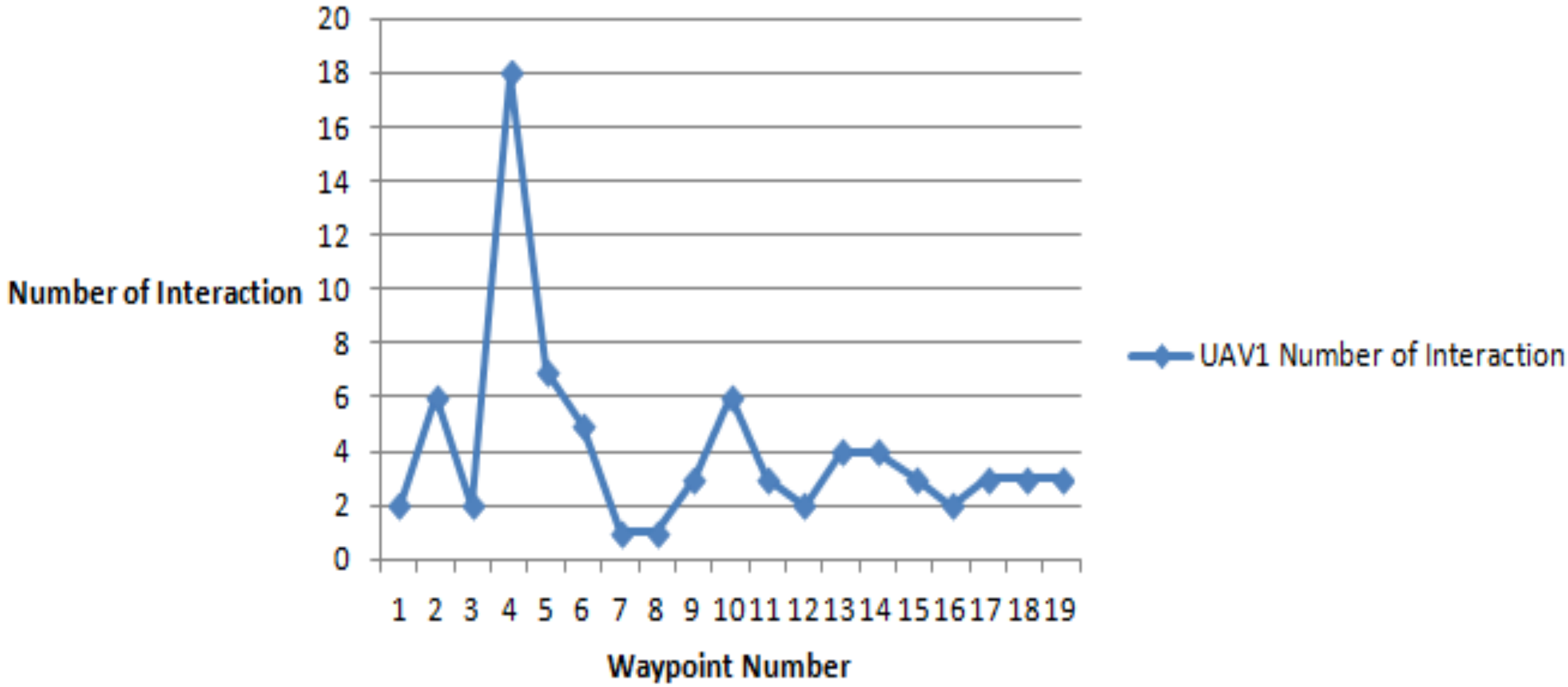


Figure 21: UAV1 Waypoints Number of Interactions

From Figure 20, the first generated waypoint (Waypoint 1) has a total of 0.3 i.e., $(0.5 \text{ x } 0.5) \log(0.5 \text{ x } 0.5)[UAV1 - UAV2] + (0.5 \text{ x } 0.5) \log(0.5 \text{ x } 0.5)[\text{UAV1} - \text{UAV3}] + (1 \text{ x } 1) \log(1 \text{ x } 1)[UAV1 - UAV4]$ as the optimisation entropy and a corresponding two (2) number of interactions from Figure 21. That is, the waypoint is accepted and has a varying optimisation entropy between the generating UAV (UAV1) and the rest of the interacting UAVs (i.e., UAV2, UAV3, and UAV4). The number of interactions reported in Figure 21 shows the total number of interactions taken to generate an accepted waypoint (which is needed to be minimised). That is, lowest entropy and number of interactions are the best combinations for a generated waypoint. As such, the best waypoint of UAV1 is waypoint 9 with 0 and 1 optimisation entropy and number of interactions respectively. The reported values in Figure 20 and Figure 21 is for the total of eighteen (18) Levy flight-based generated waypoints of UAV1 within the mission time frame T.

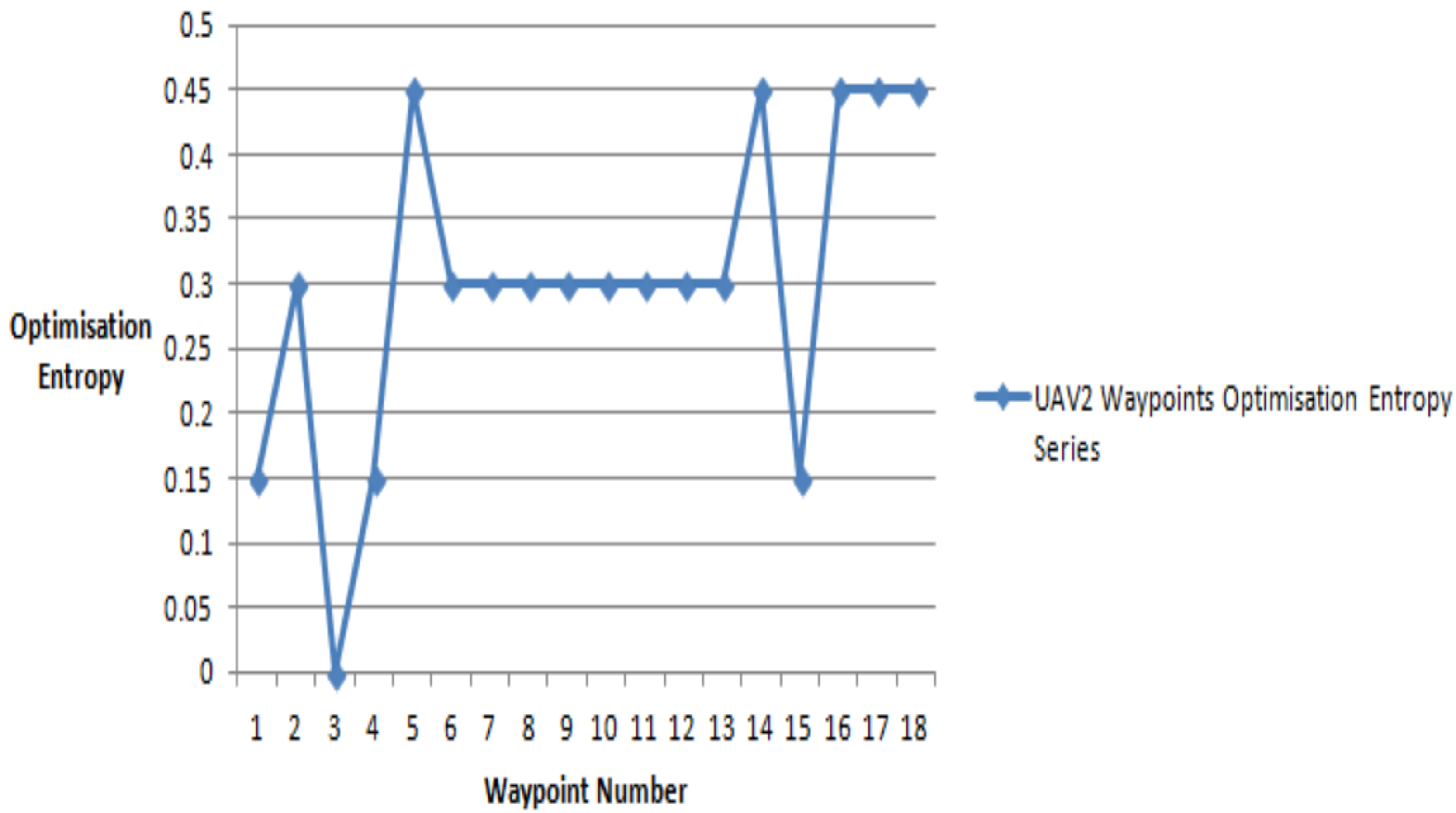


Figure 22: UAV2 Waypoints Optimisation Entropy

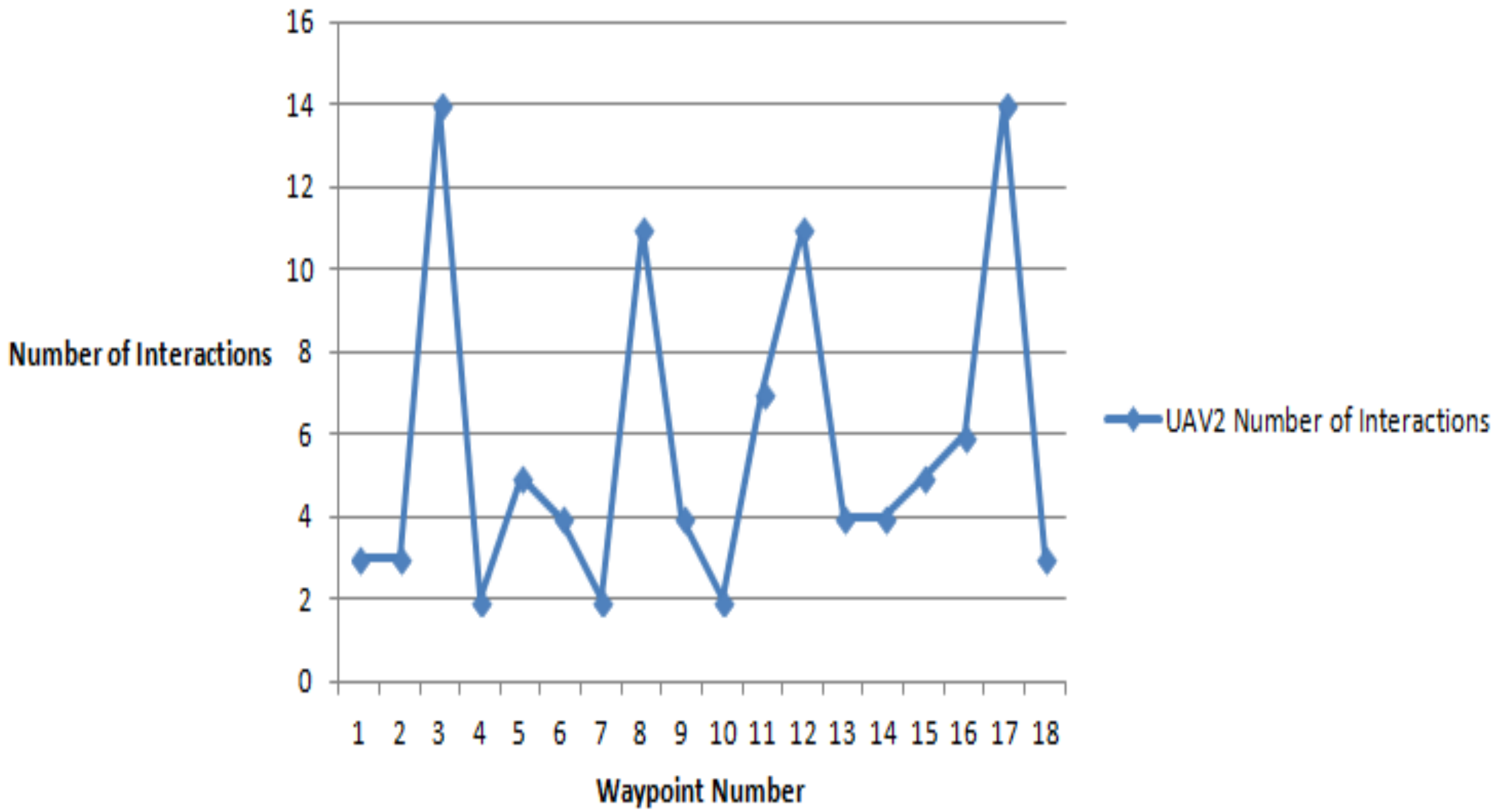


Figure 23: UAV2 Waypoints Number of Interactions

In contrast, Figure 23 and Figure 24 show the optimisation entropy and the number of interactions for UAV2 across all the eighteen (18) generated waypoints. Waypoint 3 shows a situation whereby the optimisation entropy is good (i.e., a waypoint that is far away from all other UAVs' waypoint is generated) but receives a higher number of interactions (i.e., too many agents interactions was performed before generating the waypoint).

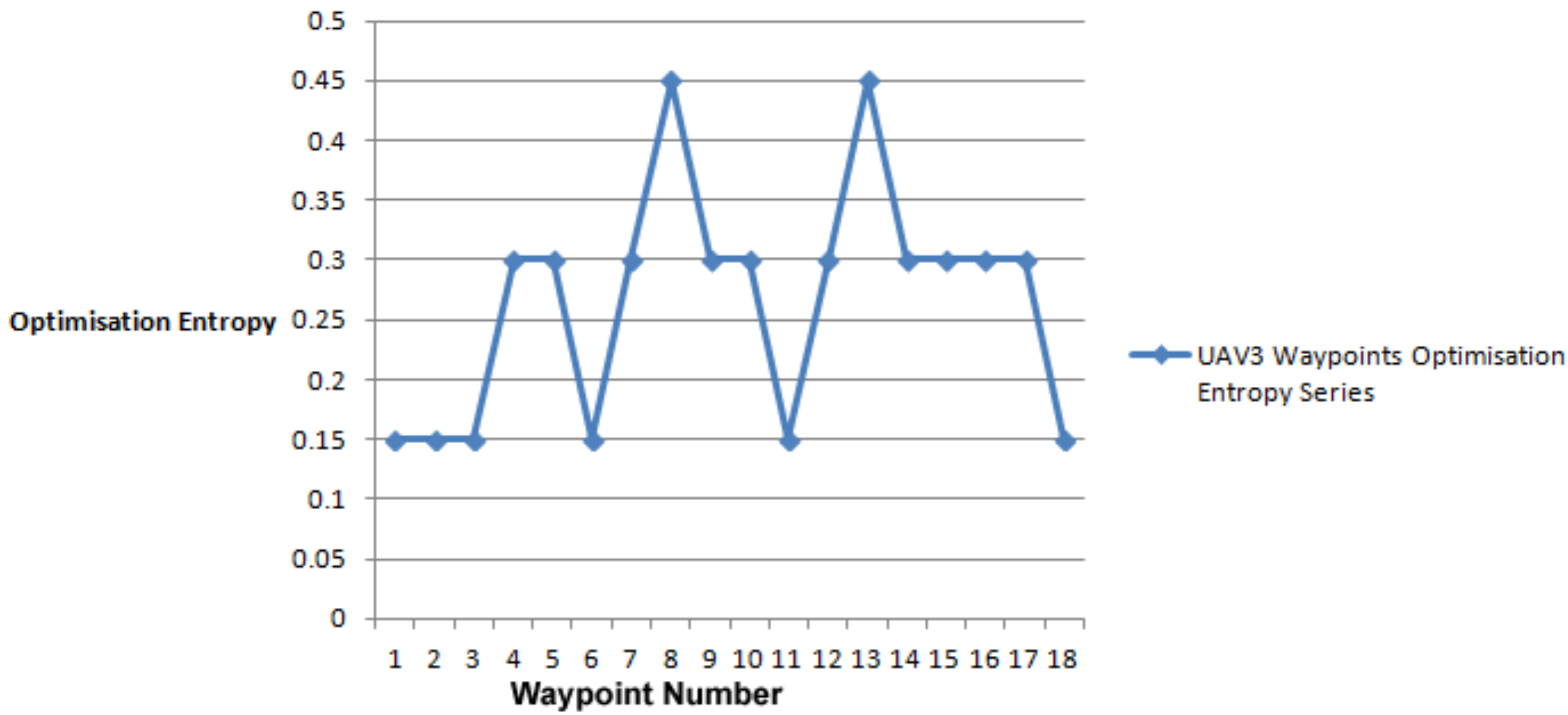


Figure 24:UAV3 Waypoints Optimisation Entropy

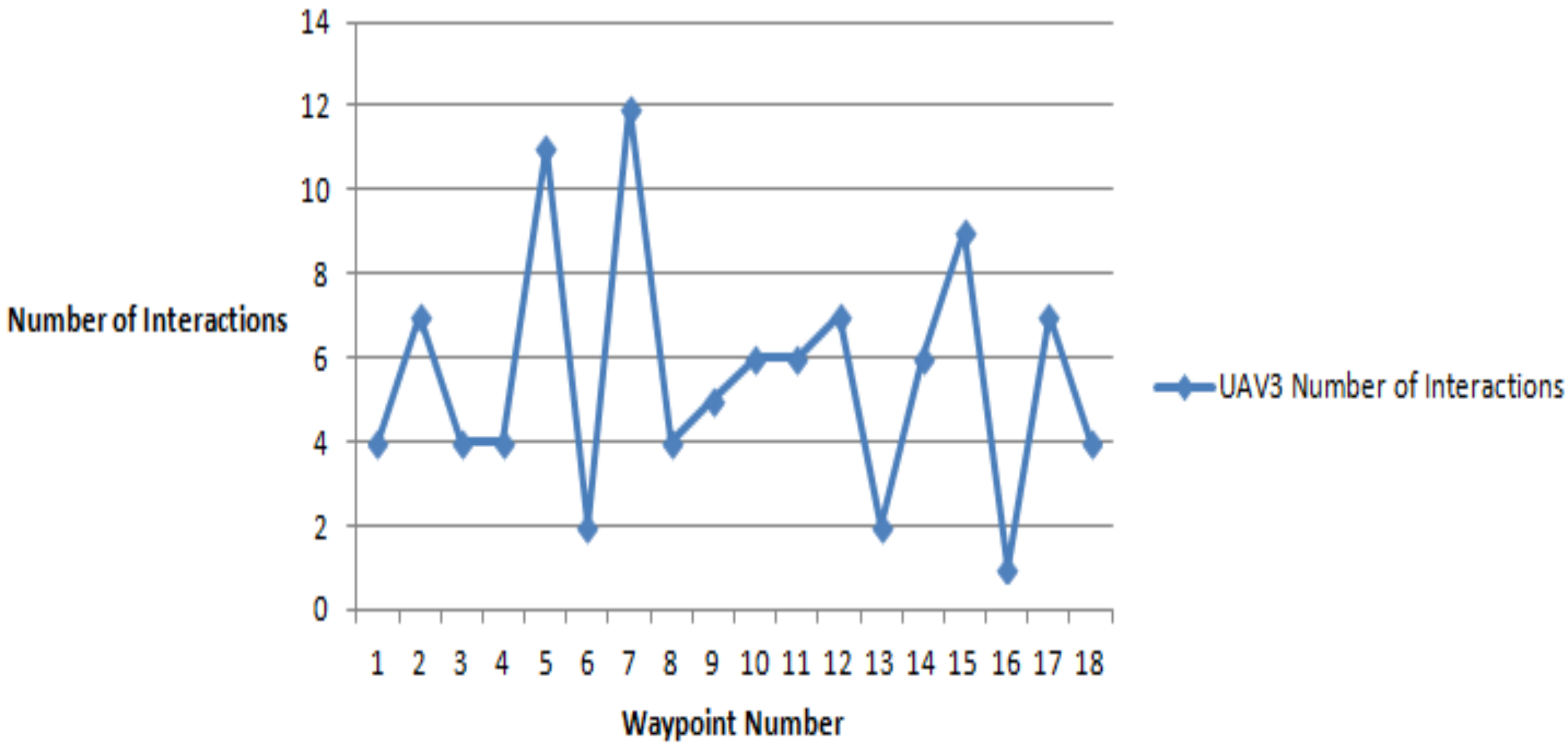


Figure 25: UAV3 Waypoints Number of Interactions

Figure 24 and Figure 25 show the UAV3's waypoint optimisation entropy and number of interactions. The optimisation entropy shows non-best values (i.e., absence of 0 value). Note that, this is as a result of the Levy flight's pseudorandom waypoint generation.

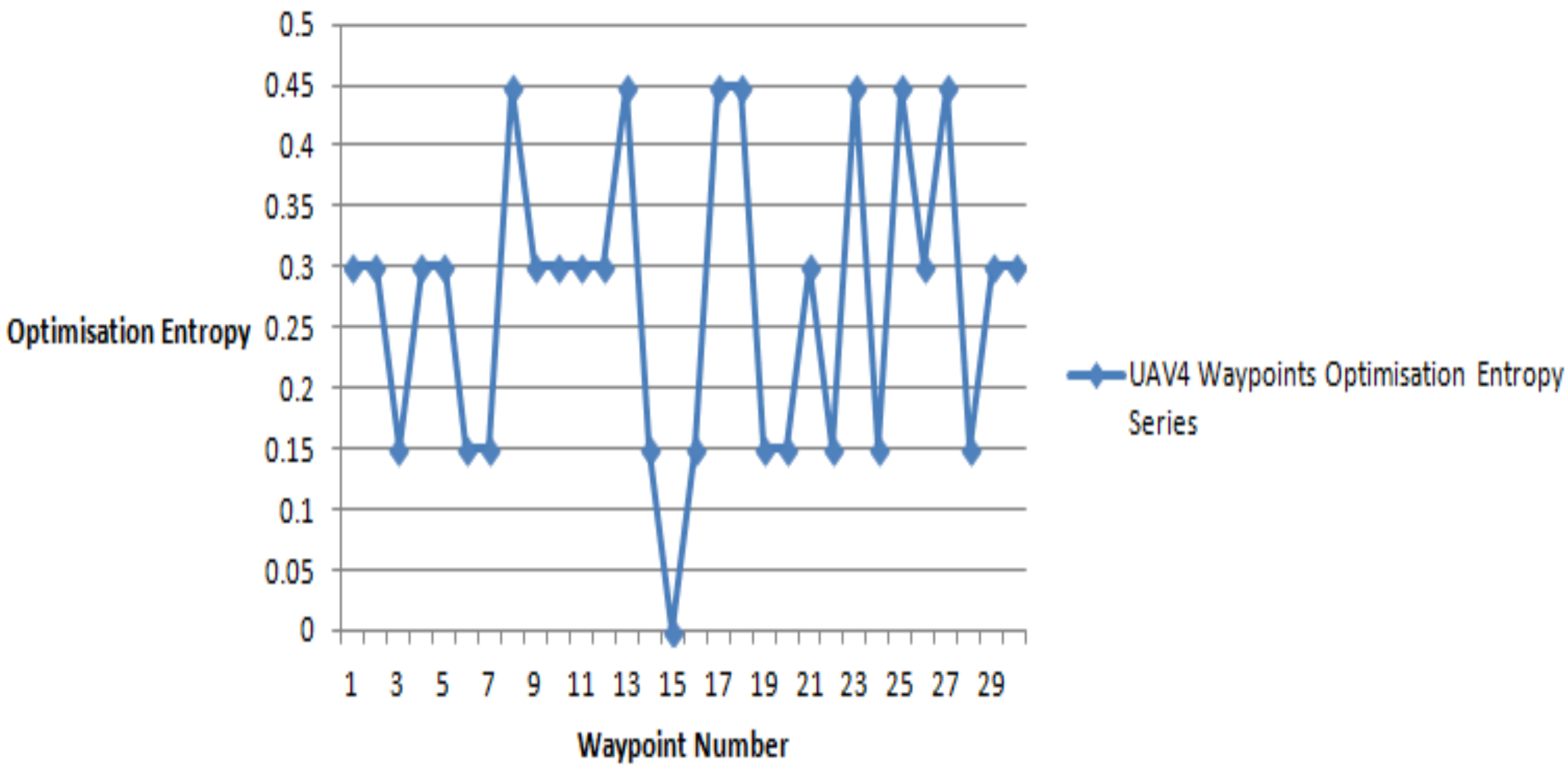


Figure 26: UAV4 Waypoints Optimisation Entropy

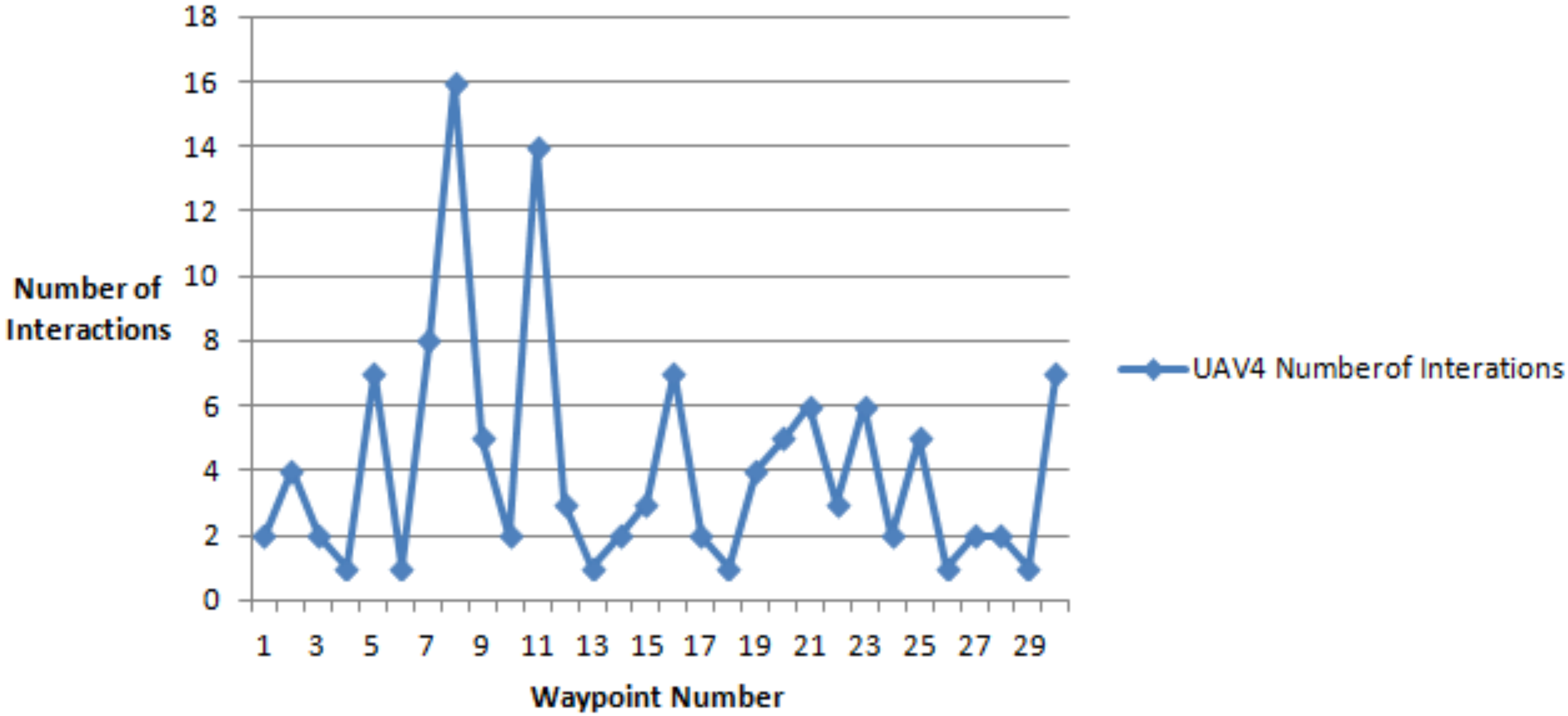


Figure 27: UAV4 Waypoints Number of Interactions

The waypoints optimisation entropy of UAV4 is reported in Figure 26. In terms of optimisation entropy, waypoint 15 has the best value whereas there are multiple waypoints in terms of number of interactions. All the waypoints of UAV4 show unique behaviour in terms of the number of agent interactions (i.e., from Figure 27).

Based on the overall agents' interactions (from Figure 20 to Figure 27), the success rate of UAV2 is lower than UAV1, UAV3, and UAV4 (i.e., by summing up the lower entropies and number of interaction values). Thus, the use of Lévy flight for the experiment demonstrates the typical operation of dynamic DCOP algorithms such as the maximum gain message (Maheswaran et al., 2004), distributed stochastic algorithm (Maheswaran et al., 2004; Verfaillie and Jussien, 2005; Wittenburg and Zhang, 2003), distributed pseudotree optimisation (Choxi, 2007; Fransman et al., 2019; Le et al., 2016), etc, and how their agents' interactions can be monitored. As mentioned earlier, the entropy and number of agent interactions for the fixed pattern approaches can be controlled by configuring the waypoints at the premission state e.g., configuring the angles, edges, and quadrants of the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) method of Chapter 3. Thus, the outcome will be less dynamic.

In terms of the number of agents interactions and optimisation entropy trade-off, this depends on the agents' resource sensitivity, space, and time constraints. For instance, a DSA system utilising agents with very limited resources (e.g., micro UAVs) needs to prioritise the optimisation entropy. Again, a system with low space/time constraints needs to prioritise number of interactions optimisation.

5.5 Discussion and Conclusion

This chapter describes how DSA comprehension and agents' interactions can be analysed using the formal properties of BBN and Shannon entropy. Figure 20 to Figure 27 illustrate the results of analysing four UAVs' interactions toward a search plan generation. The outcome shows a non-stable behaviour due to the pseudorandom waypoint generation, which demonstrates the dynamism of the system. For example, considering Figure 19, the number of waypoints and entropy measures varies at each UAV interactions. However, the fixed-pattern method was not applied because it will give the best number of interactions (i.e., minimise the number of interactions because n waypoints can be generated at once and negotiation will be at once, as proved by the result of Chapter 3). The main challenges of this approach are (i) message size is larger (i.e., the number of waypoints to be exchanged at once will be higher, unlike the Lévy flight approach) (ii) agents plan generation synchronisation needs to be managed, i.e., the matter of who generates plan first. Again, the number of agent interactions does not guarantee an effective solution especially with the pseudorandom strategies.

Generally, low entropy and the number of interactions signify good interactions and resource utilisation. Therefore, the proposed way of measuring agents' success in DCOP and DSA using BBN monitors how agents' activities optimised the outlined resources parameters within the imposed constraints. Thus, this serves as a tool for quantifying the DCOP algorithms' success especially for incomplete algorithms, such as the Maximum Gain Message, Distributed Stochastic, Max-sum, etc. (Fioretto et al., 2018; Maheswaran et al., 2004).

The results show that the number of agents interactions would not guarantee an efficient solution especially when agents generate variables randomly. Additionally, it describes how agents can make decisions, i.e., based on the BBN CPT entries. As such, a structured and non-random way of selecting variables could help to reduce the entropy values. For example, agent interactions can be structured and predicted using any structured search pattern e.g., the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) algorithm of Chapter 3. Similarly, an entropy-based method of measuring DCOP and DSA success was proposed to measure the system performance. This form of measuring DCOP success considers agent-to-agent relations and the outcome of their interactions. For example, from Figure 19 transcript, #9 to #11 shows that UAV1 optimises the best cost with UAV4 (i.e., generated the perfect waypoint), whereas relations with UAV2 and UAV3 are at the lowest cost (i.e., 50% optimisation grades).

6 Chapter 6 System Model Methods

This chapter discusses two issues (i) thesis methods, i.e., agents' simulation modelling, use case physical experiments and their results, agents data acquisition and analysis, and experiments procedure. This includes how the AMASE use case phenomena simulation got their values from physical experiments, e.g., based on fire behaviour, UAVs operation, etc. The second issue will look at (ii) the relationship between the agents' search plans in Chapter 3 and the proposed BBN as discussed in Chapters 4 and 5. The results will be investigated based on how the state's priors (derived from the agent's sensor information) could be utilised to support SA management within the system. Thus, this chapter will discuss all the thesis methods and the relationship between the search plan in Chapter 3 and the SA modelling method discussed in Chapters 4 and 5.

6.1 Introduction

Simulation methods serve as one of the easiest ways of modelling complex and most difficult systems with lesser costs and risks (Armengaud et al., 2009; F. Khan et al., 2014; Monesi et al., 2022). It is now applied in many systems models such as disaster management, aviation, military operations, robotics, etc. (Altameem and Amoon, 2010a; Armengaud et al., 2009; Gage and Murphy, 2004; Hale and Zhou, 2015; Li et al., 2015; Monesi et al., 2022; Noreen et al., 2016; Reynolds, 1987; Schwab et al., 2020; Waharte et al., 2009). While simulation offers an advantage in terms of costs and risks, it is accompanied by many limitations such as the lack of realistic parameter values, reliability issues, and prone to erroneous results especially when agents are to be operating in a dynamic and complex environment (Afzal et al., 2021; Chappell et al., 2022; Choi et al., 2021; N. v. et al., 2017). Considering the thesis use case of Chapter 1, the concept of forest fire monitoring is dangerous, expensive, and challenging to be modelled entirely using physical experiments. As such, simulation modelling was used for the use case modelling. The parameter values for the simulation model (i.e., for both the search area, UAVs, BBN, etc.) are obtained from either a physical experiment (results on this will be discussed later in this chapter) or a documented SOP material. Additionally, all the experiments, learning processes, and algorithm application procedures will be described in detail here. Therefore, this chapter focuses on the thesis methods.

Similarly, in Chapters 3 and 4, I discussed how agents' search plans can be generated and how SA can be modelled using BBN. In this chapter, a relationship on how the structure of the search plan affects the SA model (e.g., in terms of SA projection) will be discussed. That is based on whether the fixed-pattern or pseudorandom methods support the system predictability and resources management (in terms of prediction, uncertainty handling, data collection, etc.) for the proposed BBN model. Thus, the questions to be addressed in this chapter are:

i. How to effectively model a Distributed Situation Awareness system involving a team of agents?
ii. Do agents' search plan structure support SA management?

These are subquestions of the main research question: "*RQ3. How could agents' search plan support SA management?*"

The first question will be evaluated based on how easier it is to simulate the system phenomena and how realistic, efficient, and scalable the simulation's outcome is compared with a natural counterpart. The AMASE simulation framework was used based on its exceptional functionalities to model aerial robots, MVC feature, and is easy to use. Details on how it works and how to integrate the physical values were discussed in Section 6.3. For the second question, the priors values of the agent's sensor information will be used to depict the information structure and how the proposed model could easily predict the plausible future states of the priors.

6.2 Hypothesis

It is hypothesised that

i. The structure of the search plan affects the SA model of a structured search area

6.3 Methods

6.3.1 Sensor Labelling

Unique sensor labels will be recorded from the agents' mission in Chapter 3. The recording could happen on various simple agents, PCs, or host labels. The labelling procedure requires some form of textual coding, which will be mapped with the BBN states. For this thesis, the labels will be initialised at the beginning of the BBN development, i.e., from the premission planning of Chapter 4. Each sensor label for a particular node (i.e., a representative of a BBN state) must be unique. In the case of double entries from different BBN nodes (e.g., the fire detecting nodes A1-A4 of the BBN in Figure 29), the columns of the .csv storing file (memory file) will be used to identify the right node to be updated at the PCs or host level (e.g., as described in Figure 30). Each UAV sensor label will be mapped to a corresponding column (note, one node could not have two same states labels). For example, considering the BBN described in Figure 28, each UAV's sensor state has a corresponding label and node. Fire detecting UAV has a reporting value in the form of "sensor_type.state," e.g., "fire. present", etc.

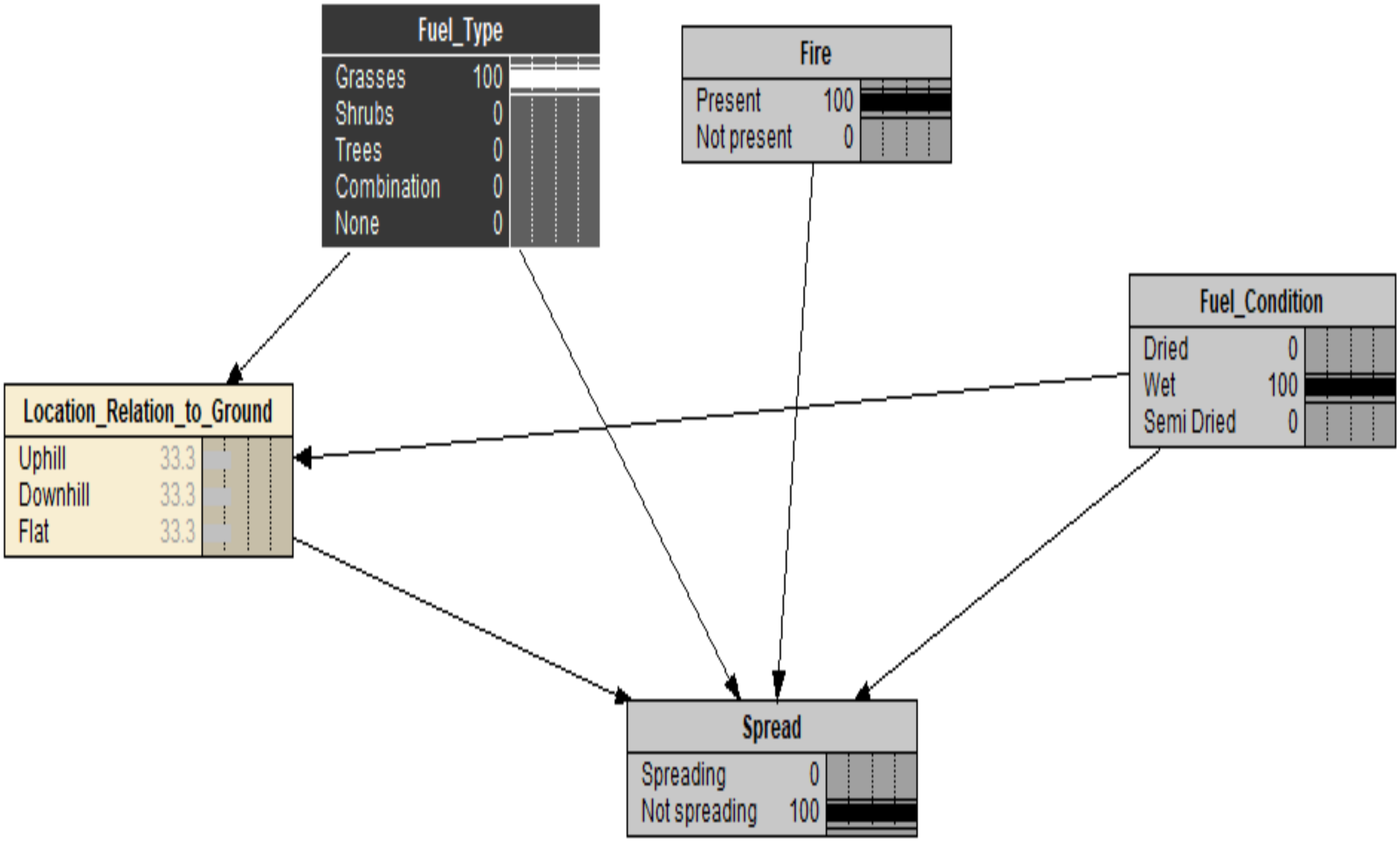


Figure 28:Example of BBN for Fire Spread BBN (PC Level)

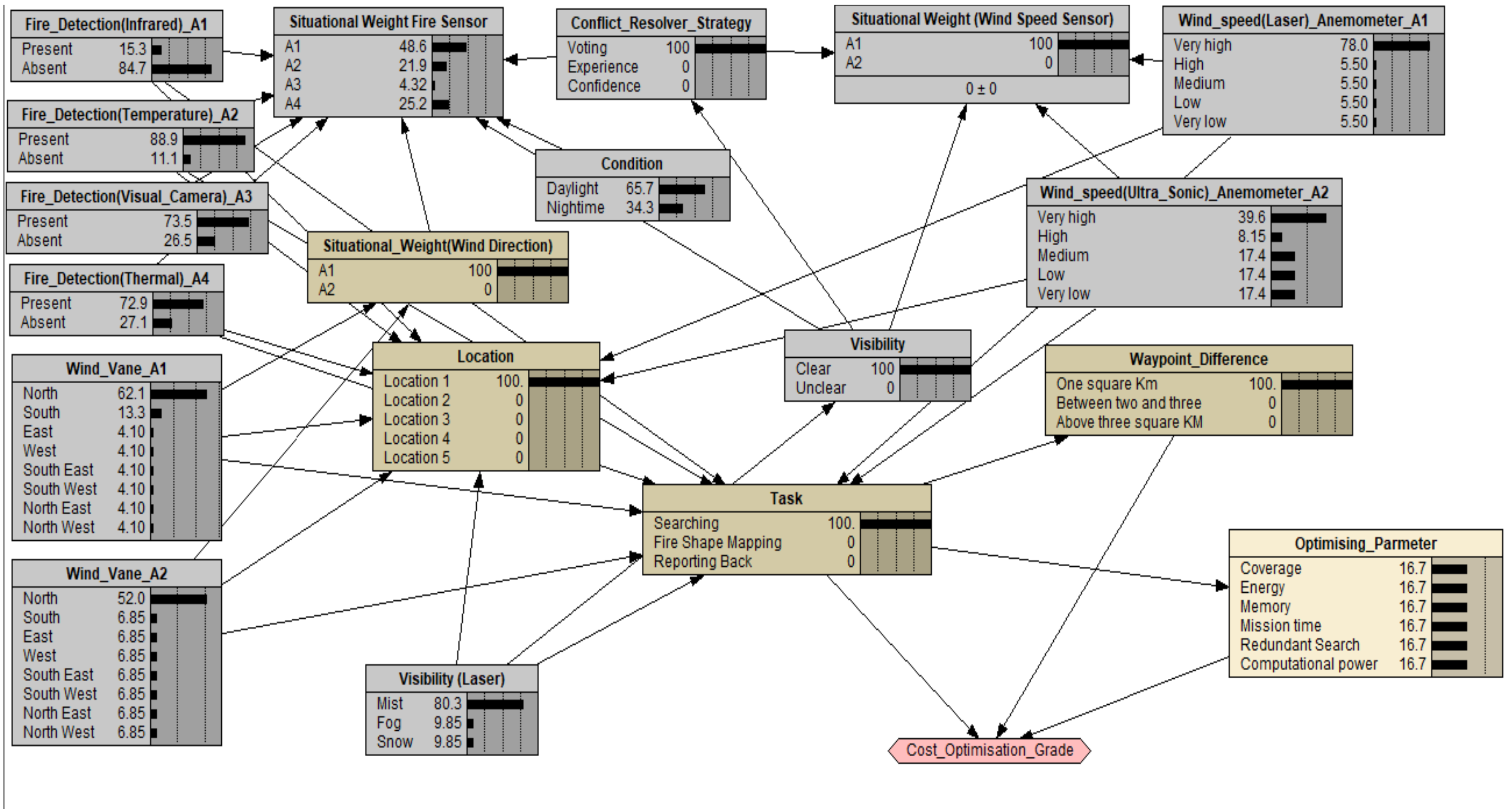


Figure 29:Example of Host BBN(Host Level)

Figure 28 and Figure 29 show two BBNs for the methods demonstration process. The BBN in Figure 28 shows a small BBN for specific concept understanding, e.g., at the PC level and the BBN in Figure 29 describes a bigger BBN with various types of nodes (as outlined in Chapter 5) for the host agent.

6.3.2 Sensor Data Recording

The sensor state can be recorded using the assigned labels from Section 6.3.1. Whenever an agent conducts a sensor sampling, it will record the sensor state, e.g., present or absent for fire detection in Figure 28 and map it to the corresponding mission time (using mission clock) and location. The recorded sensor data can be stored in many ways. The thesis uses an onboard .csv file for the sensor data, e.g., a UAV with onboard computational power, e.g., DJI matrice 100 or a mounted Raspberry pi inbuilt onboard computer. In the case of the AMASE simulation, the sensor records can be written to a .csv file on the PC or the console (i.e., followed by conversion to a .csv or .txt file). One of the key disadvantages of this approach over the latter is that the console can be reinitialised accidentally and the process resembles a fully centralised system. This thesis adopted the later approach due to its notion of distribution.

The simple agents' information exchange with the PC or host is assumed to be done wirelessly (i.e., for in mission) or manually by extracting the content of the agent's memory (post hoc). Each entry will be mapped with its corresponding location and time. For the AMASE simulation, Java classes (controller class) with file handling methods were used for the sensor data recording. For learning and analysis (as discussed in Chapter 7), each recorded sensor entry will have a unique ID (as described in Figure 30). The essence of the ID is to allow the merging of sensor information with data records. The ID can be generated from a random number generation or a serial function with or without considering the UAV's ID. For example, the *ith* character of the ID can represent the ID of the UAV, e.g., **1**003, i.e., the first character of the ID, which is 1, shows that the entry comes from UAV1, which is its entry number 003. Each round of the agent's ID generation round will be checked against a redundant value. Figure 30 describes an example of agents' recorded sensor information for the forest fire monitoring mission described in Chapters 1 and 3.

```
1   IDNUM   Fuel_Type   Fuel_Condition  Fire    Spread  Location_Relation_to_Ground Number_of_Findings
2   1615776661 Combination Dried Absent Not_spreading Flat 1
3   1362577014 Shrubs Dried Absent Not_spreading Uphill 1
4   1201345327 Shrubs Dried Absent Not_spreading Uphill 1
5   2131410112 Shrubs Dried Absent Not_spreading Uphill 1
6   1114324979 Combination Dried Absent Not_spreading Flat 1
7   1617909951 Shrubs Dried Absent Not_spreading Uphill 1
8   193252509 Combination Dried Absent Not_spreading Flat 1
9   413660474 Shrubs Dried Absent Not_spreading Uphill 1
10  1323362454 Shrubs Dried Absent Not_spreading Uphill 1
11  612471750 Shrubs Dried Absent Not_spreading Uphill 1
12  288371455 Shrubs Dried Absent Not_spreading Uphill 1
13  1847523946 Combination Dried Absent Not_spreading Flat 1
14  1947309688 Shrubs Dried Absent Not_spreading Uphill 1
15  1517076424 Shrubs Dried Absent Not_spreading Uphill 1
16  1688181995 Shrubs Dried Absent Not_spreading Uphill 1
17  1159438813 Shrubs Dried Absent Not_spreading Uphill 1
18  818295116 Combination Dried Absent Not_spreading Flat 1
19  2046166169 Shrubs Dried Absent Not_spreading Uphill 1
20  507582954 Shrubs Dried Absent Not_spreading Uphill 1
21  1580430617 Combination Dried Absent Not_spreading Flat 1
22  688971499 Shrubs Dried Absent Not_spreading Uphill 1
23  258424464 Shrubs Dried Absent Not_spreading Uphill 1
24  57945910 Combination Dried Absent Not_spreading Flat 1
25  1121861719 Shrubs Dried Absent Not_spreading Uphill 1
26  481746066 Shrubs Dried Absent Not_spreading Uphill 1
27  983394245 Shrubs Dried Absent Not_spreading Uphill 1
28  344961851 Shrubs Dried Absent Not_spreading Uphill 1
29  363633597 Shrubs Dried Absent Not_spreading Uphill 1
30  1063262355 Combination Dried Absent Not_spreading Flat 1
31  352045915 Combination Dried Absent Not_spreading Flat 1
32  2043474863 Shrubs Dried Absent Not_spreading Uphill 1
33  1770946990 Shrubs Dried Absent Not_spreading Uphill 1
34  626347287 Shrubs Dried Absent Not_spreading Uphill 1
35  897008306 Shrubs Dried Absent Not_spreading Uphill 1
```

Figure 30: BBN States Excerpts from AMASE UAVs Operation

Figure 30 shows an excerpt of the recorded sensor entries. Each column represents a node in a BBN. It is assumed (as per Chapters 1 and 4) that each situation node has a corresponding UAV responsible for submitting values for all states. For instance, an entry in #2 shows that Fuel type UAV: "Combination", Fuel Condition UAV: "Dried", and Fire UAV: "Absent" with an autogenerated ID of 1615776661.

The reported sensor record will be mapped with the current time (measured using the mission clock, e.g., the AMASE clock) and align with the corresponding location (obtained from the agent GPS or structured paths cells configuration, i.e., using speed., wind speed, wind direction, etc. reports values). For example, Figure 30 describes an excerpt of the recorded sensors data from the simple agents tasked to update the PC's BBN in Figure 28 using an update method $update(x_1, x_{2,}x_2, \dots, x_n)$ where $x_1, x_{2,}x_2, \dots, x_n$ are the passed parameters for each sensor state. For example, #3 of Figure 30 shows that the update function can be like this: *update(1615776661, Combination, Dried, Absent, Not_spreading, Flat, 1)*. Thus, the task of the update function is to write the parameter values in the corresponding .csv or .txt memory file. Note that the update function will be implemented in the controllers (e.g., Java class) and be invoked at the sensor poll based on the situation.

6.3.3 Sensor Sampling

The agent's sensor sampling can be performed in three ways: (i) constant sampling, (ii) waypoint-based sampling (i.e., individual waypoints), and (iii) periodic sampling. In constant sampling, the agent's sensor is fully activated throughout the mission. While this approach consumes agents' energy (because sensor use consumes agents' energy), it produces a more accurate presentation of the search area situation. In waypoint-based sampling, agents conduct a sensor sampling after every waypoint. This approach can be inappropriate for the agents' tasks, i.e., when the distance between the waypoint is long. For example, assume the implementation of a waypoint-based sensor sampling for the fixed-path strategies, e.g., parallel track, creeping line, Zamboni, sector search and the proposed DIMASS (as discussed in Chapter 3), the acquired sensor information could not present the situation of the search area whenever the plan has a longer-range set of waypoints.

On the other hand, waypoint-based (location-based) sampling has an advantage in terms of energy utilisation (i.e., the sensor sampling frequency is low for longer edges). This method can be good for mapping tasks. Thus, waypoint-based sampling could best fit a situation where the target is known, e.g., the known object mapping task described in Chapter 3. In periodic sampling, agents sample after every period t of their mission clock, e.g., as illustrated in Figure 30. The sampling frequency can be a fixed or varying value based on the agent's current situation γ_i . For example, during an initial search mission, the sensor sampling frequency can be fixed, e.g., after every 20 seconds. This can be reduced when a fire is detected or expected, e.g., assuming fire is expected (based on the temperature change) or detected nearby, the sampling frequency can be reduced to a lower value, say 5 seconds, depending on the speed of the UAV.

This thesis chooses a constant sensor during agents' search missions due to the dynamism of the search area, the presence of unknown targets (i.e., the location of the targets is unknown to the searching agents), and the speed of the UAVs. While the chosen sensor sampling rate could result in higher battery consumption, agents' search plan efficiency (Chapter 3) would compensate. Periodic sampling is assumed to be applied for the mapping task (when a fire is detected). Thus, the frequency of the sampling period can be initialised based on target expectation within the location (i.e., based on the previous values and other related parameters, e.g., fuel type, wind speed, wind direction, etc. for fire mapping, or SMEs assignment as described in Chapter 4 section 4.7). As such, I propose the use of Equation 12 for an adaptable sensor sampling rate based on the agent's situation.

$$\alpha = normalize(\, p+s_i+\Delta t/N)$$

Equation 12: Adaptable Sensor Sampling Computation

where p is the probability of the situation occurring given the time interval Δt, s_i is the location cell variation (i.e., a range of space that needs new sampling based on the location variation), and *n* is the number of contributing parameters.

For example, let us assume that SMEs define the probability of fire presence given a location and time to be 0.9, 200metres, and 10 minutes respectively. The location cell variation (distance threshold that necessitates sensor sampling) is 1KM. From Equation 12, we can calculate α as,

p=0.9, l=200/1000=0.2kilometres, Δt=10.

$\alpha = 0.9+0.2+10/3$

$\alpha \approx 3.7$.

Therefore, the sampling rate will be 3.7 kilometres per minute per sensor sample. This approach of sensor sampling gives an adaptable way of moderating the agents' sampling rate.

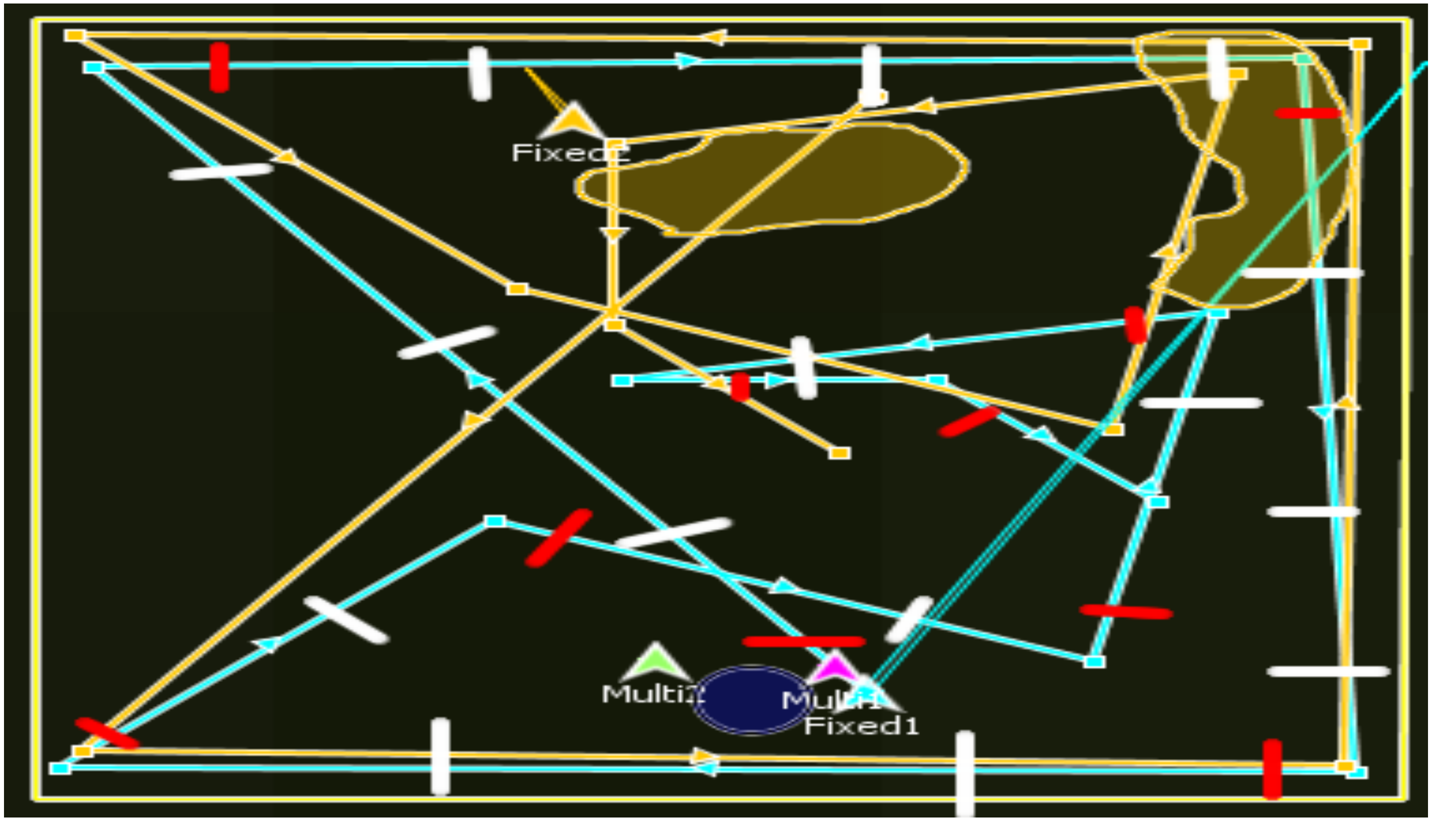


Figure 31:Example of Agents Periodic Sensor Sampling.

From Figure 31, the marked lines on the agents' paths roughly show the sensor sampling locations. This can be reconfigured based on the information or expectation of fire presence or absence as described in Equation 12.

6.3.4 Sensor Conflict (Contradictions) Handling

As described in Chapter 5, sensor conflict is inevitable due to the changing nature of the search area's phenomena over time, e.g., the distortion of visual sensors in detecting fire by fire-like objects. This can be identified from the recorded sensor data (i.e., from the .csv file) by identifying two conflicting values at the same columns, location and time frames. This can be done using an if statement to check the entries with different values across the same row (i,e., from the CSV file).

6.3.5 Featuring Engineering for the SA model

Based on the described sensor labelling above, the text sensor states would not fit directly into the BBN or the processing algorithms (as described in Chapters 7 and 8). This requires transformation into numeric forms (probabilities values). The transformation process can be achieved using the algorithm described in Chapter 4, Section 4.8. Each textual sensor outcome can be identified using if statements and transformed into a probabilistic prior value. An example of the transformation statement pseudocode can be if(value == "trees"){increment the probability of fuel type="trees"}. In the case of two states with the same text label, e.g., the value "north" for all wind direction detection UAVs represented by their respective nodes in Figure 29, this can be identified by checking the column numbers. The columns are numbered from left to right, e.g., from Figure 30, "Fuel Type" is column number 2.

Alternatively, another form of feature engineering for single-entry prediction algorithms is to change the equivalent states into a certain number, e.g., the number of each state's entries sequentially. For example, the fire node will have present = 1 and absent = 2 etc.

6.3.6 BBN Update to Support SA Model

Based on the priors update algorithm described in Chapter 4 Section 4.8, agents update their priors after every sensor poll. The update will then affect the BBN states probabilities, e.g., as described in Figure 28 and Figure 29. For example, from Figure 29, we can see that the fire occurs and is spreading fully (i.e., measured based on the probability values as described in Chapter 5). Thus, the Java controllers update the BBN states priors at every sensor update to conform to the provided sensor information. This happens through a method call from the AMASE Java controller to the NETICA API (as implemented in this thesis, codes can be found in the supplemental documents). Thus, the AMASE agents update the BBN after every sensor poll.

6.3.7 Search Area Definition

The search area comprises both static and dynamic phenomena and their parameters. Static parameters have constant values across *n* number of missions times, for example, the location of the base station is assumed to be fixed. Dynamic parameters change their values across either time or location. For example, fuel type varies across locations, whereas the wind speed changes values across periods. The change of values of the search area's fuel types across locations cell S_i is structured e.g., C1-C20 (grasses), C1-C40 (trees), C40-C60 (combination), etc., as described in Figure 32.

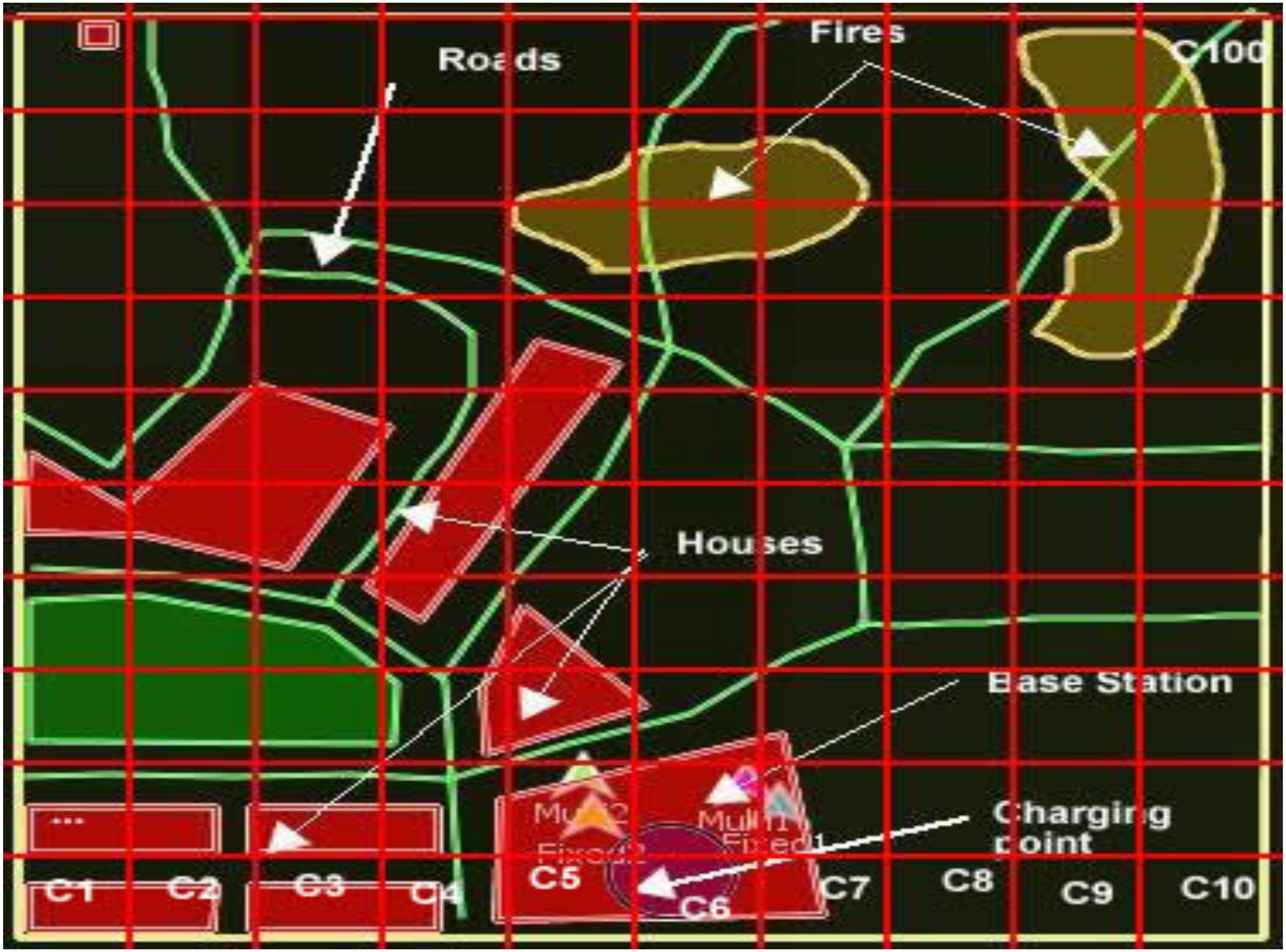

Roads
Fires
C100
Houses
Base Station
Charging point
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10

Figure 32:Example of Search Area Categorisation.

6.3.8 Priors Relation with Search Pattern Experiment

The experiment for understanding sensor information relation with search structure was implemented on the AMASE experiment described in Chapter 3. The candidate search methods selected were the best performing approaches of Chapter 3, i.e., Lévy flight and the proposed Delaunay-Inspired Multi-agent Search Strategy (DIMASS). The agents are tasked to conduct a search mission for the simulated search area under the same number of allocated mission times. The reported information is then collected and transformed into priors using the algorithms described in Chapter 4, Section 4.8. The priors are then presented visually to describe how values for different states change over time. The procedure is as follows:

i. Extract each sensor state keyword of the BBN, i.e., from the .csv or txt file
ii. Generate the priors for each state using the Algorithm described in Chapter 4, section 4.3.1.
iii. Plot the line chart of the priors using python controllers.

The outcome will show how the states' probabilities change values across time periods (i.e., various sensor samples) and which is mostly linear (for a structured search area and search plan method) or stochastic (for the dynamic and unstructured environment). The experiment uses 2000 values which allow the UAV to sample different configurations of the search area from its initial location.

6.4 Fitting Real Values to Simulation

AMASE allows the search area's simulation in a Model-View-Controller (MVC) fashion. The views elements, e.g., fires, UAVs, wind speed, etc., can be developed using XML enclosed in their corresponding tags. Each element can be referenced from its controller (i.e., Java or python classes) using a unique identification number (ID) or a name which are to be assigned by the developer. Alternatively, view elements can be controlled or instantiated using the controller functions. The second option fits updated tasks better than initialisation. For example, wind speed values can be controlled using the controller functions, e.g., randomly or time-based values change. Similarly, fire spread can be controlled using a time-based process, e.g., as described by Equation 13 using the function $l_i(x, y, h)^{t+1}$ in time difference Δt.

$$l_i(x, y, h)^{t+1} = l_i(x, y, h)^{t} \text{x}\, w(\Delta t, S_v(\vec{x}), S_v(\vec{y}), S_v(\vec{h}))$$

Equation 13: Fire Spread Modelling

where $l_i(x, y, h)^{t+1}$ is the estimated growth of the target (i.e., fire) across x,y,h axis (where h is the height) from the current position $(x, y, h)^{t+1}$, Δt is the time interval, and $w\left(\Delta t, S_v(\vec{x}), S_v(\vec{y}), S_v(\vec{h})\right)$ is the function that defines the growth factor using the search area dynamic parameters, e.g., S_v (e.g., wind speed, wind direction, fuel type, etc.) and time interval. That is, each of the functions $S_v(\vec{x}), S_v(\vec{y}), S_v(\vec{h})$ returns the velocity vector of the target (fire) mobility rate for each dimension based on the search area's assigned dynamism weight, *w*. For example, the weight *w* for dried shrubs is higher than the value for a wet one (i.e., fire spreads faster around dried shrubs fuel type than within a marshland). Similar weight assignment is performed for the influences of other variables, e.g., wind speed, wind direction, location relation to the ground, etc.

Thus, the search area's dynamic parameters, e.g., wind speed, wind direction, fire spread, etc., can be added to the AMASE framework using XML based Views (in a time-based fashion) or the Java controller. The wind speed behaviour can be extracted roughly using Google weather report or search area's weather monitoring experiment. For this thesis, a Google weather report was used for the fire spread experiments combined with direction monitoring. For example, Figure 33 shows an example of wind speed changes over time from the AMASE XML View file.

```xml
    <TranslationDirection>170.0</TranslationDirection>
  </HazardZoneChangeCommand>
  <WeatherReport Series="CMASI" Time="3000">
    <WindSpeed>0.2</WindSpeed>
    <WindDirection>90.0</WindDirection>
    <Visibility>0.0</Visibility>
    <CloudCeiling>0.0</CloudCeiling>
    <CloudCoverage>0.0</CloudCoverage>
  </WeatherReport>
  <HazardZoneChangeCommand Series="SEARCHAI" Time="3000">
    <ZoneID>4</ZoneID>
    <GrowthRate>0.1</GrowthRate>
    <TranslationRate>0.2</TranslationRate>
    <TranslationDirection>275.0</TranslationDirection>
  </HazardZoneChangeCommand>
  <HazardZoneChangeCommand Series="SEARCHAI" Time="3000">
    <ZoneID>5</ZoneID>
    <GrowthRate>0.05</GrowthRate>
    <TranslationRate>0.1</TranslationRate>
    <TranslationDirection>190.0</TranslationDirection>
  </HazardZoneChangeCommand>
  <WeatherReport Series="CMASI" Time="4000">
    <WindSpeed>0.5</WindSpeed>
    <WindDirection>45.0</WindDirection>
    <Visibility>0.0</Visibility>
    <CloudCeiling>0.0</CloudCeiling>
    <CloudCoverage>0.0</CloudCoverage>
  </WeatherReport>
  <HazardZoneChangeCommand Series="SEARCHAI" Time="4000">
    <ZoneID>4</ZoneID>
    <GrowthRate>0.4</GrowthRate>
    <TranslationRate>0.8</TranslationRate>
    <TranslationDirection>250.0</TranslationDirection>
  </HazardZoneChangeCommand>
  <HazardZoneChangeCommand Series="SEARCHAI" Time="4000">
    <ZoneID>5</ZoneID>
    <GrowthRate>0.1</GrowthRate>
```

Figure 33: Example of XML Weather Report

From Figure 33, each of the "Weather Report" elements (i.e., wind speed, wind direction, visibility, cloud ceiling, and cloud coverage) have their time values for a given mission clock in seconds, e.g., 3000 and 4000, from Figure 33. The value 3000 means the specified entries are valid for the first 3000 seconds of the mission. These values can be updated independently or jointly using Java controllers. All other dynamic parameters, e.g., fires, can be modelled similarly.

6.5 Search Area and UAVs Simulation

The thesis use case is simulated based on of a physical location and experiments. The search area consists of buildings, roads, bushes, and fires copied from a village in Gombe State of Nigeria (Chapter 3 Figure 2). The search area phenomena, e.g., fire spread rate, fuel type, wind speed, wind direction, etc., were derived from a physical forest fire experiment conducted on Liji Hill in Gombe State of Nigeria (Figure 34). The author acknowledges similar work published on the behaviour of forest fire spread in the United States, Europe, and Asia (Ingle, 2011; Merino et al., 2006; Peter Hirschberger, 2016); however, the author decided to conduct the experiment on African terrains due to omission and poor documentation in literature (Afolayan et al., 1979; International Forest Fire News, 2006; Van Wilgen, 2009). Like the rest of the world, forest fires affect Africa due to human activities such as pasture clearance, oil spillage, etc., but there is a lack of data on the effects (International Forest Fire News, 2006). For example, recently (on 10th August 2021), a forest fire killed over 65 people in Algeria[10]. Thus, the simulation selects the African terrain as a step toward describing the behaviour of forest fires in Africa. However, the outcome can be applied to other continents, e.g., the US, Canada, Turkey, etc., by varying the search area parameters from the AMASE XML code, i.e., fuel type, wind speed, wind direction, etc. Table 19 describes the experiment search area fuel configurations.

[10] Algeria forest fires: At least 65 people killed as fires spread - BBC News [WWW Document], 2021. URL https://www.bbc.co.uk/news/world-africa-58174918 (accessed 8.12.21).

Table 19: Experiment Location Fuel Type Sampling on Liji Hill, Gombe State of Nigeria

Fuel Botanical Name	Vernacular Name	Classification	Average Height (Meter)	Average Population Density (Stem per Meter Square)	Condition
Cassia occidentalis	Rai Dore or Mazamfari	Shrub	1.15	19	Mostly dried
Cynodon dactylon	Kyasuwa	Grasses	0.175	47	Dried
Azadirachta indica	Darbejiya	Trees	9	0.01	Wet
Pennisetum purpureum	Halkiya	Grasses	0.2	11	Combination of wet and dried

The fuel types were spread across the search area in Figure 2. Each location (S_i) has a particular fuel type, fuel condition, location relation to ground, etc. The fire spread values were obtained by setting a fire on the hill and monitoring the spread distance overtime period (see Figure 34) across all directions, i.e., north = 0^0, east=90^0, south = 180^0, west = 270^0, northeast >0^0 and <90^0, southeast >90^0

and $<180^{0}$, southwest $>180^{0}$ and $<270^{0}$, and northwest $>270^{0}$ and $<0^{0}(360^{0})$ as defined by the American Practical Navigator (Bowditch, 2002). All permission and safety measures were observed before starting the fire. The experiment utilises the traditional method of bush clearance conducted at the beginning of every dry season to allow easy access to the stone quarrying area. The experiment was conducted on a sunny day with an average wind speed, wind direction, and humidity values of 8KMPH, $34^{o}C$, and 12%, respectively. Figure 34 describes one of the experiment's scenes.

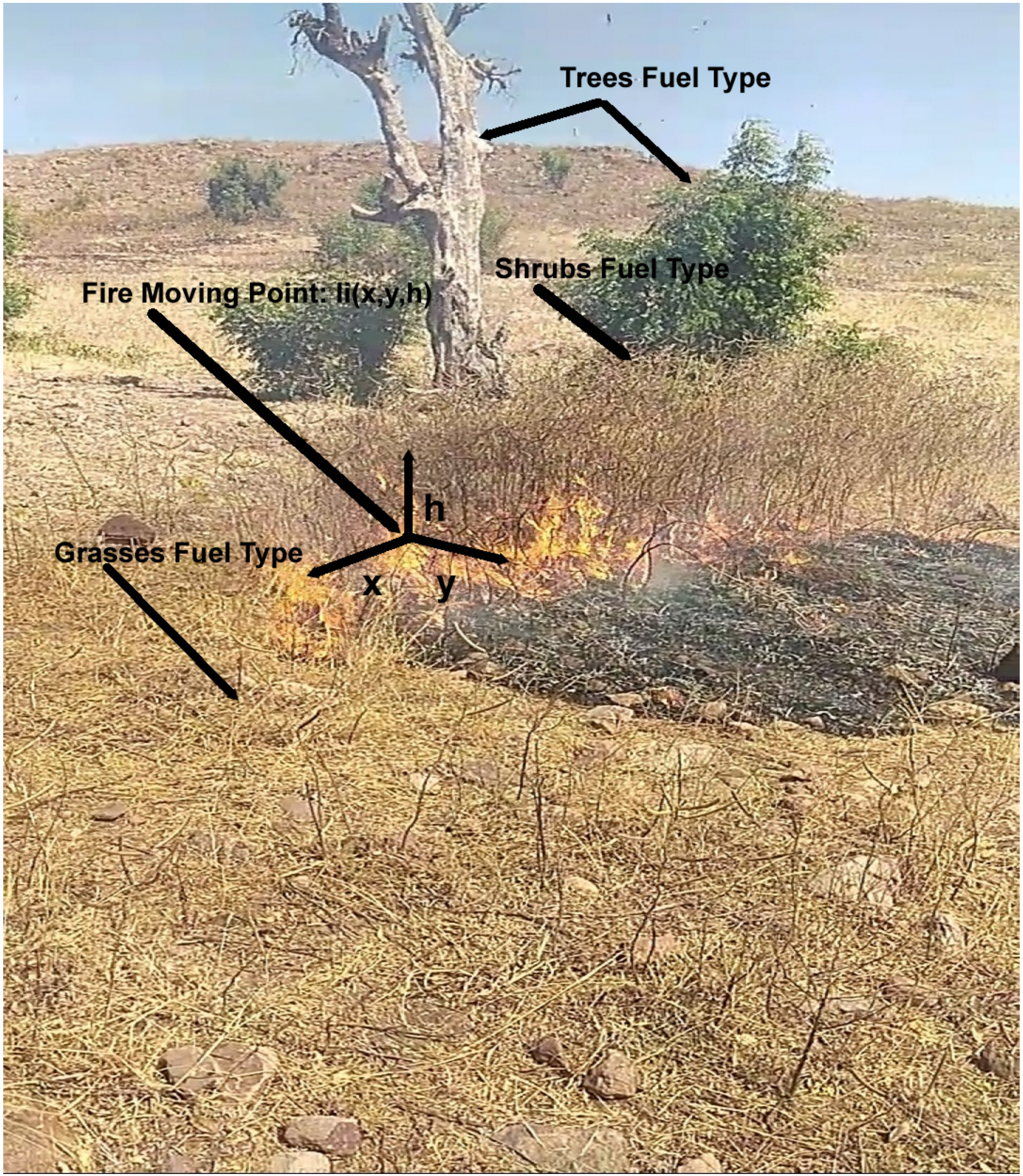

Trees Fuel Type
Shrubs Fuel Type
Fire Moving Point: li(x,y,h)
h
Grasses Fuel Type
x
y

Figure 34: Fire Spread Experiment

The fire spread rate (from Figure 34) is measured by making marks using stones and monitoring the time the fire started and the time it reached the spot in consideration of other parameters, e.g., wind speed, wind direction, fuel types, etc. The results of this experiment are reported in the following section (i.e., Section 6.6).

The UAV's specifications are based on the DJI Phantom 3, Parrot Bebop, and DJI Ryze Tello drones. The energy consumption was measured using DJI Ryze Tello Edu drones through its Python API[11]. The drone is tasked with a particular flying mode (e.g., ascending) for 5 meters and then monitors the energy consumption rates under a stable weather. The sensor's configuration was updated based on the DJI Phantom 3 camera capacity and the sensor behaviours (Alkhatib, 2014). Values and configurations of the UAVs are described in the following section.

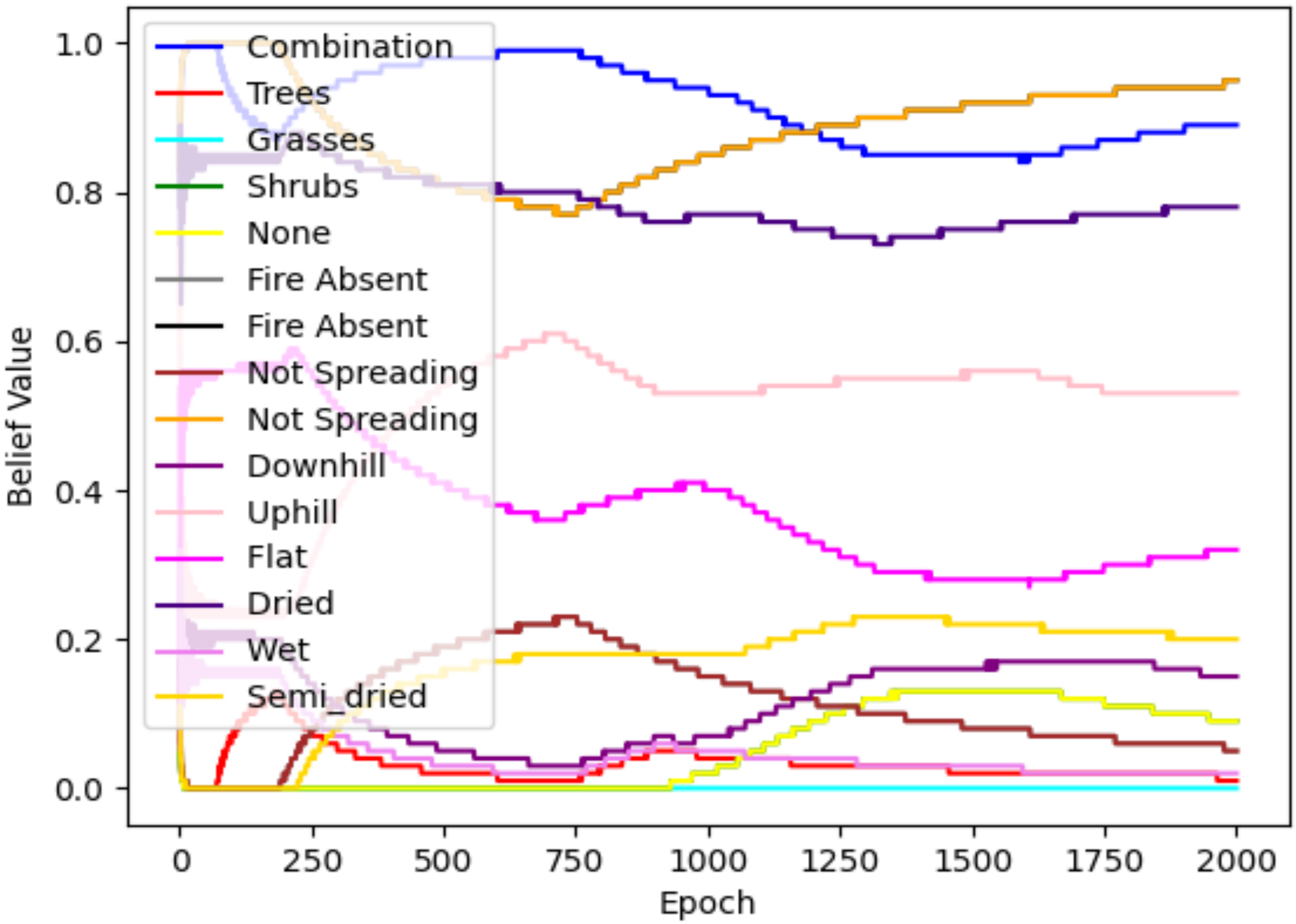
6.6 Results

The result section is segmented into two: sensor data analysis and the results of the physical forest fire experiments. For the agent's priors generation, the UAV is tasked to explore the simulated search area (Chapter 3 Figure 2) using the proposed Delaunay-inspired (from Chapter 3) and the Lévy flight (Chawla and Duhan, 2018; Sutantyo et al., 2011). The selection of these two approaches is to understand the effect of the structure of the search methods on the priors' development and, more importantly, the issues of predictions and uncertainties, as will be addressed in Chapter 7.

The results in Figure 35 and Figure 36 describe the prior initial values for the BBN nodes states of Figure 28 for the range of 2000 reported sensor values. Thus, at each entry, the prior is generated using the algorithm described in Chapter 4 Section 4.8. For example,

[11] https://github.com/code4funSydney/Tello

assume the UAV reports fire present from its first sensor poll, then the prior will be 0.75, i.e., 0.5 (initial prior) +1)/1+1 (as described in Chapter 4 Section 4.8). Thus, Figure 35 and Figure 36 describe the reported agents' sensor values converted to priors for the Levy flight and the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) search plans.



Combination
Trees
Grasses
Shrubs
None
Fire Absent
Fire Absent
Not Spreading
Not Spreading
Downhill
Uphill
Flat
Dried
Wet
Semi_dried
Belief Value
1.0
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
2000
Epoch

Figure 35: Priors Values for a Lévy Flight Mission (Coloured)

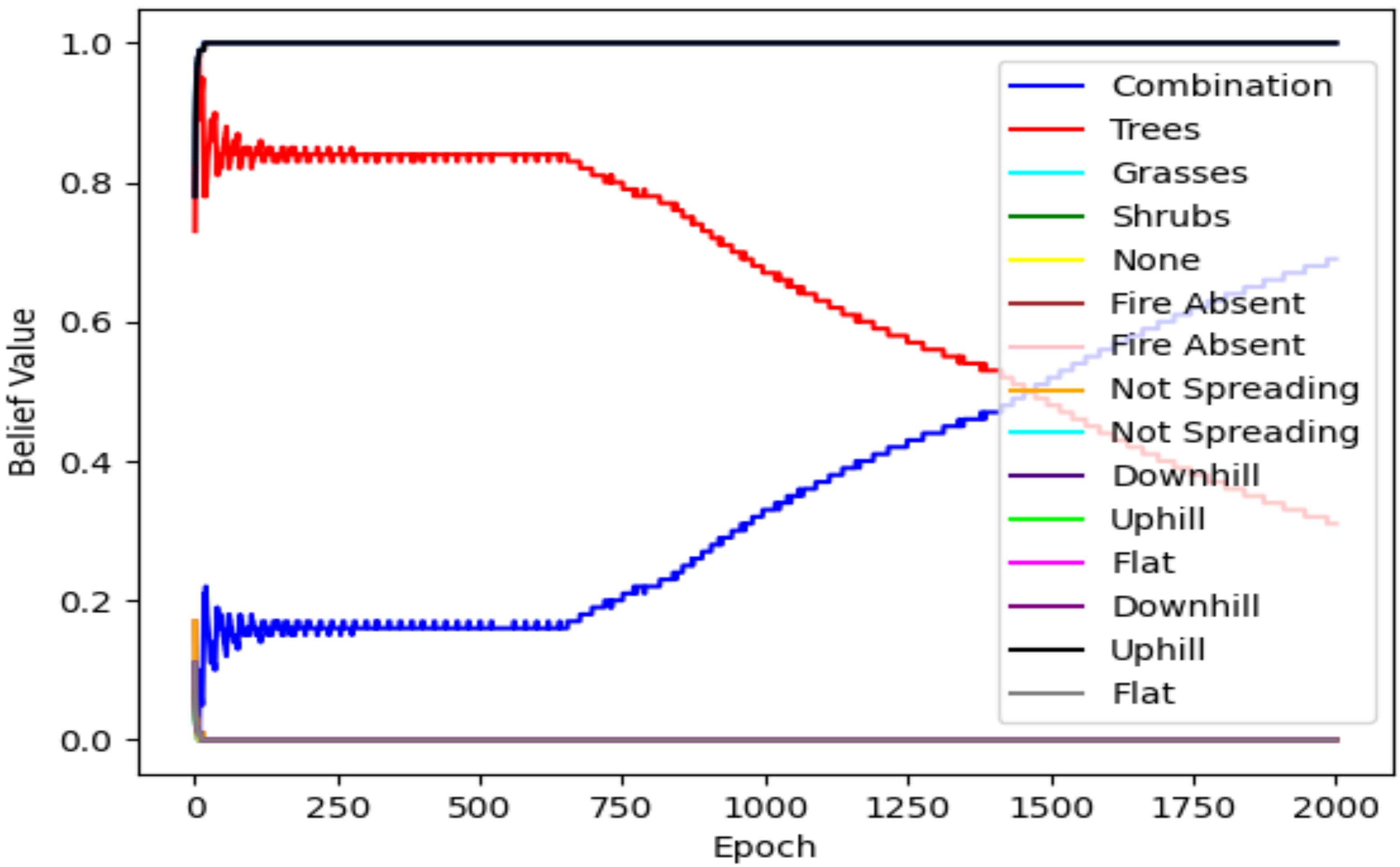


Figure 36: Priors Values for a Delaunay-Inspired Multi-agent Search Strategy (DIMASS) Mission (Coloured)

From Figure 35 and Figure 36, each line represents the state priors of the BBN in Figure 28. The Delaunay-Inspired Multi-agent Search Strategy (DIMASS) priors (Figure 36) show some structure in terms of agents' priors' updates (i.e., linear based on the search area structure described in Figure 32). This allows easy situation prediction using Bayes rules (Chapter 7) or simply using interpolations (as discussed below). The Lévy flight-based priors (Figure 35) show random values. This is based on the pseudorandom waypoints generation, which provides unstructured search area exploration. This will affect the simple prediction models (as discussed below) and, overall, could affect the DSA projection state (i.e., by affecting prediction, which is part of projection). The Delaunay-Inspired Multi-agent Search Strategy (DIMASS) approach of Figure 36 shows good structure, e.g., the cross of red and blue lines of trees and combination fuel type.

The next section of the results discusses the forest fire physical experiment and UAV configurations. Table 20 describes the results for the forest fires spread rate (i.e., from the physical experiment), while Table 21 and Table 22 describe the UAVs and sensors configurations. The spread rates were obtained by setting fire to a quarrying area in Nigeria (Liji Hill of Gombe State). The spread values were obtained by making marks on the directions of the fire spread. The results depend on the search area configurations, UAVs used, and the sensor configuration. A different combination of search, UAVs, and sensors could give a different result. However, the acquired values demonstrate the realism of the system model. For the UAVs functionality values (Table 21) were taken from the DJI Phantom 3 and Parrot Bebop drones. However, the energy consumption was derived using DJI Ryze Tello programmable drone operations. For example, descending consumption rate is measured by tasking the UAV to perform a descending flight mode for a particular distance, and then the energy consumption is calculated as a difference of energy percentage after and before flight (i.e., E_{after} – E_{before}).

Table 20: Different Locations and Forest Fire Spread Rate

#	Spread Length (Meter per second)	Time Frame (minutes)	Wind Direction (mostly)
1	0.002	14	East
1	0.009	14	Northeast
1	0.005	14	West
1	0.006	14	North
2	0.004	8	Southeast
2	0.025	8	Northwest
2	0.02	8	Southwest
2	0.0135	8	South

Table 21:UAVs Modelled

UAV Type	Flight Type	Air Speed(m/s)	Vertical Speed(m/s)	Pitch Angle(degrees)	Maximum Bank Angle(degrees)	Minimum/Maximum Speed (m/s)	Energy Consumption (%/second)	Maximum Altitude(m)
Fixed-wing	Cruising	30	0	0	30	10/40	0.049	400
	Loitering	20	0	5	30	10/40	0.0083	400
	Ascending	30	5	10	30	10/40	0.05	400

	Descending	30	-5	-5	30	10/40	0.025	400
	Dashing	40	0	-2		10/40	0.019	400
Multi-copter	Cruising	20	0	0	30	0/25	0.074	400
	Loitering	20	0	5	30	0/25	0.037	400
	Ascending	20	5	10	30	0/25	0.075	400
	Descending	20	-5	-5	30	0/25	0.05	400
	Dashing	25	0	-2	30	0/25	0.049	400

Table 22:Sensor Configuration

Sensor Type	Video Stream Horizontal/Vertical Resolution (px)	Minimum Horizontal/Vertical View	Supported Wavelength Band	Horizontal/Vertical Field of View	Elevation
Infrared Camera 1	256/192	55/55	Short-wave infrared	10/10	45^0
Infrared Camera 2	256/192	55/55	Mid-wave infrared	10/10	45^0

Infrared Camera 3	256/192	55/55	Mid-wave infrared	10/10	45^0
Spectrum Camera	256/192	55/55	Electro-optical	10/10	45^0
Thermistor (Temperature Sensor)	-	-	Heat sensor	10/10	45^0

The results in Table 20 to Table 22 were used for the AMASE simulation inserted through the XML values or the Java controllers. The complete simulation source codes (including all Java classes and XML) can be found in the supplemental document folder (i.e., from the appendix).

6.7 Discussion and Conclusion

Based on the results in Figure 35 and Figure 36, the structure of the search plan affects its predictability (i.e., simple predictions based on priors values as discussed in Chapter 3). The Lévy flights give non-organised states priors because the search method operates randomly in a structured search area. Structured approaches e.g., the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) produces more organised priors values, which can simply be predicted using an interpolation (i.e., by considering how other similar co-states evolve and estimating the target state priors based on the results in Figure 35 and Figure 36). For instance, assume a BBN with n number of states and m nodes. The values of n_i belief could be used to estimate the value of the n_j where $j \neq i$. This can be done by considering the degree of correlation among the states' priors. Equation 14 describes a state prior prediction using interpolation approach.

$$N_{i\_estimation} = w_j n_j$$

Equation 14: Situation Prediction using Interpolation

where $N_{i\_estimation}$ is the estimated value of the uncertain variable, n_j is the observed value of co-state within the BBN, and w is the transformation matrix that transforms the previously observed values into the predicted values. The value of w can be computed as a function of angle, distance difference, correlation, and the previous values to be considered as described in Equation 15.

$$w_j = d \times \Theta \times k_{prev} \times \rho$$

Equation 15: Prediction Weight

where d is the distance difference, e.g., Manhattan distance (Greche et al., 2017), Θ is the angle difference between the values, and k_{prev} is the k-previous number of previous states to be considered, and ρ is the correlation measure. In other words, Equation 14 prediction considers the similarity of the priors trend to make predictions. For example, if the wind direction prior value for the past 10 minutes is 0.9 and fuel type is grass:0.5, then the prediction or missing value of the wind fuel type will likely be 0.5 or any fraction of it based on the value of other states and their transition. That is, the prediction is based on the similarities of the trends of the prior. Equation 15 will likely be a good prediction for the proposed Delaunay-Inspired Multi-agent Search Strategy (DIMASS) priors based on the linearity and similarity of the priors' values. For instance, the exact overlapping priors' values from different states can predict similar values. Confidence in the prediction can be built by counting the number of states that suggest the same values i.e., constant previous values are expected to bring the values. Again, simple prediction using the Bayes rule could best work on the structured values described in Equation 16 due to the priors growth.

$$P(S_i|D) = \frac{P(D|S_i)P(S_i)}{\int_{S=prev_1}^{S=prev_n} P(D|S_i)P(S_i)\, dS_i}$$

Equation 16: Bayes Rule Prediction

Note that prediction here refers to simple prediction using parameters (similar to the one in Chapter 3). Thus, the agent's search plan structure can support the prediction of the state, which overall supports the SA projection state (i.e., in the case of plausible future state prediction and uncertainty handling). A key limitation to this sort of prediction and uncertainty handling is when the state's priors keep fluctuating due to the lack of structure in the search plans or the operating search area phenomena values. This sort of prediction requires iteration through the fluctuating priors. Therefore, Chapter 7 addresses the challenge using different methods such as the expectation-maximization algorithm, Gaussian Process, and time series models.

7 Chapter 7 Prediction and Uncertainty Handling

In Chapter 4, I proposed using BBN to model the system SA and described its various potential advantages. This chapter will address a solution to predictions and uncertainties challenges for the BBN states' priors. The need for addressing prediction and uncertainty handling in DSA systems is important due to agents' spatial differences, goals variation, possible hardware/software issues, sensor faults, etc. The chapter addresses single and multiple states predictions, and uncertainty handling using the Expectation-Maximisation (EM) algorithm, time series models, and Gaussian processes.

7.1 Introduction

The need for prediction and uncertain priors' values estimation for the BBN is critical to the DSA system due to agents' spatio-temporal differences, goal variation, and the search area's dynamism (as outlined in Chapter 1). For example, the past values of wind direction can be beneficial in predicting (perhaps with high accuracy) its future or missing value(s) when the reporting agent is yet to submit its sensor information or experience some failure. This could help the system make decisions, for example, where a fire will move shortly, even in the absence of information or the presence of incomplete information. Thus, the aim is to suggest the best prediction and uncertainty handling models for the proposed SA modelling tool (i.e., the BBN in Chapter 4). Like predictions, uncertainty is defined as the degree of doubtness of a state of the BBN nodes (Fioretto et al., 2018; Hoang et al., 2016; Li et al., 2019; Park et al., 2016). In this chapter, uncertainty in a BBN state generally exists in two forms: (i) a missing value and (ii) a soft finding (incomplete) value. A missing value refers to a completely unknown sensor value, for instance, due to agents or sensor failure. A soft finding refers to incomplete sensor information, e.g., due to sensor misbehaviour. For example, assume that an agent responsible for wind speed detection submits a value of 150m/s, which is unrealistic to the search area's values; this will generate some doubts about whether the value is correct or not. This sort of uncertainty is referred to as soft-finding. Therefore, based on the outlined system challenges, the uncertainties can be in one of the following forms:

i. Complete missing values: missing values due to sensor error or mission incident, e.g., UAV camera blocked by other entities, snow, fog, etc.

ii. Discretised values: uncertainties in some variables can be between an interval or set of possible values based on the learnt knowledge or SOP. For example, temperature sensor values can be distorted by the speed or altitude of the UAV. Thus, probabilities can be assigned for any possible discrete value based on its occurrence likelihood, e.g., an increase of 3^{o}C to a temperature value when a UAV operates at an altitude above 75meters. Alternatively, a list of possible values can be presented as well. The winning value will be the one with a higher likelihood.

iii. Agents' sensor values conflict (contradiction): Agents can report different values due to their sensing mode. For example, yellow dried grasses can confuse fire detecting sensor that applies image recognition (visual camera) to detect fire, i.e., a fire-like object. This mode of contradiction requires careful analysis of the situation, which may demand input from several other agents (e.g., visibility reports, weather reports, etc.) or CPT consultation, as described in Chapter 5.

iv. Update delay: occurs from information decay due to error (e.g., missing data) or data collection method. Thus, the types of uncertainties in (i) are missing values uncertainty, whereas (i-iv) are the soft findings.

It was noted in Chapter 4 that an exciting aspect of applying Bayesian Belief Network (BBN) for system SA modelling is the depiction of the logical relations among nodes designated to understand a particular situation and offers a way of measuring beliefs of the search area phenomena using probabilities. Although BBN does not give a clear insight into how future beliefs can be forecasted based on the previous transitions, it does offer a benefit for applying existing prediction algorithms to handle the prediction and uncertainties issues (Karduni et al., 2021; Pavlin et al., 2010). It offers a possibility of considering other related concepts' beliefs values in predicting or estimating a particular state prior based on the BBN logical relations and previous probabilities values, i.e., a multiple state prediction or uncertainty handling. Single state prediction can also be performed based on the values of the querying state. Thus, the critical question to be investigated is in this Chapter is:

How could prediction and uncertainties in the DSA system be addressed? Which is a subquestion to the main research question: "*RQ2. How can we manage the Situation Awareness of distributed agents?*"

One of the critical challenges to prediction and uncertainty handling is quantifying the number of previous values needed to obtain a good prediction or uncertain value estimations. This varies based on the type of the priors regression (i.e., either linear or non-linear, as discussed in Chapter 6), the algorithm applied, and the demanded accuracy (Romanycia, 2019). For instance, if the previously generated *n* number of entries can predict *k* future values with an *e* error rate using method A, then the time taken to generate the data (i.e., the quantity of the data) and error rate obtained will be used to measure training entries (i.e., to decide on prediction acceptance or rejection). Additionally, the chapter will consider proposing DSA-based metrics for grading the prediction and uncertainty handling issues.

The evaluating methods were the Gaussian process, expectation maximisation algorithm, and time-series models due to their popularity. The evaluation process utilises the experiments described in Chapters 3, 4, 5 and 6 based on the thesis use case of Chapter 1.

7.1.1 Existing Methods

Gaussian Process (GP) and time series models [autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), seasonal autoregressive integrated moving average (SARIMA)] were selected as the participant methods for the BBN single states prediction and uncertainty handling, while expectation maximisation algorithm is the candidate for the multiple states one (Dempster et al., 1977; Gan et al., 2021, 2021; Leith et al., 2004; Nath et al., 2021; Papastefanopoulos et al., 2020; Romanycia, 2019; Shumway, 1984; Tandon et al., 2020). The reason behind the selection of the methods is because of their efficiency in utilising computational power, ability to handle missing variables, popularity, and the presence of existing software packages to ease the application process[12] (Dempster et al., 1977; Hendikawati et al., 2020; Romanycia, 2019; Schwaighofer et al., 2004; Tandon et al., 2020). Although other algorithms such as Bayesian rule and conjugate gradient descent algorithms, could perform the same task, the justification for selecting the EM algorithm is based on its faster running speed, popularity, and better performance (Mandt and Hoffman, 2017; Romanycia, 2019).

7.1.2 Gaussian Process

Gaussian process is a collection of values of which any arbitrary joint distribution of those values follows a normal distribution (Banfield and Raftery, 1993; Caywood et al., 2017; Görtler et al., 2019; Hendikawati et al., 2020; Rasmussen and Williams, 2006; Wagberg et al., 2017). This feature makes prediction possible using the mean and standard deviation as described in Equation 17.

$$P(S|D) \sim N(\mu, \sum)$$

Equation 17: Prediction using GP

where S is the querying state probability, D is the training datasets defined as, $D = \{(x_1,y_1), (x_2,y_2),\ldots,(x_n,y_n)\}$ as described in Chapter 6 Figure 30, $\sum$ is the covariance matrix (as described in Equation 18) define by the kernel function $K_{ij} = K(x_i,y_i)$ for every states values. For example, the prediction of the x_i variable given the training data D (previous values) is: $P(x_i|D) = \int_{d=1}^{d=n} P(D|x_i)P(x_i)/P(D)\ \delta D$. Note that the value S is the current prior, and P(D) is the prior for the related states. For the BBN data to fit the Gaussian process, the probability transition of nodes states values is assumed to be Gaussian (i.e., based on priors update algorithm of Chapter 4 Section 4.8), and the joint distribution of nodes entries is also Gaussian based on the law of large numbers (LoLN) and central limit

[12] A Practical Implementation of Gaussian Process Regression

theorem(CLT)(Brosamler, 1988; Hsu and Robbins, 1947). The LoLN and CLT guarantee that the sensor information can be predicted effectively by considering the previous entries (Kunda, 1986). The application of Gaussian process for BBN states priors prediction and uncertainty handling in this thesis follow these steps:

Step 1: Gather the agents' sensor data in rows and columns (Chapter 6).

Step 2: compute the variance between the data.

Step 3: Select the kernel length, i.e., the difference in values length that determines whether two subsequent entries are related. The kernel length could vary based on the number of previous values (e.g., by identifying the seasonal lags using Auto-Correlation Function(ACF) or Partial auto-Correlation Function(PCF) as discussed below (Adhikari and Agrawal, 2013; Dama and Sinoquet, 2021; Tandon et al., 2020)

Step 4: Choose the kernel. The kernel function $K(S_i, S_j)$ computes how these entries relate. Radial Basis Function, linear, and periodic kernel (Görtler et al., 2019) are the most popular kernels applied in literature. However, there are many other kernels and ensemble strategies (Lu et al., 2020; Schwaighofer et al., 2004).

Step 5: Construct the covariance matrix for each pair of states entries $\begin{bmatrix} \Sigma s_{ii} & \Sigma s_{ij} \\ \Sigma s_{ji} & \Sigma s_{jj} \end{bmatrix}$

Step 6: Compute the prediction probability of the state given related entries using Equation 17, i.e., $P(S_i|S_j) \sim N(\mu, \Sigma)$

Where S_i is the querying state and S_j is the dependent state.

$$P(S_i,S_j) = N\left(\begin{bmatrix}\mu_i\\ \mu_j\end{bmatrix}, \begin{bmatrix}\sum s_{ii} & \sum s_{ij}\\ \sum s_{ji} & \sum s_{jj}\end{bmatrix}\right)$$

Equation 18:GP Equation

Where,

$$\mu = \mu_i + \sum s_{ij} \sum {s_{jj}}^{-1}(S_j - \mu_j)$$

Equation 19:GP Mean

$$\sigma = \sum s_{ii} - \sum s_{ij} \sum {s_{jj}}^{-1} \sum {s_{ij}}^{T}$$

Equation 20: GP Standard Deviation

The priors are computed using the density function:

$$p(s) = \frac{1}{\Sigma\sqrt{2}} e^{-\frac{1}{2}(\frac{S-\mu}{\Sigma})}$$

Equation 21: GP Distribution

The Gaussian process can estimate an uncertain value using Equation 22.

$$U = P(s)x \pm (Zx\sigma)$$

Equation 22: GP Uncertainty Measurement

where U is the uncertainty measure, Z is the confidence value from Z-score table, and σ is the standard deviation. The +/- symbol determines the confidence level deviation across the prediction line, i.e., above and below values as described in Figure 37. The U value is the blue line across Figure 37.

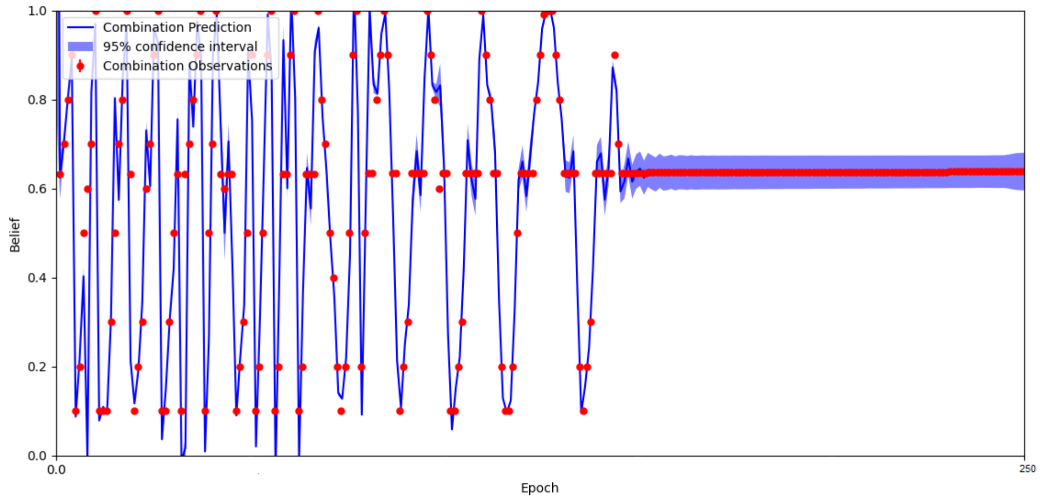


Figure 37: Example of Gaussian Process Prediction and Uncertainty Handling.

Figure 37 shows an example of GP prediction and uncertainty handling using Equation 17 to Equation 22. The dot points (in red) are the observed values, whereas the predicted values are blue surrounded by their confidence level.

7.1.3 Time Series Models

The time series models work based on parameters p,q,d, and m. The p is the k-previous trends values (where k is a positive integer), e.g., if the fuel condition has been "Dried Grasses" for a p collected data at time t, then it is expected to be "Dried Grass" at time t+1. The d is the differencing threshold value, i.e., the measure of the difference in the previous values that allows the separation of a group of data during prediction. If the difference in previous transitions is less than or equals to d, then the values are considered part of p. The variable q is the measure of error (difference in the prediction error).

In the Autoregressive (AR) Model, future state priors' prediction is computed as a linear combination of k-previous p values. The moving-average model (MA) considers the term q (i.e., size of the moving average differencing window) as the prediction limit. For instance, if q is 5, the predictors for x(t) will be x(t-1), x(t-2), x(t-3), …, x(t-5) etc. Similarly, the ARMA model considers the combined p and q parameters of the differenced data to make predictions. In contrast, the ARIMA model extends the ARMA model to consider d (differencing) windows. The SARIMA model considers the seasonal trend m (i.e., for non-linear priors' values, e.g., for Lévy flight values or unstable operation of the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) data as described in Chapter 6), which is to be determined using the Autocorreletion Function/Partial Autocorreleation Function (ACF/PCF) as described below i.e., the seasonal version of p, q, d as P, Q, D, i.e., seasonal hyperparameters. The P and Q are determined using the following rules[13]:

i. "If the PACF of the differenced series displays a sharp cut-off and the lag autocorrelation is positive, i.e., if the series appears slightly "under differenced", then consider adding an AR term to the model. The lag at which the PACF cuts off is the indicated number of AR terms".
ii. "If the ACF of the differenced series displays a sharp cut-off and the lag autocorrelation is negative, i.e., if the series appears slightly "over differenced", then consider adding an MA term to the model. The lag at which the ACF cuts off is the indicated number of MA terms".

Figure 38 describes a visual example of identifying the hyperparameters p and q for the time series models using ACF/PCF.

[13] Identifying the orders of AR and MA terms in an ARIMA model [WWW Document], n.d. URL https://people.duke.edu/~rnau/411arim3.htm (accessed 6.28.21).

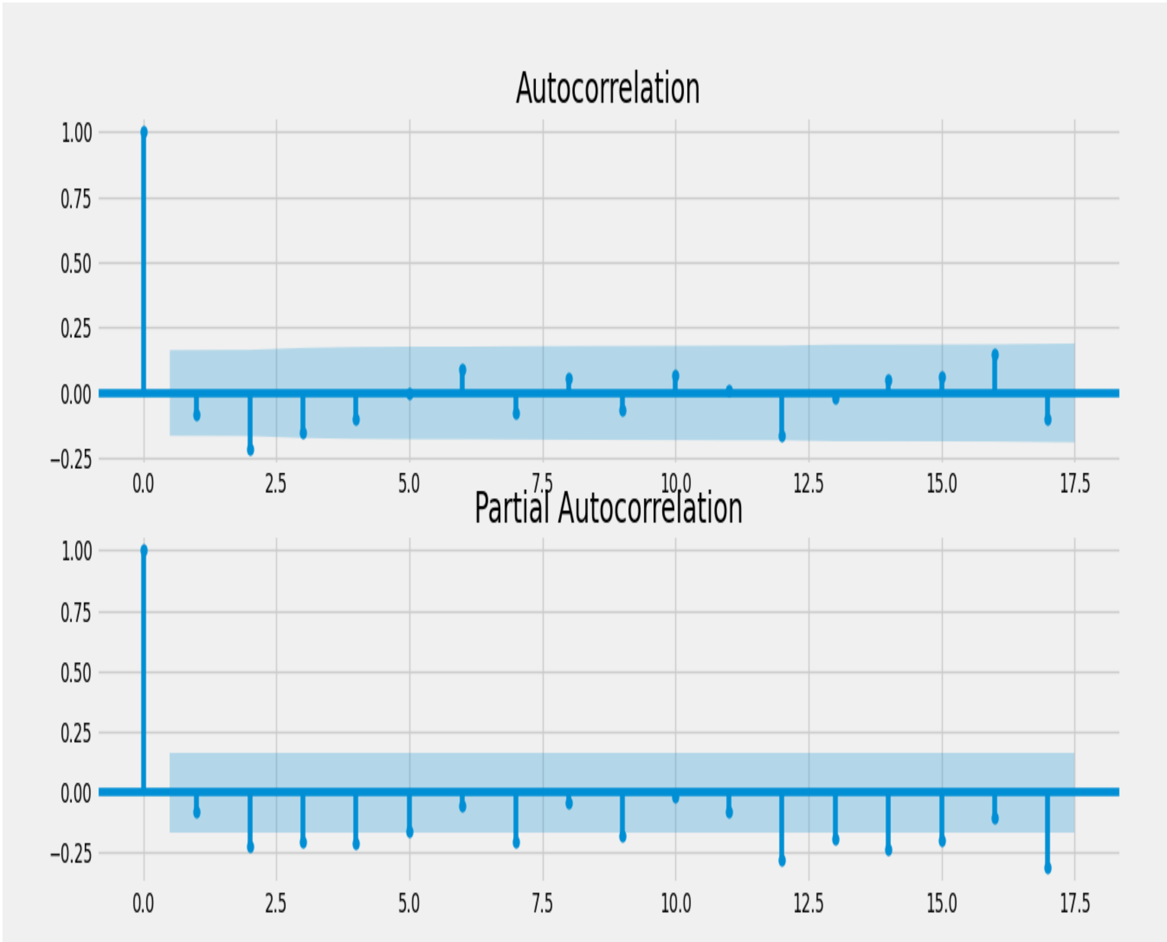


Figure 38:An Example of Hyperparameters Identification

From Figure 38 and the above rules, the values of p and q are 1, i.e., p =1 and the q= 1. Thus, identifying the time series hyperparameters in DSA system may require visual analysis by the SMEs.

7.1.4 Expectation-Maximization Algorithm

In joint states prediction, the whole BBN is considered as a single predictive model. Thus, states values are considered in group. The Expectation-Maximisation (EM) algorithm (Dempster et al., 1977) apply the following steps to make predictions or handle uncertain values for the joint states models:

Step 1. Expectation State: Predict the target state using the Bayes rule (Equation 23).

$$P(S_i|D) = P(D|S_i)P(S_i)/P(D)$$

Equation 23: EM Expectation State

where S_i is the probability of each state occurring, i.e., i= 1,2,3,…, n i.e., agent's previous perceived data. For example, when the search area experience wind direction towards north for a long period of time, then it is highly probable that it is going to be north in t future time, thus Equation 23 will likely return north as wind direction.

Step 2. Maximisation State: This step maximises the likelihood of the chosen future state using an iteration process. The algorithm replaces the selected value with the revised value (i.e., revised by going through the previous entries). Iteration continues until convergence (i.e., the difference between the previous and current values is negligible as specified by the SMEs, e.g., 0.01 or maximum iteration is reached), then the converged value is retrieved as the prediction. An intuitive feature of this algorithm is that during each iteration, the predicted value is greater than or equal to the previous value, as proved by Jensen's inequality (Adil Khan et al., 2020). Thus, the iteration process is responsible for maximising the probability of the predicted value based on previous entries. Other algorithms, such as the counting algorithm (Romanycia, 2019) i.e., ordinary Bayes rule and gradient descent algorithm (Bottou, 2010; Mandt and Hoffman, 2017; Romanycia, 2019) can be applied to train the BBN. However, the counting algorithm is limited by its inability to handle missing variables and gradient descent algorithm is limited by being computationally expensive, although both algorithms produce similar results (Romanycia, 2019). This serves as my main reason for selecting the EM algorithm.

7.1.2 Hypothesis

It is hypothesised that:

i. Prediction and uncertainty handling of DSA system depends on the agents' search pattern structure
ii. Different forms of uncertainties have different behaviours in terms of prediction and uncertainty handling in DSA system

The hypothesis will be tested on the agent's mission data generated in Chapter 3 and 6.

7.2 Performance Metrics

Error rate counts the number of wrong predictions, i.e., after values reception, which serves as the main performance metric (i.e., the measure of a number of times a BBN made wrong predictions upon reception of the correct values). This is the basis for other metrics such as logarithm loss, Brier loss, spherical payoff, root-mean-square error, etc. (Karduni et al., 2021; Morgan and Henrion, 1993; Pearl, 1978). The Error rate omits the DSA-based features, i.e., the contribution of relevant nodes aims to achieve a particular goal. Therefore, this chapter proposes the $M^{+}_{relevance}$ metric (Equation 24) to address the challenge.

$$M^{+}_{relevance} = normalised\left(\frac{\sum_{j=1}^{i=m} P(S_T^{+}) + \lambda_i P(R_i^{+})}{\mathrm{N}}\right)$$

Equation 24: Relevance Based Metric

where $P(S_T^+)$ and $P(R_i^+)$ are the probability of the correctly predicted state for the querying and related nodes entries. N is the number of the querying node and other related nodes, λ_i is the measure of relevance between the querying and related nodes (defined in Chapter 4 Section 4.3), and $i \in [1, N]$

The $M_{relevance}^+$ value ranges from 0 to 1, where 0 means the prediction accuracy is poor, and 1 signifies a high level of prediction or uncertainty handling accuracy. Therefore, Equation 24 measures the degree of accuracy of the querying node's predictions/uncertain values estimations and reflects the contribution made by the connected nodes based on their importance. Thus, this captures the relationship between nodes in a DSA system (because some will be more relevant to a specific goal than others).

Similarly, changes in prediction accuracy over time are important in determining how the DSA system is performing over mission period. This is because different nodes within a group of nodes have different prediction accuracies depending on their belief (priors' values), available data, and change rate. A pivot (a point that signifies a change in the situation) can be set to check the performance of a node or group of nodes over time using Equation 25. Thus, based on $M_{transition}$, an agent can decide whether to use the predicted value of a node based on its prediction accuracy history. This measure alerts the system about improvement or otherwise of a node's prediction. Thus, Equation 25 can monitor a node's prediction accuracy before and after a pivot.

$$\mathrm{M_{transition}} = \frac{\sum_{i=1}^{i=m}[\mathrm{P_i}(S_{t+1_i}^+) - \mathrm{P_i}(S_{t_i}^+)] + \sum_{j=m+1}^{j=n}[\mathrm{P_j}(S_{t+1_j}^+) - \mathrm{P_j}(S_{t_j}^+)]}{n}$$

Equation 25:Transition Monitoring

where $P_i(S_t^+)$ and $P_i(S_{t+1}^+)$ are the correct situation predictions at time *t and t+1*, the balance pivot (middle of the monitoring points, i.e., change of state's priors trend) selects $n/m = 2 \ \forall\, m \in n$. If $M_{transition}$ is positive, then prediction accuracy from pivot to current time improves, and the negative value indicates prediction is diminished from the pivot. The zero value of $M_{transition}$ signifies stable transition. Note that the $P(S_T^+)$ and $P(R_i^+)$ can be replaced with $M_{relevance}^+$, when $M_{relevance}^+$ transition is to be monitored.

To illustrate the proposed metrics, consider a prediction based on 'spread' as the target node. Assume the degree of relevance λ (from Chapter 4) is 0.23 for the "Spread" node, and fire presence is 0.2. The weights of the nodes are 0.7 for fuel type, 1 for fire presence, and 0.8 for location relation to the ground. From this, $M_{relevance}^+$ (Equation 24) is:

$M_{relevance}^+$ (Spread) = ([(0.26)+(1.3x1x0.26)]+[(0.26)+(0.86x0.7x0.8)]+[(0.26)+(0.23x1x0.1)]+[(0.26)+(1.2x0.8x0.9))/(1x4)

= 2.7466/4

= 0.69.

The value of $M_{relevance}$ of 0.69 (69%) indicates a good prediction accuracy of the target node with respect to the related nodes (fuel type, location relation to ground, and fire nodes). This signifies good prediction for both target and related nodes. Note that the priors will be computed using the algorithm described in Chapter 4 Section 4.8. Therefore, using the scoring rule, one can determine how the DSA system group of nodes perform in terms of projection, while the $M_{transition}$ value determines the nodes or group of nodes prediction transition. Therefore, the proposed scoring rules judge prediction accuracy on the degree of relatedness of nodes and prediction efficacy over time. Thus, the following are the performance metrics to be used in evaluating the performance of the algorithm

i. Error rate: number of times the method made a wrong prediction or uncertain values estimation out of the provided inputs. This is measured as the proportion of the false predictions normalised to 1.
ii. Prediction/uncertain values estimation interval: the time range of the prediction measured by the number of testing values (i.e., to be derived from the sensor rate, e.g., 200 samples in every 30 minutes).

iii. Surprise rate: number of times a prediction/uncertain value estimation was made with X% probability chance and found to be wrong after values reception.

iv. Resources utilisation: this is measured using time complexity and memory use.

v. The proposed metrics ($M_{relevance}$ and $M_{transition}$). This can be derived from the error rate values.

7.3 Evaluation

The experiment uses the simulated UAV's mission data using Lévy flight and the proposed Delaunay-Inspired Multi-agent Search Strategy (DIMASS) method described in Chapter 3. The processes for data collection and BBN construction were described in Chapters 6 and 4, respectively. The BBNs used for the evaluation are illustrated in Figure 39 and Figure 40. The difference in sizes of the BBNs is to describe the effect of the BBN number of nodes and data on prediction/uncertainty handling.

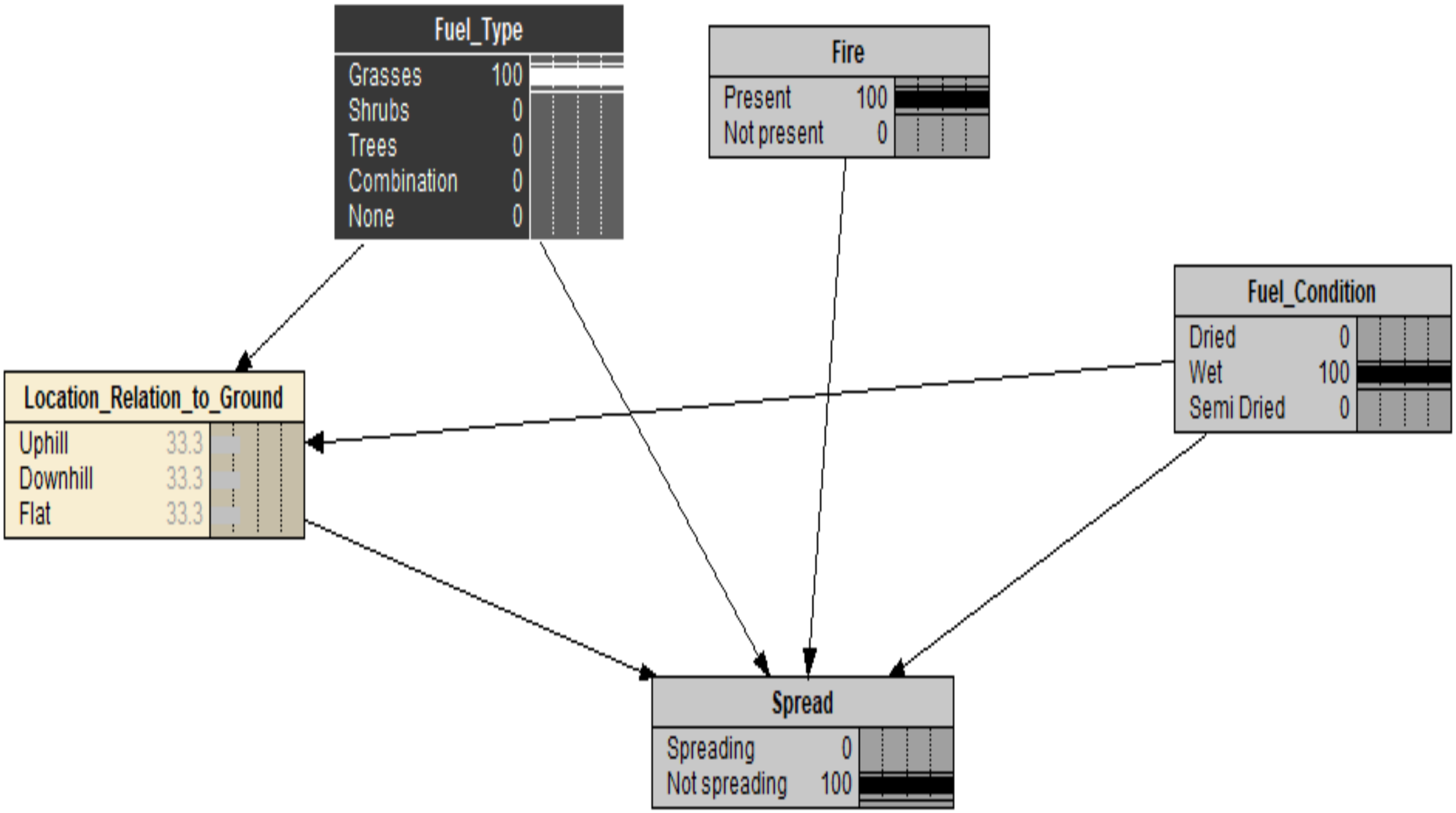


Figure 39: Example of BBN for Fire Spread (PC level)

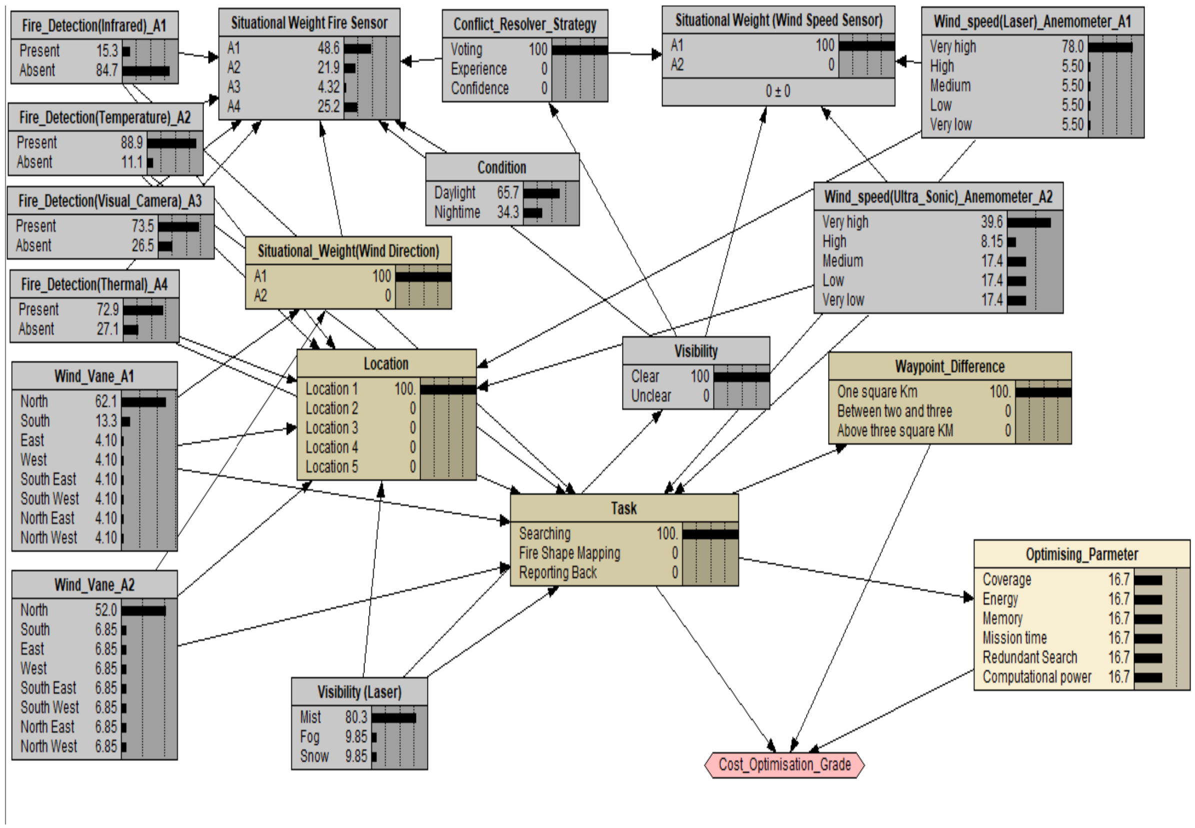


Figure 40:Larger Bayesian Belief Network (NETICA software) for SA Model at Host Level

7.3.1 Task

The task is to predict or estimate the values of the states of the BBN nodes in Figure 39 and Figure 40. The node "Fuel_Type" and state "Combination" are selected based on their dynamic priors' values behaviours as described in Chapter 6. For the single state prediction/uncertainty handling, the task was to predict the future priors of the state "Combination" of the 'Fuel_type' node from Figure 39. The experiment was run on a computer with 8GB RAM and an Intel(R) Core (TM) i3-6006U CPU @ 2.00GHZ. The experiment results were obtained by running the Python functions (i.e., for the time series models and Gaussian Process), Java methods using NETICA API (i.e., for the EM algorithm), and loading the mission data in .csv (as described in Chapter 6). That is, the experiment set up follows the following steps:

Step 1: load the .csv

Step 2: load the relevant classes e.g., the Python GP plugin, NETICA BBN learning classes, or time series models Python plugin.

Step 3: call the learning method and pass the data in form of arrays or dictionary.

Step 4: the learning methods will then return the results

7.3.2 Results

This section discusses a series of experimental results on prediction and uncertainty handling for the outlined methods.

7.3.3 Prediction Results

Table 23 shows the performance of the models predicting the "Combination" state of fuel type node from the BBN in Figure 39 for the time series models and Gaussian process. The number of predicting sensor information is 250. The number 250 is selected as the average data generated within 5 minutes of the agents mission, which is needed to present the situation of the forest fire (Ingle, 2011). The data were obtained from the AMASE agents' mission described in Chapters 3 and 6. That is, each tuple of the data contains the agents' sensor information across all the states of the nodes stored in a .csv file. The Python implementation of the algorithms (as available in the supplemental document) will then fetch data and perform predictions for every single entry. For example, considering the time series models, the parameters are defined, and the python time series plugin loads the .csv file to perform the prediction analysis. The overall error rate is then reported as the measure of the performance. Both search methods' data offer the same result for the 250 entries due to the configuration of the simulated search and the short range of the sensor sampling (i.e., the structure of an explored portion of the search area is the same due to short-range and the location of the agents).

Table 23: Single Approaches Prediction Performance on the Agents Sensor Data

Method	Number of Predicting Sensor Information	Error rate
Autoregressive	250	0.52
Moving Average	250	0.87
Autoregressive Moving Average	250	0.63
Autoregressive Integrated Moving Average	250	0.50
Seasonal Autoregressive Integrated Moving Average	250	0.27
Gaussian Process	250	0.08

The results in Table 23 show that the Gaussian process prediction model performs better than the time series models in terms of single state predictions. This is as a result of being data-driven and adaptable to the data pattern based on the mean and standard deviation of Equation 17. Additionally, the implementation of the Gaussian process requires little effort in terms of parameters specifications (i.e., it is data-driven). That is, it does not require manual parameters definition. The time series models require the definition of hyperparameters (p,d,q, P, D, Q, or m specifications) which requires SMEs judgements and could make the process post-hoc and inefficient (due to delays). Despite the claim for seasonal trends treatment by the SARIMA model (Adhikari and Agrawal, 2013; Dama and Sinoquet, 2021), the agents generated pattern was unstructured enough to make it performs poorly. This is due to having varying seasonal trends based on the search area generated priors. The next section of the results investigates the aspect of prediction/uncertainty handling concerning group of nodes and states using the EM algorithm.

7.3.4 Multiple States Prediction

The EM algorithm was used for the multiple states prediction on Lévy flight and Delaunay-Inspired Multi-agent Search Strategy (DIMASS) mission data. Similar to the single approaches to prediction/uncertainty handling, the EM algorithm loads the .csv file containing agents AMASE mission data. However, all the columns (entries of the states as described in Chapter 6) are considered for the prediction/uncertainty handling. Again, the EM consider two data samples, these are the training samples and the testing samples. The training samples will be used to generate the priors for the computation of the prediction and uncertainty handling. The testing data is the data to be predicted or estimated (i.e., the assumed missing or yet to be received values). Each of the testing data is first predicted/estimated based on the priors and check if the value was correct after reception. The proportion of the wrong predictions is reported as the error rate. That is, a learning outcome with 0 error rate is the best outcome. The EM algorithm utilises the NETICA Java API[14] implementation of EM algorithm. That is, Java methods were used to load the BBN (i.e., developed BBN as described in Chapter 6) and run the EM algorithm throughout the data. The final output consists of the defined performance accuracy values, e.g., error rate (as adopted by the thesis), logarithm loss, Brier score, etc. The experiment is repeated across varying numbers of sampling data and testing data obtained from different agents search patterns operating in the environment described in Chapter 6. Table 24 and Table 25 describe the counterpart result of training the BBN with 250 data (i.e., similar to the result in Table 23).

[14] https://www.norsys.com/netica-j.html

Table 24: Multiple States Prediction Results for Delaunay-Inspired Multi-agent Search Strategy (DIMASS) Based Data

#	Sampling Data	Testing Data	Error Rate
1	2500	250	0.32
2	2250	250	0.32
3	2000	250	0.32
4	1750	250	0.32
5	1500	250	0.32
6	1250	250	0.32
7	1000	250	0.32
8	750	250	0.32
9	500	250	0.32
10	250	250	0.32

Table 25: Multiple States Prediction Results for Lévy Flight Data Based Data

#	Sampling Data	Testing Data	Error Rate
1	2500	250	0
2	2250	250	0
3	2000	250	0
4	1750	250	0
5	1500	250	1

6	1250	250	1
7	1000	250	1
8	750	250	1
9	500	250	1
10	250	250	0.0

From Table 24 and Table 25, we can see that the multiple states prediction is more effective on stable priors inputs (i.e., the results with higher error rate e.g., #4 of Table 26 is due to the change in priors values), and the structure of the search method could affect the prediction/uncertainty (i.e., based on the variation in Table 24 and Table 25). The effect of the algorithm on the search pattern was based on the prioritisation of the recent entries by the EM algorithm. For example, according to EM algorithm, if wind direction is being north for last 10 minutes then it is probably north for the next minute (i.e., based on the Bayes rule of Equation 23), the maximisation state refines the prior estimation to adapt to sudden changes. Thus, before the adaptation, the performance becomes really poor as can be seen in #4 - #9 of Table 26. The poor performance of the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) based data is due to being structured (i.e., new entries are always pound based on the structure of the search area and fuel type varies across the search area, note that, testing data are the immediate unexperienced portion of the training data). Thus, we can say that the EM algorithm suits unstructured data than the simple prediction based on the priors' trends as describe in Chapter 6. As such, the prediction/uncertainty handling in DSA systems considers the structure of the data. For the good performance of the Lévy flight, this happens as a result of previous experience of the testing data from the training data i.e., cases of the testing data were experienced before by the EM.

7.3.5 Uncertainty Handling

To understand the uncertainty effect on both training and testing datasets, I consider the following test cases combinations:

i. Complete training versus complete testing datasets (prediction test, i.e., because testing datasets are not part of the training datasets)
ii. Incomplete training (i.e., uncertain training datasets) and complete testing datasets
iii. Complete training and incomplete testing datasets (i.e., uncertain datasets)
iv. Incomplete training versus incomplete testing datasets.

The proportion of the uncertain datasets was 25%, 50%, and 75% of the training datasets, and 24%, 50%, and 76% for testing datasets (i.e., to describe low, medium, and high levels). The selection of the missing data was sequential based on the assumption of occurrence of mission incidence, e.g., sensor failure at certain time-spaces (a portion of the datasets e.g., from 1 to 100 were complete and from 101 to 300 were incomplete i.e., uncertain). Note that, the testing datasets are mostly immediate unseen part of the mission datasets. The series of the experiments are as follows:

7.3.6 Small BBN (Figure 39) and Dataset

This experiment investigates the effect of uncertainty handling on small BBN and varying datasets using EM algorithm and datasets obtained from the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) agents' mission due to their poor performance as in Table 24. Table 26 shows the error rate across a varying number of uncertain variables.

Table 26: EM Efficiency on Prediction and Uncertainty Handling

#	Training Data	Percentage Missing	Testing Data	Percentage Missing	Error Rate

1	2500	0%	250	0%	0.42
2	2500	25%	250	0%	0.21
3	2500	50%	250	0%	0.21
4	2500	75%	250	0%	0.21
5	2500	0%	250	24%	0.47
6	2500	0%	250	50%	0.47
7	2500	0%	250	75%	0.47
8	2500	25%	250	0%	0.29
9	2500	25%	250	24%	0.29
10	2500	25%	250	50%	0.29
11	2500	25%	250	76%	0.29
12	2500	50%	250	0%	0.29
13	2500	50%	250	24%	0.29
14	2500	50%	250	50%	0.29
15	2500	50%	250	76%	0
16	2500	75%	250	0%	0.26
17	2500	75%	250	24%	0.29
18	2500	75%	250	50%	0.22
19	2500	75%	250	75%	0

Based on the result in Table 26, the EM algorithm offers the potential to cope with the missing data uncertainties in both training and testing datasets of varying values. Note that, the high accuracy obtained when BBN is exposed to the missing training and testing datasets, e.g., in #19 of Table 26, is due to the maximum likelihood search of the EM algorithm (i.e., EM assumes that the missing entries follow the observed values patterns). Thus, unexpected events within the interval of the missing values could surprise the BBN as described in Table 27.

7.3.7 Large Datasets Validation

The essence of this experiment is to investigate the behaviour of the EM algorithm given large datasets, BBN, and different uncertainty types. The surprise rate was tested under large training and large testing datasets, as described in Table 27. The reduced error rate was due to the previous experience of the test cases from the training data set (as discussed above).

Table 27: BBN Large Data Validation

#	Number of Training Data	Number of Testing Data	Experienced Before	Completeness	Error rate	Surprise Rate of X% confidence
1	75000	250	yes	yes	0.34	0
2	75000	250	yes	no (50% missing)	0.0	0
3	75000	2500	no	yes	0.13	0

4	75000	2500	no	no(50% missing)	0.25	9.25% of 10% confidence

The larger network's prediction and uncertainty handling behaviour is evaluated using three nodes of different types from Figure 40. For the utility nodes, the "Cost_optimisation_grade" node was selected, i.e., based on the nodes classification of Chapter 5 Section 5.2.1. Awareness and situation node candidates were situational_weight_fire_sensor and wind_vane_A2 (wind direction nodes), respectively. Table 28 describes the result on large BBN.

Table 28: Effect of Uncertainty on Large BBN

#	BBN training data	Missing training data	Node	Testing data	Missing testing data	Error Rate
1	2500	0%	Wind direction(Wind vane A1)	10%	0%	0
2	2500	0%	Wind Speed(Wind vane A2)	10%	0%	0
3	2500	0%	Cost optimisation grade	10%	0%	0.5
4	2500	0%	Waypoint difference	10%	0%	0.5
5	2500	0%	Situational weight fire	10%	0%	1
6	2500	0%	Situational weight wind direction	10%	0%	1

7	2500	0%	Wind direction(Wind vane A1)	10%	40%	0
8	2500	0%	Wind Speed(Wind vane A2)	10%	40%	0
9	2500	0%	Cost optimisation grade	10%	40%	0.5
10	2500	0%	Waypoint difference	10%	40%	0.5
11	2500	0%	Situational weight fire	10%	40%	1.0
12	2500	0%	Situational weight wind direction	10%	40%	1.0
13	2500	40%	Wind direction(Wind vane A1)	10%	0%	0
14	2500	40%	Wind Speed(Wind vane A2)	10%	0%	0
15	2500	40%	Cost optimisation grade	10%	0%	0.5
16	2500	40%	Waypoint difference	10%	0%	0.5
17	2500	40%	Situational weight fire	10%	0%	1
18	2500	40%	Situational weight wind direction	10%	0%	1
19	2500	40%	Wind direction(Wind vane A1)	10%	40%	0
20	2500	40%	Wind Speed(Wind vane A2)	10%	40%	0
21	2500	40%	Cost optimisation grade	10%	40%	0.5
22	2500	40%	Waypoint difference	10%	40%	0.5
23	2500	40%	Situational weight fire	10%	40%	1

24	2500	40%	Situational weight wind direction	10%	40%	1

In Table 28, the estimation error rate at *#1* consider a previously experienced values (i.e., between 2500 to 2750 of the training values). The accuracy was poor because EM priors update prioritises recent sensor values. Thus, the ability to predict or estimate a missing value is situation-based and cannot be generalised. The estimation of *#2* was more accurate because the EM algorithm's recent belief trends are in line with the testing datasets. Based on *#4,* the trained BBN was surprised a few times by low confidence (10%) because of the differences between the training and testing datasets.

7.3.8 Uncertainty Types Differences Evaluation

This experiment investigates the effect of different types of uncertainties using the large datasets for training and testing still on the "Fuel Type" node. Table 29 describes the result for the BBN training using 7500 datasets and 2500 testing datasets with different forms of uncertainties.

Table 29: BBN Large Data Validation

#	Uncertainty Type	Error Rate	Surprise Rate
1	Equal list of all cases {Combination, Grasses, Trees, Shrubs}	1	9.25% of 10% confidence

2	Equal list with two options including the right one	1	9.25% of 10% confidence
3	Equal list with two options, including the right one having the maximum likelihood of 90%	1	25% of 10% confidence

Based on the results in Table 29, the entry in #1 submits uncertain states values of equal chance. This generates a high error rate because the outcome is against the recent training priors' trends (i.e., the consideration of other options completely distorts the precision value, which is unexpected). Similar results arise at #2 and #3 despite the reduction in the number of options and likelihood probability specification. Note that, logical operations can be applied to states configuration, e.g., the negation of non-value options is the same as the exact value, e.g., for the "Fuel type " node of Figure 39, $\neg[Grasses, Shrubs, Trees] = [Combination]$. Interestingly, at #3, despite the maximum likelihood indication for the correct value (i.e., 90%), the BBN failed to estimate the values. This is as a result of the 10% likelihood consideration for the other option which deviates from the maximum likelihood computation for recent values.

Note that, the proposed metrics ($M_{relevance}$ and $M_{transtion}$) are based on the error rates and specific mission goals. Thus, the result is expected to be higher when the related nodes have higher predictions/uncertain values estimation accuracy. Thus, this depends on the error rate outcome of the related nodes and their degree of relevance.

7.3.9 Resources (Computational and Memory) Demand Results

In terms of EM and GP processing times, results from Table 23 and Table 29 show an average of 708 milliseconds for the training and validation process (i.e., for the small BBN in Figure 39) when executed on a computer with 8GB RAM and an Intel (R) Core (TM) i3-6006U CPU @ 2.00GHZ. This could support the DSA system in quick decision-making, especially in a dynamic search area (Kitchin and Baber, 2016; Stanton et al., 2006, 2009b). In contrast, for the large BBN of Figure 40, training with 28,000 datasets takes 9328 seconds. In case of an emergency system, e.g., the proposed use case of forest fire monitoring mission, this could be a delay (Ingle, 2011), as such, huge computational power is needed for larger networks (Scanagatta et al., 2019). Table 30 describes the resource demand analysis.

Table 30: Complexity Assessment

Approach	Time Complexity	Memory Demand	Comments
Expectation-Maximisation algorithm	*O(n x I x r)* where n is the number of observation entries, r is the number of co-states values, and i is the number of iterations.	*O(n x i x m)* where m is the average memory (in bytes) needed to store each BBN tuple entry.	i. The running time grows linearly with the increase in the number of iterations. As such, the application to agents with a lower level of resources needs a few iterations.
Gaussian Process	$O(n^3)$, where n is the number of data entry.	$O(n^2)$	i. Cholesky decomposition can be used to reduce the complexity of the algorithm("A Practical Implementation of Gaussian Process Regression," n.d.).
Time Series Models	$O(n^2m)$ where m is the number of parameters as discussed in Section 7.1.3.	$O(n^2)$	

Therefore, based on the outlined agents' settings (Chapter 1), simple agent has a very limited computational capacity of which resource utilisation is very critical. Similarly, the pictures compilers (PCs) need resource utilisation to certain extent due to cruising, data collection, and communication tasks. The host has abundant computational resources, of which utilisation demand is not as critical as simple agents (micro or mini UAVs) or PCs. Therefore, based on the outcome of Table 30, agent's ability to handle prediction and uncertainty handling depends on its resources e.g., large BBN and datasets training requires large computational demand. Table 31 describes the methods recommendation based on the agents' capacity.

Table 31:Agent-based Approach Recommendation

Agents Type	Recommendation for Handling Uncertainties		
	EM Algorithm	Gaussian Process	Time Series Models
Simple gent	Medium when number of iterations are small	Very low	Very low
Picture compilers	Medium	Low	Very low
Host	High	Medium	Low

7.4 Discussion and Conclusion

This chapter describes different methods of dealing with predictions and uncertainties in DSA system. The chapter categorises the approaches into single state and multiple states predictions/uncertainties handling. The evaluation considers data generated using fixed-pattern or pseudorandom methods i.e., the proposed Delaunay-Inspired Multi-agent Search Strategy (DIMASS) method of Chapter 3 or Lévy flight (Chawla and Duhan, 2018; Sutantyo et al., 2011). Single states predictions/uncertainty handling shows low performance in terms of error rate, whereas multiple states shows higher performance due to consideration of many other states (as described by the results in Table 23 and Table 29). In terms of the search patterns' structure, the fixed pattern shows stable behaviour due to the structure of the search area (as discussed in Chapter 6).

Similarly, prediction and uncertainty handling depends on the amount of data and missing values and their stability, i.e., being linear, as can be seen from the result in Table 25 and Table 26. This results from the Jensen's inequality (Adil Khan et al., 2020) (Equation 26 and Equation 27).

Belief increment:

$$\mathrm{P}\left(\int_{\mathrm{s}=\mathrm{l}_1}^{s=\mathrm{l}_{1+1}} \mathrm{zB}(\mathrm{R_r}|\mathrm{S_s})\right) \mathrm{dS} \leq \int_{s=\mathrm{l}_1}^{\mathrm{s}=\mathrm{l}_{1+1}} \mathrm{zP}\big(\mathrm{B}(\mathrm{R_r}|\mathrm{S_s})\big)\mathrm{dS}$$

Equation 26: EM Expectation State for Convex Entries

Belief decrement:

$$\mathrm{P}\left(\int_{\mathrm{s}=\mathrm{l}_1}^{s=\mathrm{l}_{1+1}} \mathrm{zB}(\mathrm{R_r}|\mathrm{S_s})\right) \mathrm{dS} \geq \int_{\mathrm{s}=\mathrm{l}_1}^{\mathrm{s}=\mathrm{l}_{1+1}} \mathrm{zP}(\mathrm{B}(\mathrm{R_r}|\mathrm{S_s}))\mathrm{dS}$$

Equation 27:EM Expectation State for Concave Entries

where B(Ss) and B(Rr) are the probabilities of the querying and related states of the BBN, and z is the optional weight factor from SMEs. That is the BBN training using EM priorities recent data over old ones. In conclusion, different forms of uncertainties have different behaviours (based on the result in Table 28) and the choice of the approach depends on the DSA system situation. EM algorithm shows good prediction/uncertainty handling with some limitations in terms of a large number of uncertain variables.

8 Chapter 8 Adaptable DSA

Learning the structure of Bayesian Belief Networks (BBNs) from operational data is challenging especially when it involves multiple structures construction and accuracy testing (score-based methods). These methods are computationally demanding and require large training datasets. A viable alternative is to develop a set of protocols to guide the structure development (constraint-based methods). The latter approach is computationally cheap with the main limitation of low accuracy. This chapter introduces a novel constraint-based BBN structure learning algorithm for the DSA system described in Chapters 1 and 4. The method work based on protocols derived from node conditional independence maximum likelihood clustering using the Gaussian Mixture Model (GMM). The aim is to ensure the adaptability of the BBN for DSA systems as described in Chapter 4. Results proved scalability, adaptability, integration of Subject Matter Experts (SMEs) and automation inputs, uncertainty quantification, and computational efficiency.

8.1 Introduction

Bayesian Belief Networks (BBNs) are well-known tools for modelling concepts and their relations in various systems. They were applied in numerous areas, such as medical diagnosis, forensic science, multi-agent system, etc. (Wang and Xu, 2014; Scanagatta et al., 2019). In terms of using BBN for SA modelling (as described in Chapter 4), the v-structure (links configuration) defines search area phenomena (as in the thesis use case) and their relations (i.e., what causes what within the system as discussed in Chapter 4). For example, Figure 41 describes a simple BBN used to understand forest fire spreads at the PC level. This BBN simply represents the concepts of fire spread based on its presence or absence, fuel condition, location relation to the ground, and fuel type.

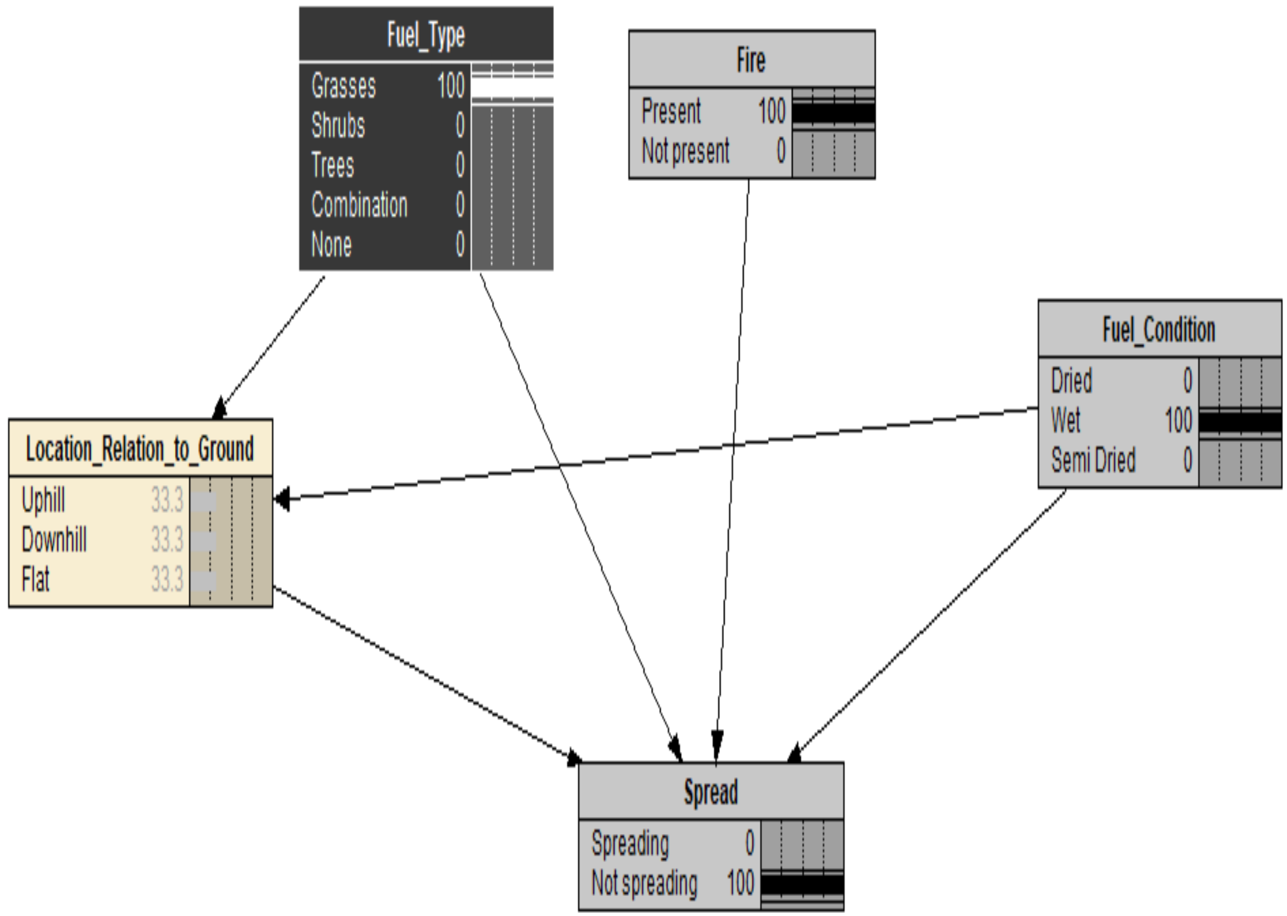


Figure 41:Fire Spread BBN (NETICA Software)

Figure 41 is one of the BBN configurations, perhaps based on the current system information or Standard Operating Procedure (SOP) documentation. The configuration could change based on the newly received information. For example, consider a similar BBN configuration with the absence of a link between the "Location relation to the ground" and "Spread" nodes (assume that it is obtained from a different location). That is, "Location relation to ground" information is independent of the "Spread" node (which means fire spread is independent of the search area terrain relation to ground, i.e., uphill, downhill, or flat). An autonomous (data-driven) or semi-autonomous (which requires certain parameters or hyperparameters definition) structural learning algorithm helps to construct the system BBN model to address the challenge. Thus, this chapter aims to answer the research question "*RQ3, how could agents search plan support SA management?*" focusing on obtaining an adaptable DSA modelling tool based on the agent's search mission information.

Based on the thesis DSA definition (Chapter 1), different agents are tasked to perform different roles in different search area situations. The combination of the output of individual agents leads to the SA management at the system level. Due to the dynamism of the search area, SA modelling in such a system is also a dynamic (Kitchin and Baber, 2017; Stanton et al., 2006, 2009). That is, SA is context-based. For example, from Figure 41, fire detection information interpretation depends on the sensor type and mode of operation. Thus, detecting a fire using a visual camera sensor during night-time could be more reliable than during daytime, when fire-like objects could distort the sensor's reliability. Thus, SA management requires context-based interpretation of the agents' situation (i.e., situation assessment). Therefore, this chapter aims to (i) introduce a constraint-based structural learning algorithm for the proposed BBN for DSA modelling and (ii) describe its application for managing DSA adaptably within the described system.

8.1.1 Existing Methods

BBN structural learning algorithms are generally categorised into: score-based, constraint-based, or hybrid approaches. Score-based methods construct n number of the BBN and select the best performing BBN (in terms of prediction) using testing i.e., testing the candidate network against the prediction of unseen parameters (Bregoli et al., 2021; Karduni et al., 2021; Scanagatta et al., 2019). This version of the algorithm produces a better outcome, but it is super-expensive (i.e., it requires large computational resources) due to a large number of BBN candidates generations.

On the other hand, constraint-based methods construct the BBN model using derived protocols, e.g., based on the Conditional Independence (CI, i.e., nodes degree of relevance) measure (Bari, 2011; Bregoli et al., 2021; Scanagatta et al., 2019; Zhang et al., 2020). The simplest version of the constraint-based structural learning algorithm is the Chow-Liu algorithm which constructs the BBN based on the weighted CI measure (Altarriba and Halonen, 2020; Bhattacharyya et al., 2021; Huang et al., 2003). Further improvement of this algorithm is the PC (Peter and Clark) algorithm which imposes BBN development based on derived protocols using CI measure, i.e., the BBN is first developed and restructured based on inter-nodes CI values (Spirtes et al., 1991; Bouckaert, 1995; Bregoli et al., 2021; Le et al., 2019; Madsen et al., 2017; Marcano et al., 2020; Scanagatta et al., 2019; Scutari, 2015; Scutari et al., 2019; Zhang et al., 2020). In contrast, hybrid methods combine the strategies of score-based and constraint-based methods (Bregoli et al., 2021; Le et al., 2019; Li et al., 2019; Scanagatta et al., 2019; Scutari, 2015; Zhang et al., 2020). That is, candidates' networks are generated, and protocols are developed to guide during best network selection with no assuarance of a better performance (Scutari, 2015). This thesis contributes to the family of the constraint-based algorithm by developing an algorithm that clusters the CI and nodes' goal-based weight (i.e., to reflect a DSA system) using Gaussian Mixture Model (GMM). The outcome fits the use case and proves resource utilisation, scalability, and adaptability. Additionally, the developed protocols consider various DSA system features such as the SMEs' merging, agents' goals variation, number of inputs measurement, and partial updates.

8.1.2 BBN Structural Learning for DSA System

Figure 41 describes an example of a BBN structure for forest fire monitoring. The probability of the parent node's states is defined jointly by the combination of dependent nodes' states entries. For example, the probability of the state of the node "Spread" from Figure 41 could be derived from "Fire", "Location relation to the ground", "Fuel type", and "Fuel condition" nodes' states. The combination of each dependent node's states probabilities determines the parent node entry, which is maintained by the parent node's Conditional Probability Table (CPT) as described in Chapter 4. For instance, Table 32 describes an example of CPT entry for the spread node.

Table 32: Example of CPT for Spread Node of Figure 41

#	Fuel Type	Fire	Location Relation to Ground	Fuel Condition	Spread	
					Spreading	Not_spreading
1	Grasses	Present	Uphill	Dried	95%	5%
2	Grasses	present	Uphill	Wet	70%	30%
3	Grasses	Present	Uphill	Semi_dried	80%	20%
4	Grasses	Present	Downhill	Dried	85%	15%
…	…	…	…	…	…	…
89	None	Not_present	Flat	Semi_dried	0%	100%

From Table 32, the probability measure of the dependent nodes' states determines the CPT value of the parent node. For example, #1 shows that the fuel type is grasses and in a dried condition, while fire is present on uphill terrain. This leads to a 95% belief of fire spreading (a rapid spread). The spreading belief is different when the fuel type is wet or semi-dried as in #2 and #3, respectively. Therefore, the CPT probability value quantifies the degree of the node's states beliefs. Overall, all states of the BBN and their probablities describe the DSA of the search area.

Similar to the BBN priors' initialisation, the edges configuration of the BBN can be initialised based on the domain SOP or learnt from agents' previous mission data. Different algorithms were proposed for the latter case. This can be categorised generally into score-based, constraints-based, and hybrid methods (Bari, 2011; Park et al., 2013; Scanagatta et al., 2019; Zhang et al., 2020).

As partially discussed above, score-based algorithms construct n number of candidates networks and then test each network's accuracy in predicting unseen data (i.e., the likelihood P(D|N) where D is the data and N is the network). The best performing candidate network (identified using low error rate) will be recognised as the most likely network. This approach has been enhanced using different strategies such as Tabu search (by avoiding previous poor-performing candidates), probabilistic Tabu search version (Bouckaert, 1995), genetic algorithms (Larranaga et al., 1997) etc. Due to the high number of candidates generations, the score-based method requires enormous computational power, although it produces more accurate results (Scutari et al., 2019).

In contrast, constraint-based strategies utilise statistically derived protocols (e.g., entropy measure, Pearson correlation, etc.) to depict relating nodes (Le et al., 2019; Li et al., 2019; Scanagatta et al., 2019; Scutari et al., 2019). For example, if the entropy measure between two nodes V_i and V_j is greater than the one between V_j and V_i, there will be a causal link from V_i to V_j, i.e., $V_i \rightarrow V_j$. Constraint-based algorithms are faster with little computational demands (Scanagatta et al., 2019). This has been improved using Markov blanket (Qi et al., 2021; Scutari, 2015), Chow-Liu tree (Huang et al., 2003), and developed heuristics (Qi et al., 2021; Scanagatta et al., 2019). However, deriving effective constraints protocols is challenging (Scanagatta et al., 2019; Scutari et al., 2019). The combination of score-based and constraint-based methods leads to a hybrid approach (Scutari et al., 2019).

In this chapter, I propose a constraint-based method that clusters (classify) the degree of dependency (using conditional independence) and inter-nodes experts assigned weights (obtained from SMEs weighting as described in Chapter 4 or SOP documentation). The clustering process considers the maximum likelihood probability P(D|N) using the Gaussian Mixture Model (Banfield and Raftery, 1993), where D is the data and N is the BBN structure (candidate network). The maximum likelihood selection is to produce a highly expected network. That is a network with a high chance of having a higher prediction score. Thus, the proposed approach's target is to inherit the best features of the score-based and constraint-based methods in combination with DSA system formalisation. The constraints protocols were based on Reichenbach's Common Cause Principle (RCCP) (Hitchcock and Rédei, 2020), i.e., if $P(V_i, V_j) > P(V_i) \times P(V_j)$ then the nodes V_i and V_j are inter-related. The proposed algorithm depicts the level of dependency by classifying parents, children, grandchildren, etc., based on the maximum likelihood of the conditional independence and inter-nodes SMEs weights (from Chapter 4) using GMM. The outcome demonstrates how the proposed algorithms could help in maintaining the system's DSA.

8.1.3 Situations and Agents Variation across DSA System and the Need for an Adaptable DSA Model

As outlined in Chapter 1, the simple agents (exploring agents) perform the system information acquisition tasks. The simple agents will be submitting their individual information to their respective picture compilers. The picture compilers (PCs) will then report their information to the managing host interacting with the SMEs. Therefore, the information follows from the simple agents to PCs, host and finally, the SMEs. Similarly, the command from SMEs follows in an opposing direction. Thus, SA varies across agents' levels. For instance, SA to the simple agent means performing the sensor poll and navigating effectively (e.g., avoiding repetitive search, collision, etc.). For the PCs, SA means managing the collected simple agents' sensor information (e.g., using BBNs), sensor information conflict management, and situation understanding. The host maintains SA by managing multiple PCs' SAs, uncertainty handling (missing information management), sensor conflicts resolutions, and SMEs commands integration. The human experts analyse the presented SA model logically and make decisions, e.g., where to send evacuation vehicles, where to start fighting the fire, etc. Hence, SA definition within the system is based on the agent's level.

Considering the changing nature of the search area, SA across various agents' levels needs to be modelled in an adaptable manner. Thus, the structure of the BBN controlling the SA model of Chapter 4 (the proposed BBN) needs to be adaptable (i.e., responding to the current fed information in real-time). This can be achieved through instant priors updated based on the agents' sensors information (as described in Chapter 4 Section 4.8) and the node's relation management.

8.1.4 Conditional Independence Measures

The probability of how an input at a particular node, say, V_i, could affect the probabilities of another node, say, V_j can be measured using conditional independence (CI) metrics. Different strategies were proposed to compute those metrics, such as the Pearson correlation (Karduni et al., 2021; Kumar, 2007), Shannon entropy measure (Shannon, 1959), probability variance, etc., (Karduni et al., 2021; Kitchin and Baber, 2017; Neapolitan, 1990; Pearl, 1988). The choice of a particular measuring technique depends on the application domain. For example, the variance measure provides a value ranging between 0 and 1, while the entropy measure is not. The normalisation process is required for every set of values while using an entropy measure (Equation 28). Although the number of entropies conversion to the normalised value of 1 could be cumbersome, the metric measures uncertainty (Scanagatta et al., 2019; Skotarczak et al., 2018) and the number of data needed (as discussed in Section 8.5).

$$Entropy_{\lambda(S_s|R_r)} = -\sum_{s=1}^{s=n}\sum_{r=1}^{r=n}\frac{P(S_s,R_r)Log_x P(S_s,R_r)}{P(S_s)P(R_r)}$$

Equation 28:Nodes Relation Measurement using Entropy

where $P(S_s)$ and $P(R_r)$ are the probability of the querying and related nodes with their respective states sets s, and r, $P(S_s,R_r)$ is their joint probability, and x is the base for the logarithm, which determine the measuring units, e.g., Shannons if x=10 or bits if x=2 i.e., similar to Equation 28 of Chapter 5.

The entropy measure quantifies the level of dependencies among the nodes. For example, assume that a fire detecting UAV reports “present”, and fuel condition detecting UAV reports “dried”. Therefore, based on the BBN states’ priors updating algorithm described in Chapter 4 Section 4.8, the CI measure will be:

P(Fire=Present) = 0.75, P(Fire=Absent) = 0.25, P(Fuel Condition = Dried) = 0.67, P(Fuel Condition = Wet) = 0.165, and P(Fuel Condition = Semi_Dried) = 0.165.

The entropy measure between "Fire" and "Fuel Condition" nodes will be :$\lambda\left(Fire_s|Fuel\ Condition_f\right) = -\sum_{s=1}^{s=n}\sum_{f=1}^{f=n}\frac{P\left(Fire_s,Fuel\ Condition_f\right)log_2P(Fire_s,Fuel\ Condition_f)}{P(Fire_s)P(Fuel\ Condition_f)}$

Step 1: Expanding the second summation

$$\lambda\left(Fire_s|Fuel\ Condition_f\right) = -\sum_{s=1}^{s=n}[(\frac{P(Fire_s,Fuel\ Condition=Dried)log_2P(Fire_s,Fuel\ Condition=Dried)}{P(Fire_s)P(Fuel\ Condition=Dried)}) + \frac{(P(Fire_s,Fuel\ Condition=Wet)log_2P(Fire_s,Fuel\ Condition=Wet)}{P(Fire_s)P(Fuel\ Condition=Wet)}) + (\frac{P(Fire_s,Fuel\ Condition=Semi\ Dried)log_2P(Fire_s,Fuel\ Condition=Semi\_Dried)}{P(Fire_s)P(Fuel\ Condition=Semi\ Dried)})]$$

Expanding the first summation

$$\lambda\left(Fire_s|Fuel\ Condition_f\right) = [(\frac{P(Fire=Present,\ Fuel\ Condition=Dried)log_2P(Fire=Present,Fuel\ Condition=Dried)}{P(Fire=Present)P(Fuel\ Condition=Dried)}) + \frac{(P(Fire=Present,Fuel\ Condition=Wet)log_2P(Fire=Present,Fuel\ Condition=Wet)}{P(Fire=Present)P(Fuel\ Condition=Wet)}) + (\frac{P(Fire=Present,Fuel\ Condition=Semi\ Dried)log_2P(Fire=Present,Fuel\ Condition=Semi\_Dried)}{P(Fire=Present)P(Fuel\ Condition=Semi\ Dried)})] + [(\frac{P(Fire=Absent,\ Fuel\ Condition=Dried)log_2P(Fire=Absent,Fuel\ Condition=Dried)}{P(Fire=Absent)P(Fuel\ Condition=Dried)}) + \frac{(P(Fire=Absent,Fuel\ Condition=Wet)log_2P(Fire=Absent,Fuel\ Condition=Wet)}{P(Fire=Absent)P(Fuel\ Condition=Wet)}) + (\frac{P(Fire=Absent,Fuel\ Condition=Semi\ Dried)log_2P(Fire=Absent,Fuel\ Condition=Semi\_Dried)}{P(Fire=Absent)P(Fuel\ Condition=Semi\ Dried)})]$$

Substituting the values leads to

$\lambda(Fire_s|Fuel\ Condition_f) =$ [(0.75x0.67log$_2$(0.75x0.67)/0.75x0.67))+ (0.75x0.165)log$_2$(0.75x0.165)/0.75x0.165 + (0.75x0.165)log$_2$(0.75x0.165)/0.75x0.165] + [(0.25x0.67log$_2$(0.25x0.67)/0.25x0.67))+ (0.25x0.165)log$_2$(0.25x0.165)/0.25x0.165 + (0.25x0.165)log$_2$(0.25x0.165)/0.25x0.165]

$$\lambda(Fire_s|Fuel\ Condition_f) = 5.42bits$$

That is, the expected reduction in entropy of the "Fuel condition" node given a finding at the "Fire" node is 5.42bits. Note that, the lower the entropy value the higher the relations.

Summarily, the choice of the statistical criteria for measuring the conditional dependency of the nodes depends on the application domain. For example, if the number of agents interaction, number of datasets needed, and uncertainty in entropy reduction are needs to be monitored, then entropy (Equation 28) is a good choice (this will be discussed in detail with a supported result in Section 8.5). However, when computational demands need to be utilised (e.g., UAV's onboard processing by avoiding conversions), then probability variance (Chapter 4 Equation 8) or Pearson's correlation could be used to avoid multiple normalisation of values (i.e., to 1) computations. The proposed algorithm utilises the CI to separate the BBN nodes relations, e.g., parents, children, grandchildren, etc, of the BBN graph.

8.2 Hypotheses

It is hypothesised that the experiment would show that

i. The adaptability of the proposed structural learning algorithm depends on the fed data and priors update algorithms.

ii. The proposed solution generates the candidate's BBN based on the maximum likelihood priority (i.e., most probable candidate for the score-based methods).

8.3 Performance Metrics

Performance measures were based on the following:

i. Adaptability: the ability of the approach to construct appropriate BBN based on the agents' fed data. This is measured based on how the BBN changes given different datasets.
ii. Processing time (i.e., time needed to run the algorithm). This is measured as the time taken by the algorithm to produce a result (i.e., BBN output).
iii. Scalability: the ability to handle n number of nodes and m datasets with a stable processing time. This is measured using the processing time and the number of nodes.
iv. SMEs-automation inputs integration (feature). This is identified as the ability to integrate both automation and SMEs inputs (i.e., it is a feature metric).

8.4 Proposed Solution

The proposed algorithm utilises the conditional independence measure (Section 8.1.4) with or without the SMEs edges (BBN causal links) critical weight assignment (as described in Chapter 4 Section 4.7) to impose the learning constraints (Section 8.4.1). Maximum likelihood clustering using the Gaussian Mixture Model (GMM) was used to separate the BBN links hierarchy, i.e., separate parents, children, grandchildren nodes etc.

8.4.1 Constraints (Protocols) Development

The algorithm's protocols were based on Reichenbach's Common Cause Principle (RCCP)(Cartwright, 1988; Feyerabend, 1959; Hausman and Woodward, 1999). Summarily, RCCP states that if the joint probability of two nodes is more than their individual probabilities' product, i.e., $P(V_i,V_j)>P(V_i) \times P(V_j)$, then V_i is causing V_j, or V_j is causing V_i, or both V_i and V_j are both caused by another node say, V_k (Hitchcock and Rédei, 2020).The proposed algorithm outcome is formed by considering the fact that the higher the CI measure and assigned SMEs critical weight (Chapter 4 Section 4.7), the stronger the link, and this indicates the causality direction. Therefore, the algorithm constructs the BBN structure based on the following protocols.

i. Protocol 1: for all pairs of nodes (V_i, V_j) such that $i \neq j$, if $\lambda(V_i|V_j) > \lambda(V_j|V_i)$, then there will be a link from V_i to V_j (i.e., $V_i \rightarrow V_j$), i.e., V_j is dependent on V_i subjected to the CI and critical weight clustering (protocol 3).

Based on the relevance computation in Section 8.1.4, e.g., using entropy, probability variance, or any method, if $\lambda(V_i|V_j) > \lambda(V_j|V_i)$, then receiving a piece of information for node V_i affects the V_j node. Thus, the Conditional Probability Table (CPT) of V_i, has a higher chance of having an entry from V_j because $P(V_i|V_j)>P(V_j|V_i)$ based on Equation 28. Therefore, whenever $\lambda(V_i|V_j) > \lambda(V_j|V_i)$, there will be a link from V_i to V_j subjected to the CI and the critical weight measure (protocol 3).

ii. Protocol 2: if $(\lambda(V_i|V_j) = \lambda(V_j|V_i)$ or $\lambda(V_i|V_j) = 0$, then the link is void (dead link)

Similarly, based on Equation 28, if $P(V_i|V_j) = P(V_j|V_i)$, findings at the V_i node have the same effect on probability at V_j, thus no causal relation.

iii. Protocol 3: maximum likelihood categorisation of the CI values and critical weight segments causality strength. The clustering function α, e.g., GMM assign each link e_i to a cluster C_i, i.e., $\alpha : e_i \rightarrow C_i \ \forall \ e_i \in E$. Each link belongs to a cluster, and clusters are ranked based on CI measures.

For the clustering task, the chapter utilises the Gaussian Mixture Model (GMM) (Xuan et al., 2001; Zivkovic, 2004) based on the assumption that the BBN states priors are Gaussian from the law of large Number and the Central Limit Theorem (Etemadi, 1981; Hsu and Robbins, 1947). Again, GMM is proven to provide an outstanding clustering better than other approaches such as the k-means (Muñoz et al., 2021; Murray and Perera, 2022; Zhang et al., 2015; Zivkovic, 2004). The GMM is defined by the function $\alpha(e_i) = \sim N(\lambda(e_i)|\mu,\sigma)$ where $N(\lambda(e_i)|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e(-\frac{(\lambda(e_i)-\mu)^2}{2\sigma^2})$, $\lambda(e_i)$ is the CI value for the edge (link) e_i, μ and σ are the mean and standard deviation of the links CI values distribution. Whenever critical weight (w) is used, the variance σ is replaced with a covariance matrix $\Sigma_{\lambda w}$ for all clusters. If there exists an already assigned link e_i, the algorithm mediates to one of the descendants' nodes using protocol (iv).

iv. Protocol 4: mediation node selection is based on conditional independence measure, critical weight, popularity (number of indegree links) and authority (number of outdegree nodes) in order of preference and number of values.

Proof.

Relevancy and critical strength were proved in protocols i, ii, and iii above.

However, based on the conditional probability $P(V_i|V_k) = P(V_k|V_i)P(V_i)/P(V_k)$, where k =1,2,3,…n, and V_k is the number of dependent nodes (indegree node). Thus, the higher the popularity, the larger the $P(V_i|V_k)$ and CI value $\lambda(V_i|V_k)$. An interesting question to ask is what is the trade-off between indegree and outdegree assignment? For example, considering Figure 42, assuming “H” is part of the BBN, where does the mediated node ‘H” best fit? The condition is that, its link is not as strong as A→B and A→C based on the clustering process. That is, will the "H" node best fit B or C?

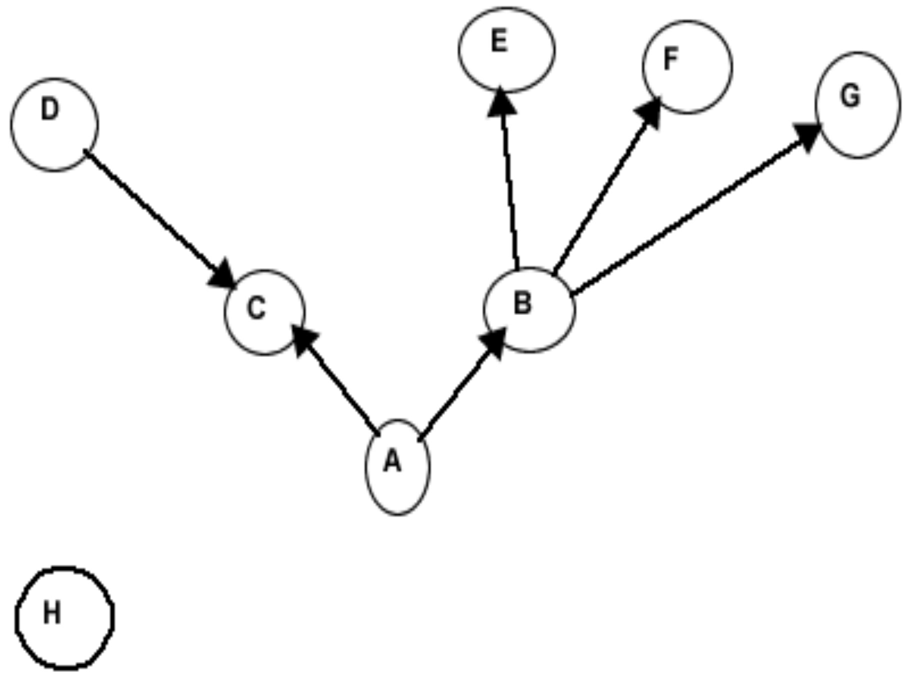


Figure 42: Degree of Transition Role

The answer is that the mediation depends on the provision of protocol (iv). That is, if $\lambda(H|C) > \lambda(H|B)$, then H will be attached to C, and vice versa. When $\lambda(H|C) = \lambda(H|B)$, a low number of indegree or outdegree of the nodes will be used, because the higher the number of related nodes (indegree or outdegree), the lower the chance of conditional probability being low, e.g., it is most probable that $P(B|E,F,G)<P(C|D)\ \forall\ P(E,F,G)>0$, although, an exception could exist when E,F,G>D. Note that, from Bayes protocol, $P(E,F,G)= P(E)P(E|F)P(G|E,F)$.

v. Protocol 5: reversed (links in the opposite direction) are removed based on CI and critical weight strength i.e., based on protocol (iv).

vi. Protocol 6: network construction is cluster-based. i.e., the strongest link has to be considered first with higher priority.

Proof

Protocol (i)

vii. Protocol 7: super-powered link (existing link but overridden by a strong link): protocol (i) will be applied based on the CI or critical weight values.

The outlined protocols (i) to (vii) will be used to generate the structure of the BBN. Section 8.4.2 described the method for incorporating SMEs entries.

8.4.2 SME Inputs Incorporation

Following RCCP and protocols (i) to (vii) of Section 8.4.1 alone could be inconsistent with the search area phenomena behaviours. For instance, many authors have questioned the validity of the RCCP in some specific exceptional cases. In Cartwright's factory example of (Cartwright, 1988 p108–109), stated "...if a factory releases P amount of pollution after 'S' operation, and produce a chemical C, then $P(C,P|S)>P(C|S)P(P|S)$ and in reality neither C nor P causes each other". Many other objections can be found in (Salmon, 1984; Schurz, 2017). Therefore, as suggested by (Glymour, 1999), the generalisation of RCCP to describe real-world situations requires more than a hypothetical description. Thus, the chapter offers a process termed as exceptional cases identification for specifying the exceptions in the DSA model.

The exceptional cases identification method simply means the specification of unrealistic dependent variables and mutually exclusive events using joint or conditional probabilities of the nodes monitored through the BBN CPTs. For example, from Figure 41, we can see a number of RCCP putative cases. Fire will not be spreading while it is absent, or fire will not spread while there is no fuel (i.e., based on the knowledge of SOP), although the agent prior update of Chapter 4 Section 4.8 could settle this issue. Thus, these exceptions within a DSA system can be specified using the CPTs of the nodes. For example, P(Fire=absent |Spread=spreading) = P(Fire =absent) instead of P(Fire=absent |Spread=spreading) = P(Fire=absent) P(Spread=spreading)/ P(Spread=spreading) i.e., because fire spreading depends on its presence. For mutually exclusive events e.g., the joint probability of the nodes can be used. For example, P(Fuel Type = none, Spread = spreading) = 0. Thus, regarding DSA specification, the exceptional cases identification method could help during SMEs SOP incorporation to control the structural learning process. The SMEs entries can be collected using the Thurstone's paired comparison (Allen, 1994) method described in Chapter 4 Section 4.7.

8.4.3 The Proposed Algorithm

The proposed algorithm uses protocols (i) to (vii) derived from Section 8.4.1 to develop a BBN structure representing the agents SA model across various situations. Algorithm 3 describes the proposed structural learning algorithm.

Algorithm 3. The Proposed BBN Structural Learning Algorithm

```
Input: Agents data, nodes critical weight, CI values for each pair of nodes λ(Vi|Vj) ∀ i≠j, and configuration matrix M[].
Output: The directed BBN graph G(V,E,M)
Initialise the number of edges E = 0
```

Initialize the configuration matrix e.g., adjacency matrix M= []

For each pair of node V_i,V_j

Compute $\lambda(V_i|V_j)$

If $\lambda(V_i|V_j) > \lambda(V_j|V_i)$ // protocol (i)

Add the link e_i, such that, e_i: $[V_i \rightarrow V_j] \rightarrow 1$ to the configuration matrix $M[e_i]$

Else if $\lambda(V_i|V_j) <= \lambda(V_j|V_i)$ // protocol (ii)

Mark the link $V_i \rightarrow V_j$ as unsuccessful, add e_i: $[V_i \rightarrow V_j] \rightarrow 0$ to the configuration matrix $M[e_i]$

Merge relevance $\lambda(V_i|V_j)$ for successful links with their corresponding experts' critical weight W_i if any

12	Select the optimal number of clusters, e.g., using the elbow method
13	Identify the number of clusters C = 1,2,3,...n
14	Cluster using Gaussian Mixture Model and obtain the cluster set C = $\{c_1, c_2, c_3, ..., c_n\}$, $c_i = \{e_1,e_2,e_3,...,e_n\}$ and $i \in [1,N]$ // that is each cluster contain set of related links, and link is defined jointly by the relevance value and critical weight i.e., $e_i{:}\lambda_i \times w_i$.
15	If there exists a link $e_i{:}Vi \rightarrow Vj$, where i,j, $\in C_i$, any link attempt, say, $e_J{:}Vi \rightarrow Vm$, where i $\in C_j$, and j<i, the, e_J will be linked to one of the descendants of V_i using the mediation protocol (iv) // a link from a lower cluster will be attached to a descendent of the higher cluster node using

(i) relevance measure $\lambda(V_k|V_{i...n})$ priority or combined with a critical weight measure

(ii) popularity (indegree) measure and authority (outdegree) number of degrees// protocol (iv)

If there exists a link e_i:$Vi \rightarrow Vj$, with $\lambda(V_i|V_j)$ = X and a subsequent link e_J:$V_j \rightarrow V_i$, with $\lambda(V_i|V_j)$ = Y, then e_J is super-powered if X>Y, same protocol for super-powering a node //protocol viii

$e_i \rightarrow M[e_i]$ // add A to the list of edges

Return G(V, E, M)

Therefore, the application of Algorithm 3 to the DSA system follows the below steps:

i. Define the conditional independence measuring function
ii. Assign the critical weight, if any

iii. Select the optimal number of clusters algorithm, e.g., elbow method, average silhouette method, gap statistic method etc.

iv. Apply Gaussian Mixture Model for the clustering task

v. Apply the derived protocols of Section 8.4.1

vi. Apply exceptional cases identification if available Section 8.4.2 (optional)

vii. Generate the network

Based on the controlling protocols of Section 8.4.1, the following proposition holds:

Proposition 2: *A BBN generated using Algorithm 3 receives maximum likelihood in terms of prediction compared to other randomly generated candidates of the score-based strategies.*

Proof:

From the derived protocols, a link from node S to R is feasible if and only if $\lambda(S_s \mid R_r) > \lambda(R_r \mid S_s)$*, i.e., protocol 1.*

Both $Log(P(R|S))+Log(P(S))$ *and* $\lambda(S_s \mid R_r)$ *are maximal if and only if* $P(R|S)$ *is maximum. Since Algorithm 3 is always looking for the strongest link using GMM, then* $P(S|R)$ *is always maximum. Note that,* $P(S|R) = P(R|S)P(S)/P(R)$ *in* Equation 28. *The optimal number of clusters (e.g., Elbow) gives the optimal number of hierarchies based on* $\lambda(S_s \mid R_r)$ *and weighting method strength. Thus, using the Gaussian Mixture Model (GMM) also gives the appropriate link hierarchy with maximised* $P(S|R)$*. Summarily, the proposition is stating that a BBN generated using the proposed algorithm (Algorithm 3) is likely to be among the best-performing candidate of the score-based approach.*

Therefore, we can conclude that the higher the number of strong nodes, the higher possibility of having BBN with good prediction (i.e., reduced error rate). For example, a network with the 10 most powerful nodes could have a higher prediction confidence (prediction score) than the one with 5 nodes.

8.4.4 Algorithm Application Example

The proposed algorithm's demonstration follows the implementation of the AMASE experiment in all of the thesis chapters (i.e., the multi-UAV mission for forest fire monitoring). From Figure 42, fire has been spotted at two locations (the polygons in the top right as in Chapter 3). The task of the UAVs is to explore the environment (using the searching algorithm described in Chapter 3) and construct the SA model using BBN. The UAVs utilise the proposed Delaunay-Inspired Multi-agent Search Strategy (DIMASS) algorithm (Chapter 3) for search plan generation.

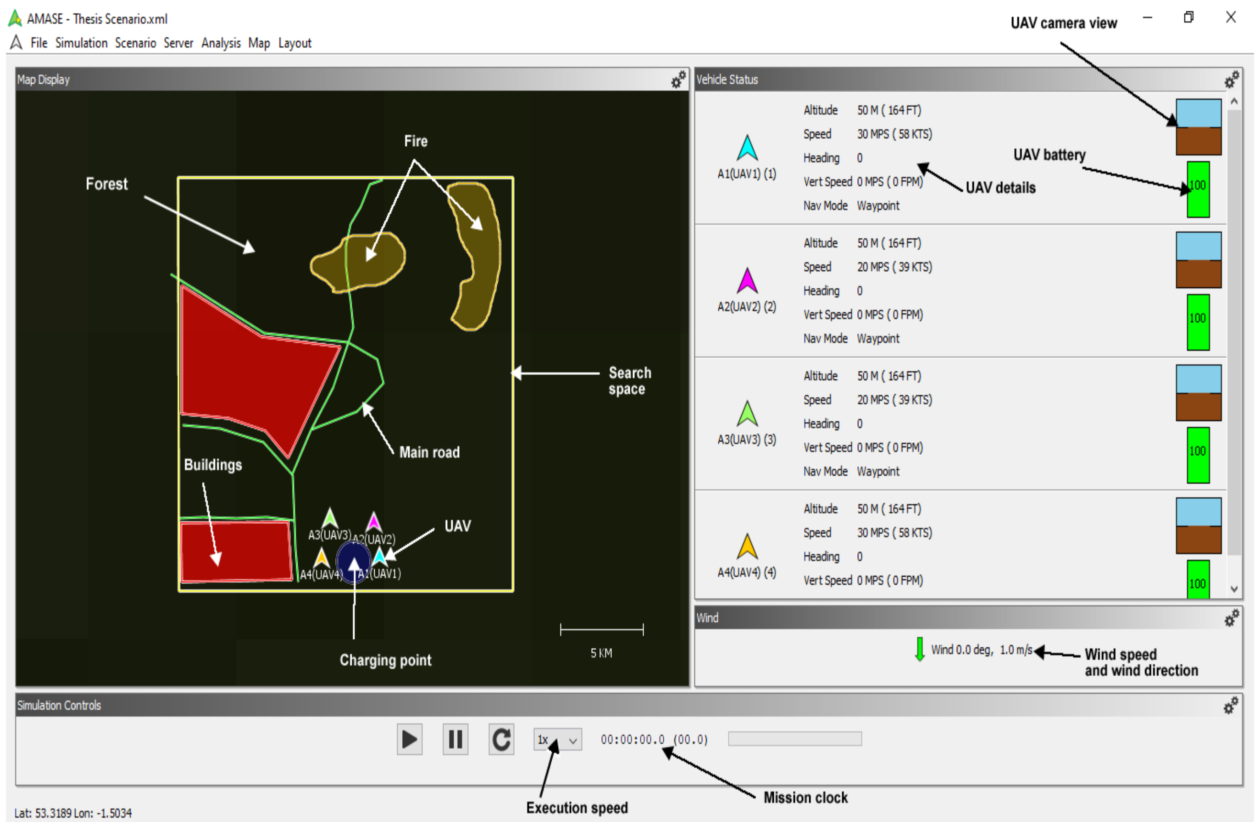


Figure 43:AMASE simulation

The PCs or hosts will generate the probability priors using the algorithm described in Chapter 4 Section 4.8 after every information reception from the simple agents. In this chapter, I assume the SME's critical value described in Table 33.

The priors are derived from the UAVs sensor data (i.e., as described in Chapter 4 Section 4.8). Therefore, the network structure construction using Algorithm 3 entirely depends on the prior's current values. Table 33 describes priors for 100,000 entries from different simple agents at the fire spread monitoring PC level.

Table 33: Example of Prior Probability for States of the BBN

Fire Node	P(present)= 0.51	P(absent)=0.49			
Spread	P(spread)=0.15	P(not_spreading)=0.85			
Fuel Condition	P(dried)=0.46	P(wet)= 0.41	P(semi_dried)= 0.13		
Fuel Type	P(shrubs)= 0.11	P(trees)= 0.05	P(Combination)= 0.70	P(grasses)= 0.14	P(none)= 0
Location Relation to Ground	P(uphill)= 0.37	P(downhill)= 0.31	P(flat)= 0.31		

The next step is to compute the relevance measure among nodes. This will allow us to depict the successful links. The successful links will then be mapped with the critical weight and apply the optimal number of clusters algorithm to get the best network hierarchy. The chapter selects the elbow method (Thorndike, 1953) as the optimal number of clusters finding algorithm due to its popularity. The elbow method algorithm cluster the inputs by computing the inertia (inter-cluster distance) and select the point where number of cluster has no or little effect on the inertia. For example, Figure 44 describes the elbow method outcome for the entries in Table 34. The relevance values of Table 34 were obtained using the states priors as described in Section 8.1.4. The critical weight are assumed SME values based on the forest fire SOP (as learnt from the physical experiment of Chapter 6 and documented reports). For exmple, from #1 of Table 34, the "Fuel Type" node is firmly related to the "Fire" node i.e., fire is related to fuel type, as such the value of 0.9 is assign to it. Note that, the elbow number of clusters and GMM execution takes in the .csv version of Table 34 and run the Python functions for the respective algorithms (the supplemental document folder contains an example of the source codes).

Table 34: Relevance and Critical Weight Values

Link	**Type of Link**	**Relevance Value (λ)**	**Normalised Relevance Value(λ/λ_{max})**	**Critical Weight**
Fuel Type to Fire	Successful	48.58	**0.60**	**0.9**
Fire to Location Relation to Ground	successful	10.77	**0.13**	**0.6**
Fuel Type to Spread	successful	51.84	**0.64**	**0.9**
Fuel Type to Fuel Condition	successful	81.05	**1**	**0.2**
Fuel Type to Location Relation to Ground	successful	78.99	**0.96**	**0**
Fuel Condition to Location Relation to Ground	successful	21.07	**0.26**	**0.1**
Fire to Spread	dead	6.85	**0.08**	**1**

Fire to Fuel Condition	dead	11.59	**0.14**	**0.7**
Spread to Fuel Condition	dead	13.55	**0.17**	**0.8**
Spread to Location Realisation to Ground	dead	12.73	**0.16**	**0.7**

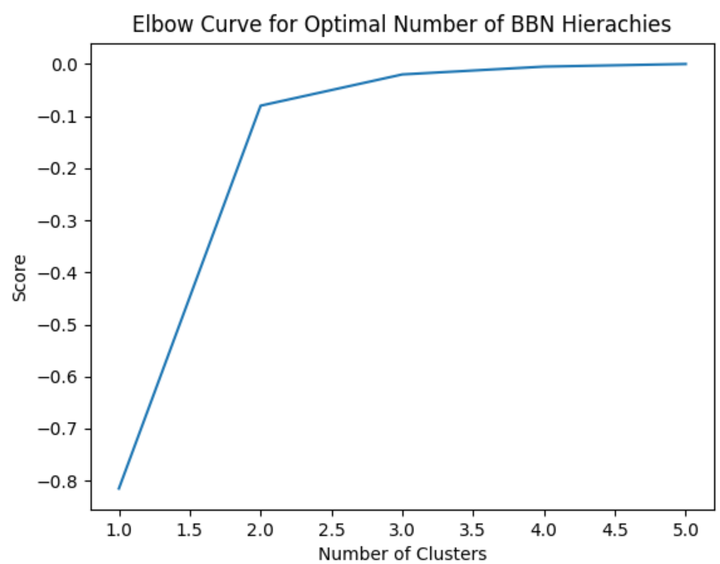


Figure 44:Elbow method number of BBN links hierarchy

From Figure 44, the optimal number of clusters is 3. Therefore, the Gaussian Mixture Model (GMM) would receive the number of clusters and assign each successful link to a cluster as described in Figure 45.

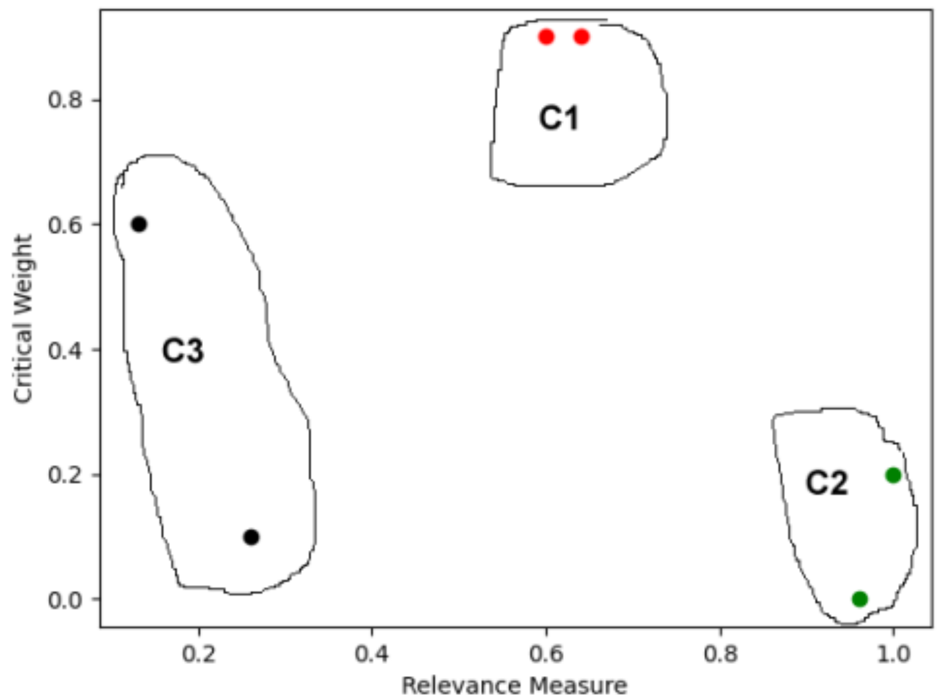


Figure 45: Elbow method number of BBN links hierarchy

The GMM outcome (Figure 45) categorises the links for each cluster (C1, C2, and C3). Figure 46 describes the produced BBN model based on Algorithm 1. The links from "Fuel Type" to "Spread" and "Fire" nodes (from C1) were the strongest links (in terms of CI and critical weight). Note that, in the absence of critical weight, univariate clustering can be used perhaps by assigning 0 as critical weights to each link). The links from “Spread” nodes to “Fuel Condition” and “Location Relation to Ground” were mediated respectively from the fuel type based on their relevance measure of Table 34 (although some of the links were dead, but they will be used for mediation and super-power nodes purposes).

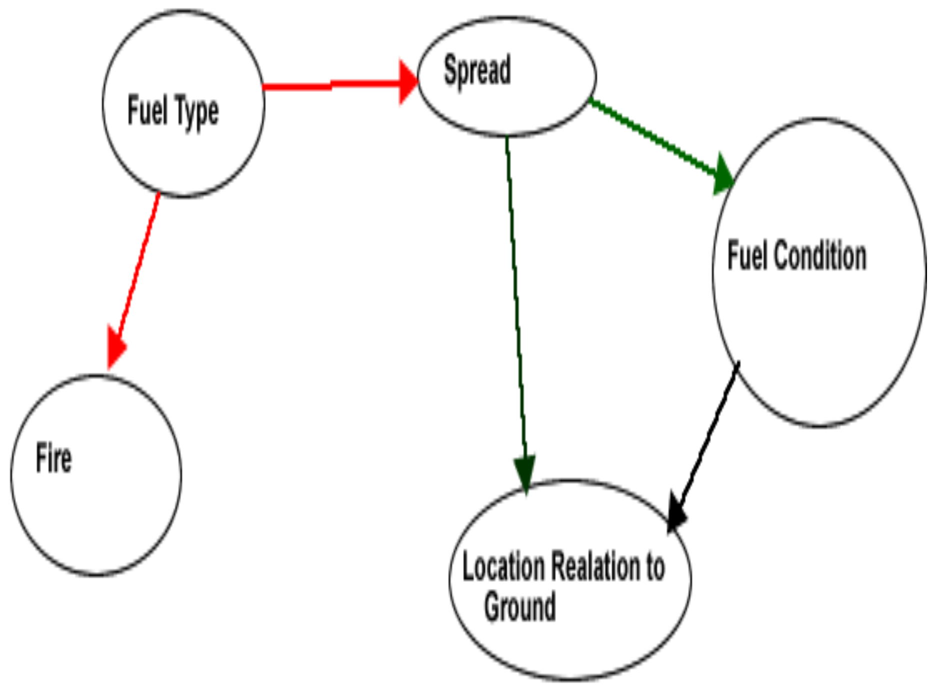


Figure 46: The Constructed BBN

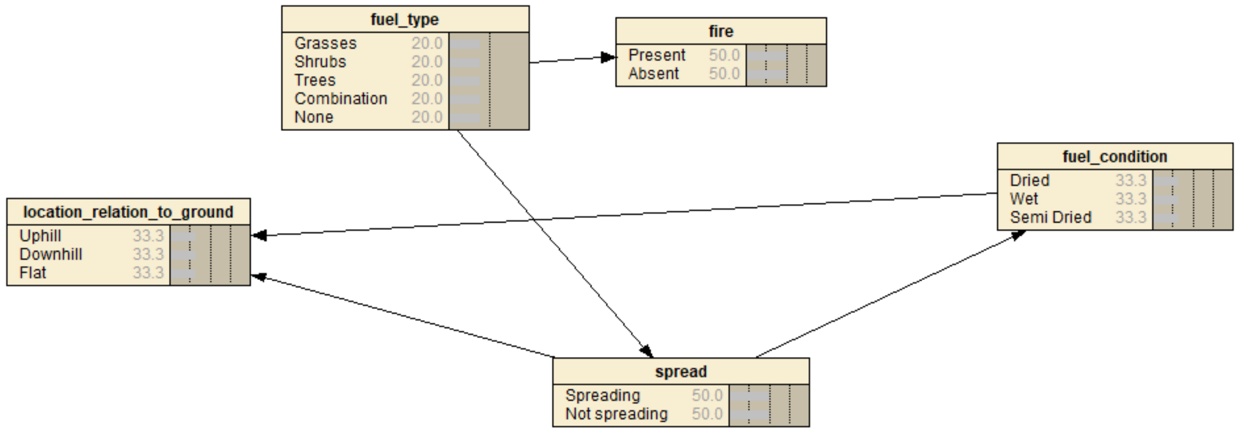


Figure 47: The Constructed BBN (NETICA SOFTWARE)

Figure 46 and Figure 47 (NETICA version) describe the versions of Figure 41 BBN generated using Algorithm 3 and 100,000 of mission data. The links from "Fuel Type" to "Spread" and "Fire" nodes were the strongest links (at the top of the clusters (i.e., C1), i.e., strong in terms of CI and critical weights from Figure 45. The next links are the links from "Fuel type" to "Fuel Condition" and "Location Relation to Ground" nodes (from C2 of Figure 45 i.e., cluster 2). However, based on protocols (i) and (iii), the links were not possible from "Fuel Type". That is, they have to be mediated via the children of the "Fuel Type" node because they are from a lower cluster, C2, which is less than C1. Based on protocols (i) and (iv), the mediation goes to the "Spread" node based on its CI measure Table 34 i.e., λ(Spread|Fuel Condition) and λ(Spread|Location Relation to Ground) > λ(Fire|Fuel Condition) and λ(Fire|Location Relation to Ground). The final links (from C3 i.e., the weakest ones) are the links from "Fire" to "Location Relation to Ground" and "Fuel" Condition to "Location Relation to Ground". The link "Fire" to "Location Relation to Ground" is not possible because the cluster power is weak (i.e., protocol 3). Thus, the link has to be mediated. The best mediating node is "Spread" to the "Location Relation to Ground" node which is super-powered (protocol vii) by an existing stronger link (from a stronger cluster C2).

The BBN generation process in Figure 46 did not consider the exceptional cases identification method. Therefore, a different result could be generated by imposing the exceptional cases identification method of Section 8.4.2. For effective management of DSA, the BBN will be suggested to the SMEs as a SA presentation model, in which the experts could decide on its acceptance, rejection, or modification (using exceptional cases identification and critical weights adjustment).

Uncertainty and the number of data needed for the learning process can be measured based on the number of successful links generated. For example, using ten (10) pieces of data, the successful links of Table 34 are only 2 in number. This means there will be unconnected nodes, which signifies the need for more data or partial updates (i.e., few links update only).

8.5 Algorithm Evaluation

To evaluate the scalability of Algorithm 3, I simulated 1000 and 2000 successful links (i.e., obtained from at least 2000 and 4000 nodes, respectively) and critical weights generated randomly to break the record of the existing 200 nodes (Zhang et al., 2020). Each node was assumed to have one million number of states. The algorithm was run on a computer with 8GB RAM and an Intel (R) Core (TM) i3-6006U CPU @ 2.00GHZ. The evaluation process starts by clustering the relevance values and critical weights of the successful links (note that the computational time iterating through the number of states was considered in the result). Table 35 describes the results for the optimal number of clusters process. Note that, the processing time is for both GMM and elbow process.

Table 35: Experiment Results

States for each node (variables)	Successful links	Elbow Optimal Number of Hierarchy	Processing Time (seconds)
1000000	1000	15	3914.21
1000000	2000	16	20263.73

Based on observation, the algorithm spends most of its time computing the optimal number of clusters (i.e., inertia for each cluster). Therefore, I propose the use of Lemma 1 to estimate the range of the cluster.

Lemma 1: given the normalised 0 to 1 values of standard deviation (σ_i and σ_j) and mean (μ_i and μ_j) of the relevance value and critical weight, and μ_i,μ_j,σ_i, and $\sigma_j \neq 0$ i.e., the provided values are non-zeros.. The estimated number of clusters K is

$$K \leq n||log[(\left||\sigma_i \text{ x } \sigma_j|\right| - ||\mu_i - \mu_j||)]||$$

Equation 29: Number of Clusters Equation

The proof is laterally observational based on the test. Thus, I can say that the lemma works for the tested 2000 number of links.

Although Equation 29 is based on observation and mean/standard deviation relations, it paves the way for estimating the optimal number of clusters without using complex calculations as in (Patil and Baidari, 2019). The data used for Table 35 is describe in Table 36.

Table 36: Successful Links

Number of Link	Relevance Weight Mean (σ_i)	Critical Weight Mean (σ_j)	Relevance Weight Standard Deviation (μ_i)	Critical Weight Standard Deviation (μ_j)
1000	0.508478	0.493945	0.286922	0.288833

2000	0.497447	0.492898	0.287508	0.291099

Note that the values of Table 36 are based on the randomly generated values. Therefore, the estimation for Table 35 will be:

For 1000 links:

K estimate will be 1000(||0.286922 x 0.288833||+||0.508478-0.493945||)

1000(0.082872+0.014533) ≈ 97. Of course, 15 ≤ 97.

For 2000 links:

2000(0.287508 x 0.291099+||0.497447-0.492898||)

2000(0.083693+0.004549) ≈ 176. Of course, 16 ≤ 176. Note that, the estimated values (i.e., 97 and 176 in this case) are passed as parameters during the elbow method computations.

Therefore, by applying Lemma 1, the elbow algorithm will efficiently estimate the number of clusters (Table 37). Instead of iterating 1000 or 2000 times for the total number of links, it will only iterate 97 and 176 times (as estimated) respectively and produce the same result (because the estimated number of clusters is higher than the real value).

Table 37: Reduced Number of Clusters Performance Comparison

Number of States (variables)	Number of Links	Elbow Outcome	Number of Threads	Estimated number	Processing Time(seconds)

				of clusters	
1000000	1000	15	1	97	62.94
1000000	2000	16	1	176	195.88

Therefore, based on Table 37, the application of Lemma 1 reduces the algorithm's running time.

8.5.1 Comparison with the Existing Methods

There are several structural learning algorithms that relied on CI measures. However, they differ from the proposed method in many ways. Table 38 describes the difference and novelties of the proposed method with the existing ones.

Table 38: Comparison with the Existing Methods

Existing Method	The Proposed Method	Novelty of the Propose Method over the existing method
Chow-liu method (Altarriba and Halonen, 2020; Bhattacharyya et al., 2021) generates a CI-weighted edge BBN (i.e., similar to the propose method with the exception of clustering and	The proposed method applies clustering to the CI and critical measures, and then apply the derived protocols in order to separate the nodes based on CI cohesion. Thus, produce a BBN with a set of strongest links.	i. SMEs inputs can be merged i.e., using Thurstone's paired comparison as discussed in Chapter 4. ii. The use of GMM clustering and protocols leads to a most

protocols) and construct the BBN structure.		likely candidate BBN i.e., based on proposition 2. iii. Exceptional cases can easily be filtered out.
Peter and Clark (PC) algorithms (Bouckaert, 1995; Bregoli et al., 2021; Le et al., 2019; Madsen et al., 2017; Scanagatta et al., 2019; Spirtes et al., 1991; Zhang et al., 2020). This approach assumes a connected undirected graph and the direction is derived from CI measures.	The proposed method applied a clustering to the CI measure and derived protocols in order to separate the nodes based on CI cohesion.	i. SMEs inputs can be merged ii. The use of GMM clustering and protocols leads to a most likely candidate BBN i.e., based on Proposition 2. iii. Exceptional cases can easily be filtered out.
Score-based approaches (Bari, 2011; Scanagatta et al., 2019; Scutari, 2015; Zhang et al., 2020). This approach constructs n number of candidates BBN and selects the one with best prediction.	The proposed method applied a clustering to the CI measure and derived protocols in order to separate the nodes based on CI cohesion. Thus, this is a constraint-based approach	i. SMEs inputs can be merged ii. The use of GMM clustering and protocols leads to a most likely candidate BBN i.e., based on Proposition 2. iii. Exceptional cases can easily be filtered out.

		iv. Computational power reduction as discussed in Section 8.3 i.e., the highest performing score-based approach handled was 200 nodes (Zhang et al., 2020).
Hybrid approaches (Bari, 2011; Bregoli et al., 2021; Le et al., 2019; Scanagatta et al., 2019; Scutari, 2015; Scutari et al., 2019; Zhang et al., 2020). This is a combination of score-based technique and constraint-based solutions. That is the constraint guide the generation of the candidate BBN.	The proposed method applied a clustering to the CI measure and derived protocols in order to separate the nodes based on CI cohesion. Thus, this is a constraint-based approach	i. SMEs inputs can be merged ii. The use of GMM clustering and protocols leads to a most likely candidate BBN i.e., based on Proposition 2. iii. Exceptional cases can easily be filtered. iv. Computational power reduction as discussed in (Scutari et al., 2019).

8.6 Discussion and Conclusion

In this Chapter, I proposed a constraint-based BBN structural learning algorithm that considers inter-nodes conditional independence (CI) measures and critical weight to impose the learning constraints. The CI measure and the assigned essential weights from experts are clustered using GMM to obtain the network hierarchy based on the maximum likelihood of a link belonging to a cluster. The proposed algorithm was applied to DSA management for the thesis use case of Chapter 1. The results proved the following benefits:

i. Computationally inexpensive: based on the result in Table 37 and Table 38, the algorithm could handle many nodes and states values using little computational demand.

ii. Supports for SA projection of future situations: the proposed algorithm assigns links based on the maximum likelihood value (Proposition 2), which supports the situation prediction process using the Bayes rule.

iii. Uncertainty handling: uncertainty and the number of training data needed can be measured using the utilised CI measure. Based on the result in Section 8.5, the higher the changes in the fed data, the higher the number of links for the candidate BBN. Thus, a lower number of links shows poor availability of information update (i.e., no need for new BBN construction) or stable situation. That is, low number of links can happen not only based on insufficient data but also in terms stable priors (i.e., lack of priors' variation).

iv. Adaptability: learning in a dynamic DSA system requires a model adaptation (Kitchin and Baber, 2017) in both parametric and structural updates. The proposed algorithm is adaptable because of the consideration of agents' state priors. That is, agents' priors will only be used for the BBN construction.

v. Scalability: based on the results in Table 37 and Table 38, the proposed algorithm handles huge numbers of nodes with little resources. This is due to the efficient protocols used and the result of Lemma 1.

vi. SMEs-automation inputs integration and reusability: critical weight demonstrates how SMEs inputs can be integrated with the agents (e.g., UAVs) inputs to construct the BBN model that present the system DSA. Additionally, priors can be reused by other agents within the DSA system by assigning goal-based critical weights.

Additionally, the proposed Lemma 1 could estimate the optimal number of clusters for any categorisation process such as the k-means algorithm and so on. Regarding the DSA system, this chapter claims that the proposed algorithm would support autonomous (data-driven) or semi-autonomous DSA modelling, which could alert SMEs about the environmental phenomena causational changes. The SMEs would accept or correct the suggested network (using SMEs exceptional cases identification process or critical weight adjustment) to support joint SMEs-automation mission planning, i.e., human-automation joint planning. Additionally, in poor experts' experience, the algorithm could handle the SA presentation task.

Thus, this chapter proposed a constraint-based BBN structural learning algorithm based on maximum likelihood clustering of CI values among nodes of the BBN and SMEs' critical weights assignment. The algorithm was described using a simulation of the SMEs-UAVs team for managing DSA in forest fire scenes (i.e., the use case described in Chapter 1). The algorithm proved adaptability, heterogeneity handling, scalability by handling many nodes using little computational demand. Additionally, I proposed a lemma that could reduce the computational demand by estimating the expected number of clusters for the learning process.

9 Chapter 9 Discussion, and Conclusion

This chapter discusses the summary of the thesis and its position in the current stage of the literature. The discussion also highlights the novelties of proposed methods, their contributions (theoretically and applied concepts), and how the research questions were addressed. Again, the limitation of the methods was outlined as the future work.

9.1 Introduction

Refocussing on the thesis research questions (Chapter 1), this chapter will elaborate on how these questions were answered, novelties and contributions of the developed methods and algorithms, and the established future research directions. The thesis is divided into four main pillars. The first pillar includes problem definition and literature review (Chapters 1 and 2). The second pillar addresses the issue of agents' coordination for search activity and support for DSA management (Chapter 3). Chapters 4, 7, and 8 discuss how the agent's information can be transformed to present the system DSA, which serves as the third pillar. This involves the issues of information transformation to present SA, prediction and uncertainty handling, and how SA could be modelled in a dynamic system. The final thesis pillar focuses attention on merging the issues of agents' search activity coordination and the DSA management using the selected use case of forest fire monitoring as discussed in Chapters 5 and 6. Each of these pillars is targeted to address the outlined research questions.

9.2 Recap of the Thesis Research Questions

As outlined in Chapter 1, the thesis research questions are:

9.2.1 RQ1. How can we obtain a constraint-based search method for agents with limited resources operating in dynamic search areas?

Chapter 3 addresses this research question (RQ1) and lays a foundation for tackling research question RQ2 (i.e., based on the search plan's predictability feature, which supports prediction). The chapter proposes an efficient, scalable, adaptable, and predictable algorithm for generating agent search plans based on the constraints outlined in Chapter 1. The algorithm was applied to the thesis use case of forest fire searching, and a performance comparison with the existing popular solutions was conducted. The result proved a superior performance over the existing methods. The focus was on resource utilisation and how the proposed method supports easy prediction (i.e., the basis for the SA projection support). The results were based on a clear definition of performance metrics that affect agents, mission, and algorithm implementation. Results proved resources utilisation (based on the defined metrics), scalability, adaptability, predictability, easy application on real UAVs, and feasibility for a simple agents' situation prediction. The method adopted was a controllable path (flexible) and global system protocol based on the Delaunay-triangulation theorems. This produces a good search plan due to being flexible (easy to control based on the angle, quadrants, and edge), scalable, adaptable, and predictable (based on the overall control protocols as derived by the Delaunay-Inspired Multi-agent Search Strategy (DIMASS) algorithm of Chapter 3). Theoretically, this contributes to the set of hybrid area coverage algorithms that focuses on simplicity, system control (based on the mathematically derived protocols and methods as described in Chapter 3), resources utilisation, and the proposed system constraints (i.e., the version of DSA in the presented forest fire use case). This allows resources utilisation and easy situation prediction better than the fixed-pattern (in terms of adaptability and coordination) and pseudorandom (in terms of coordination, resources utilisation, and predictability) methods.

Additionally, the control protocols are simple, which allows easy deployment on low-capacity agents. The agents' resources utilisation problem was modelled as a Distributed Constraint Optimisation (DCOP) with mathematical modelling of various system activities. This shows how DCOP operate in a very dynamic and realistic environment using the case of forest fire monitoring. The formalisation of DSA and DCOP shows how various DCOP challenges can be demonstrated (e.g., the issue of uncertainty, finite/infinite horizon concepts, practical demonstration of variables operations, and the showcase of DCOP features, e.g., multi-objectivity in DCOP demonstrated using parameters changing based on various mission tasks, etc). Thus, the developed method took a step towards establishing resource-efficient and DSA-supported search method. This addressed the main limitations of poor coordination, DSA management, and resources limitations which are the main challenges of applying UAVs in various domains such as disaster management, surveillance, etc. (Cabreira et al., 2019, 2018; Chawla and Duhan, 2018, 2015, 2015; Jensen-Nau et al., 2021). Exiting solutions focus attention only on either the resource utilisation (Bevacqua et al., 2015; Bolander et al., 2018; Cabreira et al., 2019, 2018; Chawla and Duhan, 2018, 2015, 2015; Jensen-Nau et al., 2021, 2021; Kappel et al., 2020; Nebel et al., 2019; Sutantyo et al., 2011; Yang and Suash Deb, 2009; Yang, 2012, 2010) or the agent's local SA/DSA management (Berger et al., 2021; Heintzman et al., 2021; Ozkan and Kilic, 2022; Quintin et al., 2017) which limits their performance in terms of full system DSA support for agents with limited resources. Summarily, industries can utilise the proposed solutions and achieve agents' coordination (especially search activity coordination) with little resources when compared with the existing methods. Therefore, Chapter 3 addresses the limitation by developing an efficient, scalable, adaptable and predictable agents search algorithm based on geometrically derived protocols. The solution contributes to the set of hybrid methods by considering system protocols combined with geometric theorems (theorems from Delaunay-triangulation). The methods performance comparison was conducted on the thesis use case (i.e., a team of distributed agents tasked to conduct search activity in a dynamic environment). The results proved a better solution across the outlined challenges (measured using the mission sensitive parameters i.e., coverage, energy, redundant search, computational power, path divergence, scalability, adaptability, and predictability as outlined in Chapter 3. Finally, the proposed agents' search coordination algorithm demonstrates easy application on real UAVs by utilising the existing UAVs apps. The author tests it on DJI and Parrots Bebop drones to test its practical application.

Implementing the proposed Delaunay-Inspired Multi-agent Search Strategy (DIMASS) algorithm (Algorithm 1 in Chapter 3) on physical UAVs is easy and straightforward. The process starts by selecting the seeds waypoints (e.g., the longest non-cross waypoints in Figure 9 of Chapter 3), then a function (in any programming language, e.g., Python, or Java) can be developed to generate the remaining waypoints by taking the waypoints parameters, e.g., *generateWaypoint(Lx,Ly,e, q,Θ,h,n)*, where *Lx, Ly* are the longitude and latitudes of the current waypoint, *e* is the edge length of the opposing layer, *q* is the projecting quadrant (i.e., first to fourth), *Θ* is the projecting angle, *h* is the height of the waypoint (e.g., to avoid collision), and *n* is the number of waypoints in a layer of the solution (i.e., based on the Delaunay triangulations theorems discussed). In other words, the function *generateWaypoint(Lx,Ly,e,q,Θ,h,n)* produce a waypoint based on passed location, edge generation protocol, angle protocol, and a number of waypoints in a layer. Waypoints latitudes and longitudes differences can be computed using the Haversine formula or Euclidean distance can be used for planar coordinates. This can be implemented in any programming language (e.g., Java, as provided in the supplemental documents folder).

The generated plan can be transferred easily to the UAVs using the respective drone controlling applications downloaded from either Google Play Store or Appstore. The implementation can simply utilise the apps' waypoints planner function, e.g., DJI GO, DJI pilots, FreeFlight6, FreeFlight Pro, etc., for DJI and Parrots drones. Each UAV can be controlled by its respective application running on a tablet or mobile phone. For example, Figure 48 describes a single UAV plan (i.e., based on the controlling app screenshot). The plan in Figure 48 is created by simply clicking and dragging the waypoints from the Parrot drone and FreeFlight6 android app. Alternatively, waypoints can be sent to the UAVs via Python code, e.g., DJI Tello Python (i.e., for programmable drones) Application Programming Interface (API)[15]. The author tried all the mentioned methods, and the result looks similar to the simulated version (as per Figure 9 of Chapter 3). Parrot Bebob 1 &2 and DJI Phantom 3 Standard were used (i.e., to describe drones' heterogeneity). Thus, a key selling feature of the proposed method is the use of simple agents (low capacity and cheap) UAVs to solve the area coverage problem with minimised cost and easy emergence of Distributed Situation Awareness (i.e., based on agent's organisation and predictability features).

[15] https://github.com/code4funSydney/Tello

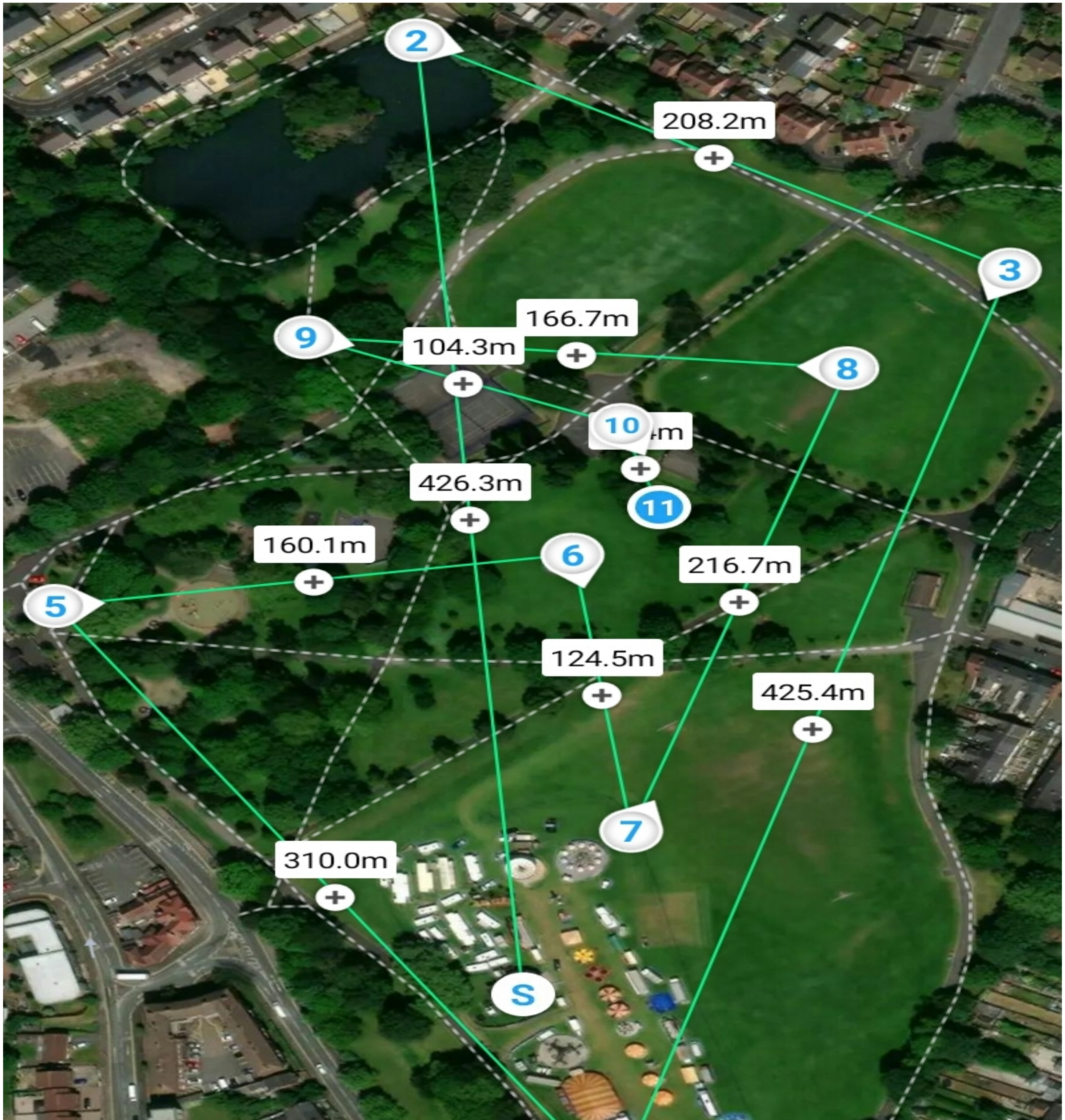

2
208.2m
3
166.7m
9
104.3m
8
10
11
426.3m
160.1m
6
216.7m
5
124.5m
425.4m
7
310.0m
S

Figure 48: Delaunay-Inspired Multi-agent Search Strategy (DIMASS) Algorithm Implementation on Real-UAV (DJI Pilot App)

From Figure 48, the UAV flight starts from waypoint S (which may be the base station or any arbitrary starting point). The seed waypoints are S, 2, 3, and 5. The second layer waypoints are waypoints 6-9, and the third layer waypoints are waypoints 10 and 11.

As such, the summarised answer to the research question in RQ1:

RQ1. How can we obtain a constraint-based search method for agents with limited resources operating in dynamic search areas?

Is by developing a search method that is scalable, adaptable, predictable and utilises the agents' resources. All these can be achieved using good system and individual agents' protocols, as shown by the results of Chapter 3. Achieving a good system protocol seems to be very difficult; the author tried the geometrically derived protocols from the Delaunay triangulation process and this shows a good result for the agent's search problem. This is due to being easy to control (based on the controllable path elements, i.e., angles, quadrants, and edges length). For example, the redundant search can be avoided and coverage can be maximised by adjusting waypoints. This allows the ability to control the path and optimise the resources parameters, e.g., by minimising redundant search, number of agent interactions (by being structured due to the system protocols), memory use (by being computational simple and reducing the number of agents interactions), computational power (by being computationally simple), mission time, energy, and maximising coverage (by avoiding redundant search and creating highly separable waypoints). In terms of qualitative features, predictability is maintained by using the system protocols derived from the Delaunay-triangulation layers and protocols. This allows Picture Compiler (PC) or host to predict a simple agent's location to arrange data collection or recovery from failure.

Similarly, scalability is maintained by utilising the applied Delaunay-triangulation theorems. For example, a protocol on layer waypoints reflection, refraction, and seed waypoints variations were developed to support multiple agent coordination. Thus, this coordinates the agent's search without too many interactions (remember, limited interaction is part of the constraints), manages the agents' computational power (based on the result of low computational demand from Chapter 3), and adapts to the dynamic environment changes (e.g., paths can be changed to map fire shapes by changing the angles, quadrants, and edge lengths which maintains the adaptability of the search plan). Predictability is maintained by the structure imposed through the plan generation protocols (i.e., the Delaunay triangulation theorems). The predictability supports the SA management of Chapters 4, 5, 7, and 8 by allowing simple agent situation prediction. The success of the search plan was measured using the defined metrics, and a clear description of how this can be done was made. In summary, a clear answer to this research question (RQ1) is to develop a search plan generation method that is simple (i.e., in terms of memory use, control, and computational power), adaptable, predictable, and scalable. The proposed method results from Chapter 3 (DIMASS) proved the achievement of resource utilisation (by producing waypoints with minimal redundant search and higher coverage), scalability, predictability (to support agents' situation prediction), and adaptability. Thus, the proposed method suggests that, instead of tasking agents to conduct complex search activities (e.g., waypoints processing via complex interactions), why not make their tasks simple using system protocols.

9.2.2 How can we manage the Situation Awareness of distributed agents?

In order to support how the search mission information can be presented to reflect the system SA, Chapter 4 proposes using Bayesian Belief Network (BBN). This is in contrast to the existing static and non-dynamic approaches of propositional networks, ontologies, fuzzy logics, and concept maps. The thesis develops algorithms and methods, application process, sensor information transformation to reflect the system SA, system SA modelling, and how agents’ contributions can be integrated using Thurstone’s paired comparison all using the thesis use case. Chapters 4, 7, and 8 describe the application of BBN to the system’s use case DSA management and the potential for a better method than the existing use of the propositional network, ontology, fuzzy logic, and concepts maps in terms of the outlined DSA features as discussed below (Berger et al., 2021; Bouvry et al., 2016; Burov, 2021; Galton and Worboys, 2011; Lohia et al., 2019; Salmon and Plant, 2022; Stanton et al., 2006, 2009; Kethavarapu and Saraswathi, 2016; Zhang et al., 2021). The outlined advantages of using BBN are (i) phenomena multiple states presentations, (ii) good interface (similar to the ontology, concept maps, and propositional network), (iii) measuring phenomena states using probabilities, (iv) ability to handle predictions and uncertainties using learning algorithms (v) agents' heterogeneity handling, (vi) adaptability management based on states probabilities, and (vii) search area information reusability (by revising the states priors). Thus, Chapter 4 proposes a flexible and adaptable method of modelling the system DSA and describes various DSA/BBN the concepts, and Chapters 7 and 8 support the outlined advantages claims with experiments and results.

Chapter 7 describes how prediction and uncertainties (in the form of missing information or soft findings) can be handled to support the SA projection. The issues were addressed in terms of single or multiple states (i.e., a combination of different BBN states), prediction, and uncertainty handling. The result from the application of expectation-maximisation (EM) algorithms shows good performance for unstructured data in comparison with time series models and Gaussian process not only in terms of prediction/uncertain values estimation accuracy but also in terms of DSA management in a system with varying agents (i.e., by considering agents role to minimise agents' interactions and parameters definition). The chapter proposes DSA-based metrics based on the error rates to reflect mutual information relation and agents' goals variation. Thus, the proposed metrics fit the DSA system better than the accuracy metrics such as the logarithm loss, Brier score, spherical payoff, etc. The proposed method is tested against different forms of uncertainties, varying sizes of mission data, and resources management.

Chapter 8 builds on Chapter 4 to describe how an adaptive DSA model using BBN can be emerged within the system. A structural learning algorithm was developed and applied based on the BBN state's priors and SME's goal-based weights. The priors and SMEs weights are clustered using Gaussian Mixture Model (GMM) and then passed to derived protocols. The protocols are based on the BBN node's conditional independence, which is an additional advantage to the existing strategies of concept map, propositional network, ontology, and fuzzy logic. The algorithm contributes to the classes of constraint-based BBN structural learning algorithms and demonstrates how it can improve the DSA adaptability, resource efficiency, and scalability. The chapter also proposes a lemma to increase the scalability of the structural learning algorithms. Existing solutions of constraint-based methods, such as the PC-algorithm

(Bouckaert, 1995; Huang et al., 2003; Le et al., 2019; Levin et al., 2000; Madsen et al., 2017; Qi et al., 2021; Sanz-Pena et al., 2021; Scanagatta et al., 2019) focus only on the conditional independence protocols whereas the propose solution pays attention to respective agents contributions, DSA management, and resources utilisation (remember, the reason is due to the use of simple UAVs). The proposed method allows exceptional case filtering (based on the SMEs' judgment) to avoid a pure probabilistically derived structure as in the score-based methods (Bregoli et al., 2021; Scanagatta et al., 2019; Zhang et al., 2020). Finally, the results proved scalability, adaptability, and resource management better than the existing methods, and a lemma was proposed to reduce the computational demand. Thus, Chapter 8 formalises the concept of BBN structural learning with DSA in a team of varying agents. Remember, all the outlined contributions were in consideration of agents' resource utilisation due to the use of micro and mini UAVs as discussed in Chapter 1. Collectively, Chapters 4, 7, and 8) contribute to a number of theories and methods in DSA. These are outlined as follows:

i. Step-by-step depiction of Endsley's three stages of SA (perception, comprehension, and projection as in (Endsley, 1995)) in DSA and their assimilation with DSA's phenotype and genotype schemata. Based on the proposed BBN method discussed in Chapter 4, 7, and 8, the Endsley's stages of perception is flexibly (not statically) presented by the BBN's states probabilities. For example, the detection of fire by a simple UAV leads to the update of fire present state probability (which is a measure of belief) of the fire node within the BBN based on sensor reliability weight e.g., 40%, 90%, or 100% etc., as demonstrated in Chapter 4. This is at the simple agent's level (phenotype level), which could be very different at the genotype level (e.g., the picture compilers or host level). For instance, if the reporting UAV is using a visual camera and the operating time is day-time, the PC or host can perceive fire absence even if the UAV reported fire presence based on the fact that the UAV's sensor is confused by a fire-like object, e.g., yellow building or dried grasses. Thus, perception at the genotype level is emerged by considering other contributing information and their contextual weight (derived from the logical understanding of the combined information). Similarly, the comprehension and projection for the simple agents could be making small jumps to map the fire and controlling the jumps based on the predicted rule-based fire spread (perhaps using simple inbuilt fire spread protocols). This is very different with the PC and host levels. At the PC level, other environmental phenomena e.g., wind direction, wind speed, and other agents sensor information (e.g., infrared, temperature, etc. sensors) need to be considered (based on their states probabilities values) to understand the fire spread and make predictions, e.g., where the fire will move next (i.e., at host level), when to send firefighters (i.e., at host level based on the complex analysis of large system phenomena). Thus, this shows how various versions of SAs from varying agents with varying goals and situations can be managed within the DSA system using BBN. As such, managing the BBN states' probabilities across nodes at genotype and phenotype levels demonstrate the process of emerging DSA of the system at both hierarchies and heterarchies. It is beyond any reasonable doubt that the existing tools of ontology, concept maps, propositional networks, and fuzzy logic lack these flexibility features.

ii. Flexible situation presentation: existing methods of concept maps, propositional networks, ontologies, and fuzzy logics present the environment situation in a static fashion, e.g., fire presence or absence (i.e., Boolean). The use of BBN opens doors for flexible belief presentation in the form of probabilities. The flexible belief presentation is guided by the existing probabilities and conditional probabilities theorems.

iii. Prediction and uncertainty handling: Chapter 7 discusses some methods that can be applied to handle prediction and uncertainty handling in DSA. EM algorithm shows good performance in handling the prediction and uncertainty issues in a data-driven fashion (i.e., with minimal specifications of parameters). Thus, this not only describes how DSA can be managed using BBN but also how the issues of prediction and uncertainty can be dealt with.

iv. BBN structure update to manage the system DSA: one of the key features of applying BBN to DSA management is the ability to update not only the priors of the states as in (i) above but also the update of the structure of the BBN to represent current situation-based states priors. In Chapter 8, the thesis develops a method that considers the link strength of the BBN and accommodate agents' (both automation and Subject Matter Experts) contributions during BBN structural updates.

Therefore, a summarised answer to the research question:

RQ2. How can we manage the Situation Awareness of distributed agents?

Based on the results of Chapters 4, 7, and 8, the answer is to develop a DSA modelling tool that allows agents' contributions integration (e.g., using probabilities and the Thurstone's paired comparison as described in Chapter 4), flexibly measures agents' beliefs (priors update algorithm of Chapter 4), allows predictions and uncertainty handling (using various learning algorithms of Chapter 7 or simple predictions as described in Chapters 3 and 7), and is adaptable to various environmental situations using structural learning algorithm of Chapter 8. BBN demonstrates an ability to provide these features as described in Chapters 4, 7, and 8. Algorithms and methods were developed to show how each aspect will be addressed. The prediction and uncertainty handling issues were addressed using learning in either a single aspect or combined system information. Of course, the learning has to be efficient and fast enough to produce the prediction of the dynamic environment, and the Expectation-Maximisation (EM) algorithm and Gaussian Process demonstrate good results in addressing the issues. The aspect of the adaptable SA model was addressed using the developed structural learning algorithm.

9.2.3 RQ3. How could agents' search plan support SA management?

In addition to the DCOP formalisation with DSA, Chapter 5 discusses two things: (i) how to manage conflicts in agents' data and (ii) how to monitor agents' interactions. The BBN Conditional Probability Table (CPT) addresses the agents' sensor conflict using probability measures, which derived its values from various states' priors' combinations or SMEs assignment. Thus, the CPTs receive updates based on the priors generation algorithm or SMEs weight allocation (Chapter 4). Each agent's corresponding situation has a candidate entry within the BBN. In terms of agents' interactions to utilise resources, Shannon's entropy and the BBN state priors were applied to handle the challenge. This ensures a measurable agents interactions technique (i.e., based on resource utilisation), unlike the focus on interaction monitoring as in (Kitchin and Baber, 2017; Wiltshire et al., 2018) or unmonitored interactions as in (Amador et al., 2014; Fioretto et al., 2017; Hoang et al., 2016; Maheswaran et al., 2004). The proposed method also describes how agents' activities (i.e., transitions of situation-actions) can be coordinated using BBN CPTs. This concept of agents' interaction was modelled as DCOP and formalised with DSA. Thus, this provides answers on how agents' situations and actions transitions can be handled (i.e., from sensor information perception to CPTs consultation and actions). To my knowledge, this is the first time DCOP is formalised with DSA systems (i.e., based on the DSA-DCOP concepts of Chapters 3 and 5). The benefit of the formalisation is the efficient situation understanding with minimal resources.

Chapter 6 discusses modelling methods for agents and search area modelling, dynamic phenomena modelling, sensor data collection and analysis, physical experiments to improve the credibility of the simulation, learning methods, and the relationship between the search planning methods and data collection. Results of physical experiments using real forest fires and UAVs operation were discussed. This is to address the issues of poor simulation and system models, as complained by many authors (Altameem and Amoon, 2010b; Ayub et al., 2020; Bevilacqua et al., 2017; Gage and Murphy, 2004; Galland et al., 2014, 2013; Heintzman et al., 2021; Rathbun et al., 2002). The relation between search plan and information shows that structured search plans (e.g., the proposed DIMASS strategy of Chapter 3) offer organised data in a barely structured search area based on the result in Chapter 6. This allows simple prediction which could support the SA projection (i.e., prediction of plausible future state and uncertainty handling). In the case of SA projection for unorganised data (obtained from pseudorandom strategies), Chapter 7 addresses that using expectation-maximisation algorithm learning

Thus, the summarised answer to the research question:

RQ3. How could agents' search plan support SA management?

Is by providing a search plan that supports SA management features (e.g., supports belief measurement, information organisation to present SA, prediction and uncertainty handling). Results from Chapter 6 shows that structured search plans generate organised data (i.e., data that follows a certain structure) in a structured search area. Thus, having organised data eases the SA aspect of prediction (e.g., simply using interpolation or Bayes rule as described in Chapter 6). In the case of an unorganised search area, the EM algorithm demonstrates an ability to address the prediction and uncertainty handling (as shown in Chapter 7). Based on the result of Chapter 5, a good search plan can support SA without much agents interactions (i.e., too much agents interactions do not guarantee good SA management and resource utilisation). Thus, it is part of the support of a search plan method for SA management to demand less number of interactions (as discussed in Chapter 3). For example, from the result of Chapter 5, some smaller numbers of agent interactions are more fruitful than larger ones in terms of resources utilisation and SA management (i.e., agreement on the waypoints to be selected as described in Chapter 6). Thus, the chapter's (Chapter 6) proposes a way of managing the interactions of agents during search plan and how this could support the system DSA management.

9.3 Methods Application Adaptability

The methods describe in this thesis focus attention on the complex issue of forest fire monitoring. This allows dynamic system modelling and potential for numerous application in other fields. This is based on the fact that, solutions to forest fire scenes' problems could be extended to addressed issues in various dynamic and complex systems (Weick 1995). The key features need for the application of the thesis methods and algorithms to other systems is the presence of (i) distributed agents, (ii) outlined constraints e.g., limited energy, communication range, etc. (fully outlined in Chapter 1 Section 1.1) (iii) dynamic environment, and (iv) the need for SA management across agents levels (both phenotype and genotype schemata). This conforms to various number of systems' specifications such as other disaster management e.g., flood control, avalanche management, etc., patrol system, banking system, wireless sensor network, ad hoc communication using a team of distributed agents, collision avoidance, Simultaneous Localisation and Mapping (SLAM) problems, multi-robots coordination, etc., (Alkhatib et al., 2014; Bouvry et al., 2016; Bevacqua et al., 2015; Bolander et al., 2018; Cabreira et al., 2019, 2018; Chawla and Duhan, 2018, 2015; Ghamry and Zhang, 2016; Jensen-Nau et al., 2021; João, 2012; Muñoz et al., 2021; Nebel et al., 2019; Nurzaman et al., 2009; Ozkan and Kilic, 2022). For instance, the solutions to agents' area coverage problem addressed in Chapter 3 can be extended to address the problem of ad hoc communication management (e.g., during disaster management) by a team of UAVs, wireless sensor distribution, etc. Similarly, the DSA management concept using BBN can be addressed by configuring the nodes and states probabilities to conform with the specified problems e.g., the nodes can be changed to address network presence, network load, etc., instead of the current fire presence or absence (remember this can be in line with considering the agents coordination using the proposed solution in Chapter 3) for the ad hoc communication system. As such, the approach of DSA management in consideration of agents’ coordination provides a wider range of solutions to the system control and information processing to the extent that simplest agents can be applied (i.e., the overall system cost is reduced). Merging the system control and effective information processing using DSA promotes the system’s efficiency by reducing cost and risks (as can be seen in the thesis described use case of forest fire monitoring which reduce cost and risks by utilising simple UAVs). However, the solutions provided by this thesis targeted one of the most complex systems which could be applied to other less complex problems. For example, in the event of wireless sensor area coverage, they are non-mobile agents; as such, the solution provided in Chapter 3 can address the challenge without applying the agents’ positioning (e.g., the concept of reflection discussed in Chapter 3) protocols while maintaining scalability (i.e., the concept of agents' reflection or refraction). Despite the large number of potential applications offered by the thesis methods and algorithms, some very simple approaches can utilise the existing alternatives. For example, the basic concept of depicting the relationship among system phenomena in psychology (i.e., non-dynamic and less complex representation) could require just a demo, e.g., using concept maps by applying a pern and paper sketch

Similar to the propositional networks, ontologies, concept maps, and fuzzy logic, BBN can be initialised by using the SMEs’ knowledge and a simple sketch of environmental phenomena, perhaps using pen and paper (as discussed in Chapter 4). However, BBN has a large number of design softwares such NETICA[16], Bayes server[17], etc. These softwares provide not only an easy interface for developing BBN but also an ability to apply learning algorithms for predictions, uncertainty handling, and structural learning. They, however, provide an API for integration with other softwares e.g., integration with the AMASE (agents simulation software adopted by the thesis) and IDEs (integrated Development Environments) such as Eclipse, Netbeans, etc.

[16] https://www.norsys.com/netica.html
[17] https://www.bayesserver.com/

9.4 Positioning the Research in Current Literature

The first part of the thesis addresses the issue of agents' search mission coordination under the imposed constraints. This tackles the thesis's first objective, i.e., "Develop an efficient way of coordinating the automation agents to conduct search activity". Many researchers address the problem in consideration of resource utilisation using fixed-pattern, pseudorandom, or hybrid methods (Bevacqua et al., 2015; Bolander et al., 2018; Cabreira et al., 2019, 2018; Chawla and Duhan, 2018, 2015; Ghamry and Zhang, 2016; Jensen-Nau et al., 2021; João, 2012; Muñoz et al., 2021; Nebel et al., 2019; Nurzaman et al., 2009; Ozkan and Kilic, 2022; So and Ye, 2005; Sutantyo et al., 2011; Vasile and Zuiani, 2011; Vincent and Rubin, 2004; Waharte et al., 2009b; Yang and Suash Deb, 2009; Yang, 2010; Yanmaz et al., 2011; Zhang et al., 2005); however, this thesis focuses attention not only on resources utilisation but also considers DSA support features through focusing on adaptable, scalable and predictable agents' activities. Although few researchers (Berger et al., 2021; Heintzman et al., 2021; Quintin et al., 2017) believed that it is a novel direction, clear definition of the metrics, model operation in a dynamic environment, mathematical modelling, DSA-support methods and algorithms, and resources utilisation model (as formalised using DCOP) was only developed in this thesis. The thesis suggested new success measuring metrics and system-based protocols (i.e., contrary to the local agent's rule) that allow agents' situation prediction and support for SA management. The protocols were built on top of the Delaunay-triangulation theorems and derived as search plan protocols (control rules). Regarding the model, to my knowledge, this is the first attempt to formalise DCOP with DSA and agent's search activity. Thus, this shows a practical application of DCOP in a dynamic environment and agents' operation to manage the system DSA (as described in Chapters 3 and 5).

The thesis objectives derived from RQ2 and RQ3 were addressed by proposing a Bayesian Belief Network to model the system SA. Learning algorithms were applied for unstructured agents' information prediction and uncertainty handling. Existing works used propositional networks, ontologies, concept maps, and fuzzy logics (Baader et al., 2020; Burov, 2021; D'Aniello et al., 2018, 2015; Galton and Worboys, 2011; Kokar et al., 2009; Li et al., 2018; Liu et al., 2013; Lohia et al., 2019; Stanton et al., 2006; Stanton, 2016; Uma Pavan Kumar Kethavarapu and S. Saraswathi, 2016; Zhang et al., 2021). However, the use of BBN demonstrates additional advantages in terms of multiple states presentation, belief measurement, prediction ability, uncertainty handling, and adaptability (as discussed in Chapters 4,5,7, and 8). The outlined specific concepts were first developed in this thesis. Algorithms were developed to describe how agents' sensor information can update the BBN at various levels. To illustrate its feasible application in a dynamic system, the thesis uses a case study of a team of agents' mission for forest fire monitoring based on simulations and physical experiments. This form of simulation looks more realistic in presenting a dynamic and complex system (Ayub et al., 2020; Bevilacqua et al., 2017; Galland et al., 2014, 2013). To my knowledge, this is the first time BBN was applied to formally describe agents DSA. The adopted direction seems attractive to researchers within the field of DSA (Rosário et al., 2021). However, the thesis approach can be distinguished by the following novel features:

i. DSA model and its adaptability: this is modelled using BBN's formal properties that allow belief measure and presentation, situation-based BBN configuration using structural learning, and agent interactions analysis. In terms of the belief measurement, the existing approaches of the propositional network, concept map, ontology, and fuzzy logic offer only a single presentation per situation (Baader et al., 2020; Burov, 2021; Galton and Worboys, 2011; Kokar et al., 2009; Liu et al., 2013; Lohia et al., 2019; Stanton et al., 2001;N Stanton et al., 2001; Stanton, 2016; Stanton et al., 2009, 2006; Stefanidi et al., 2022; Uma Pavan Kumar Kethavarapu and S. Saraswathi, 2016). Thus, these approaches provide only the qualitative presentation of how agents access information within a DSA system and omit the agents' efforts toward DSA management.

ii. Multiple state interface: in addition to the belief measurement, BBN is similar to existing methods in terms of interface and provides measurable state presentations which could easily be updated by SMEs or automation agents.

iii. Prediction and uncertainty handling: the thesis took a step toward addressing the concept of prediction and uncertainty handling (as part of the DSA projection) within a DSA system using learning. Various learning algorithms and methods were applied to describe how predictions and uncertainties can be handled concerning the DSA system constraints. The prediction issue was segmented into simple parameters based on prediction (estimation, e.g., as described in Chapter 3) whenever possible and learning algorithms for unstructured data. The outcomes uniquely investigate how predictions and different forms of uncertainties can be addressed based on mission learning. This significantly contributes to the aspects of understanding the concepts of prediction and uncertainty handling in the DSA system, especially involving varying agents. This is different from the existing methods that look at independent states of the node predictions, such as the work of (Murray and Perera, 2022; Sulistyawati et al., 2011; Wang et al., 2021; Zhao et al., 2021). Again, we proposed measuring metrics to grade predictions and uncertain values estimation based on agents' goals.

iv. Adaptable SA management: the thesis proposes a novel structural learning algorithm for SA model presentation based on the varying situation. The algorithm uniquely considered clustering the CI measures among nodes and internodes assigned weights using the Gaussian mixture model and Bayesian-derived protocols (Chapter 8). The outcome contributes to constraints-based structural learning, which I believe better suits the DSA system. This is different from the CI-weighted approaches (e.g., Chow-Liu tree) or pure PC (Peter and Clark) algorithms (Bari, 2011; Bouckaert, 1995; Bregoli et al., 2021; Dama and Sinoquet, 2021; Le et al., 2019; Madsen et al., 2017; Marcano et al., 2020; McCaskey et al., 2020; Meloni et al., 2009; Sathe et al., 2013; Scutari, 2015; Zhang et al., 2020) because related nodes can easily be identified using the CI measures.

v. Monitorable agents' interactions: Chapter 5 describes how agent interactions can be measured in terms of resource utilisation using Shannon's entropy augmented with the formal properties of the proposed Bayesian Belief Network (BBN). Additionally, the BBN CPTs update process was described in detail. This allows agents to transit between various actions and differentiate between useful and useless agents' interactions, unlike the approaches in (Kitchin and Baber, 2017; Wiltshire et al., 2018). Overall, this provides a way of managing DSA in a resource-efficient manner.

vi. Sensor information conflict resolution: conflicts in agents' information can be addressed using the CPT of the BBN through learning or SMEs allocation as described in Chapter 5. This reduces the number of agent interactions as in interaction-based methods, e.g., publish-subscribe technique, consensus etc.(Ghamry and Zhang, 2016; Haksar and Schwager, 2018; Merino et al., 2010; Salerno et al., 2005). All agents' situation-action transitions are monitored using the BBN CPTs.

vii. Reusability: the developed model can be reused for varying missions by changing the nodes' priors and CI measures. This is very different from the existing methods.

viii. Resources utilisation: the proposed methods consider agents resources utilisation to allow easy application.

Generally, the thesis took the direction of managing DSA within a team of varying agents in a cost effective manner. This combines the fields of DSA management for a team of agents and resource utilisation.

9.5 Future Work

The research presented in this thesis leads to a number of research fields worthy of investigation. For instance, different routes can be taken in terms of BBN formalisation with the concept of DSA e.g., in Human Factors and Ergonomics fashion, large agents coordination, schema-based performance analysis, SA factors (memory, workload, etc) consideration and performance analysis, etc. Exploring these could lead to a number of interesting research questions. For instance, the question of "How BBN supports Subject Matter Experts (SMEs) operating in dynamic and uncertain situations?", "Does SMEs and other agents organisation using BBN. support DSA management?", "How BBN promotes SMEs and UAVs cooperation in managing the system's DSA?", "To what extend BBN supports system control?", "Does BBN reduce the effect of large number of agents' coordination? etc. Indeed, this generates a number of interesting research areas toward understanding the concept of agents' coordination, DSA management, and system resources utilisation especially when involving large scale of heterogeneous agents. I know these challenges were triggered by the adoption of BBN to model systems' DSA and they can be addressed in many ways. For instance, one of the obvious approaches to address the BBN adoption in human-automation team is to assess the SMEs' performance using any of the SA assessment techniques e.g., Situation Presence Assessment Measurement (SPAM), Situation Awareness Rating Technique (SART), Situation Awareness Global Assessment Technique (SAGAT) with respect to similar system settings (Kitchin and Baber 2016; Endsley 2000).

The limited number of agents, collected data, and BBN concepts must, as in SA studies, be acknowledged. Further research could look at the challenges of large-scale agents coordination, collected data, and BBN representations (which, perhaps have a large number of limitations, such as BBN structural learning management for large networks, SMEs weight allocations, constructions, etc.). For example, the issues of SMEs' weight assignment for a large number of nodes need to be addressed. One of the possible solutions is to approach the problem in a divide-and-conquer fashion i.e., by dividing the number of nodes into small portions and assigning SMEs weights to groups. Similarly, the issue of exceptional cases monitoring using CPTs' will require a similar divide-and-conquer technique, especially for the large BBNs.

Furthermore, although the thesis focuses attention on the use of BBN to model DSA, extending this to other models of SA, e.g., shared SA, team SA, situated SA, sense-making etc., could expand our understanding of the theories of SA. Although I can not suggest how these can be achieved, but the application in the field of DSA (as adopted by the thesis) could help in expanding the application of BBN not only in the field of DSA, but also at the general level of SA research tree.

Again, one of the limitations of the proposed agent's coordination method is a large number of agents to be controlled over n dimensions. For example, assume a number of 50 UAVs to be coordinated using the proposed DIMASS method of Chapter 3; the number of distinct control protocols over n dimension will be difficult to develop. Thus, the definitions and propositions proposed in Chapter 3 require further extension to accommodate a large number of agents as well as customisation to fit various systems settings based on their features e.g., wireless sensor network coverage problems, ad hoc communication relay problems, other disaster management, etc. Similarly, searching for the best solution, i.e., by adjusting the edges, angle, and quadrants of the search plan involving many agents, will be very difficult due to a large number of waypoints and agents. Although solving these issues could require complex mathematical derivations (as seen from work done in Chapter 3), this will pave the way for simplifying agent coordination tasks and overall DSA emergence. Again, collision avoidance for a large number of agents will require sense and avoid or waypoints altitudes variations. Still, achieving this for many agents will be very difficult to maintain.

The aspect of the real-time human configuration of BBN and analysis of how supportive the presented BBN will be to individual SMEs based on their goals need further evaluation. This is to understand the changing nature of the environment and how SMEs' activities are affected by those changes in the presented BBN.

Overall, since the thesis focuses attention on the technical aspect of DSA management for the human-automation team (with a higher focus on the automation aspect), resources utilisation, and search mission coordination, future work can pay attention to the human factors and how the developed models, methods, and algorithms support the SMEs decisions and action in DSA or other models of SA. This could lead to dozens number of problems concerning DSA and agents' coordination. For instance, considering the thesis use case (i.e., forest fire monitoring), an assessment study can be made to analyse the effect of SMEs' decision on agents coordination and DSA management, DSA presence evaluation and emergence latency, factors promoting or demoting the DSA management when BBN is adopted, etc.

References

Adhikari, R., Agrawal, R.K., 2013. An introductory study on time series modeling and forecasting. LAP Lambert Academic Publishing, Chisinau, Republic of Moldova.

Adil Khan, M., Pečarić, Đ., Pečarić, J., 2020. New refinement of the Jensen inequality associated to certain functions with applications. Journal of Inequalities and Applications, 76. https://doi.org/10.1186/s13660-020-02343-7

Afolayan, T.A., Ajayi, S.S., Ajayi, S.A., 1979. Reasons for further burning experiments in west african savanna woodland. The Commonwealth Forestry Review 58, pp253–265.

Afzal, A., Katz, D.S., Le Goues, C., Timperley, C.S., 2021. Simulation for robotics test automation: developer perspectives, 14th IEEE Conference on Software Testing, Verification and Validation (ICST), Porto de Galinho, pp263–274. https://doi.org/10.1109/ICST49551.2021.00036.

Alkhatib, A.A.A., 2014. A review on forest fire detection techniques. International Journal of Distributed Sensor Networks 10. https://doi.org/10.1155/2014/597368.

Allen, M.A., 1994. Thurstone’s method of paired comparisons: review of an old but still-useful measurement protocol. Annual Meeting of the Southwest Educational Research Association, San Antonio, US. 15.

Altameem, T., Amoon, M., 2010. An agent-based approach for dynamic adjustment of scheduled jobs in computational grids. Journal of Computer and System Sciences Internationl 49, pp765–772. https://doi.org/10.1134/S1064230710050114.

Altarriba, E., Halonen, J., 2020. Analysis of ship voyage data based on chow-liu tree augmented naïve bayes-method to support biofouling management. Journal of Maritime Research 17, pp47–54.

Amador, S., Okamoto, S., Zivan, R., 2014. Dynamic multi-agent task allocation with spatial and temporal constraints. International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS ’14. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp495–1496.

Armengaud, E., Watzenig, D., Karner, M., Steger, C., Weiß, R., Netzberger, C., Kohl, M., Pistauer, M., Pfister, F., Gall, H., 2009. Combining the advantages of simulation and prototyping for the validation of dependable communication architectures: the TEODACS approach. SAE International Journal of Passenger Cars 2, pp309–318. https://doi.org/10.4271/2009-01-0763

Aydin, T., Gokturk, M., 2018. Measurement of Situational Awareness in the Supervisory Tasks. 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), Adana, Turkey, pp1–6. https://doi.org/10.1109/ASYU.2018.8554011.

Ayub, S., Petrunin, I., Tsourdos, A., Al-Rubaye, S., Stapylton, G., Dent, G., 2020. In-flight entertainment datalink analysis and simulation, AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), San Antonio, USA, pp1–10.

Baader, F., Borgwardt, S., Koopmann, P., Thost, V., Turhan, A.-Y., 2020. Semantic technologies for situation awareness. Künstliche Intelligenz 34, pp543–550.

Baber, C., Morin, C., Parekh, M., Cahillane, M., Houghton, R.J., 2011. Multimodal control of sensors on multiple simulated unmanned vehicles. Ergonomics, pp792–805.

Baber, C., Nathan, M., Sagir M., Y., 2021. A Mixed-initiative Effect Planning Tools. HFES 65th Annual Meeting 2021, Maryland, US.

Baek, H., Lim, J., 2018. Design of future UAV-relay tactical data link for reliable UAV control and situational awareness. IEEE Communications Magazine 56, pp144–150.

Bailon-Ruiz, R., Bit-Monnot, A., Lacroix, S., 2022. Real-time wildfire monitoring with a fleet of UAVs. Robotics and Autonomous Systems, p104071.

Banfield, J.D., Raftery, A.E., 1993. Model-Based Gaussian and Non-Gaussian clustering. Biometrics, pp803–821.

Bari, M.F., 2011. Bayesian Network Structure Learning. 4th Annual Meeting of Asian Association of Algorithms and Computation, Japan.

Berger, C., Doherty, P., Rudol, P., Wzorek, M., 2021. Hastily formed knowledge networks and Distributed Situation Awareness for collaborative robotics. Autonomous Intelligent System 1. https://10.1007/s43684-021-00016-w

Bevacqua, G., Cacace, J., Finzi, A., Lippiello, V., 2015. Mixed-initiative planning and execution for multiple drones in search and rescue missions, 25th International Conference on International Conference on Automated Planning and Scheduling, ICAPS'15. AAAI Press, Jerusalem, Israel, pp315–323.

Bevilacqua, M., Tsourdos, A., Starr, A., 2017. Particle swarm for path planning in a racing circuit simulation, 34th IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Turin, Italy, pp1–6.

Bhattacharyya, A., Gayen, S., Price, E., Vinodchandran, N.V., 2021. Near-optimal learning of tree-structured distributions by Chow-Liu. 53rd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2021. Association for Computing Machinery, New York, USA, pp147–160.

Bjurling, O., Granlund, R., Alfredson, J., Arvola, M., Ziemke, T., 2020. Drone Swarms in forest firefighting: a local development case study of multi-level human-swarm interaction. 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society, NordiCHI ’20. Association for Computing Machinery, New York, USA, pp1–7.

Boissonnat, J.-D., Dyer, R., Ghosh, A., 2013. The stability of delaunay triangulations. International Journal of Computational Geometry and Application, 23, pp303–333. https://doi.org/10.1142/S0218195913600078

Bolander, T., Engesser, T., Mattmüller, R., Nebel, B., 2018. Better eager than lazy? how agent types impact the successfulness of implicit coordination. 6th International Conference on Principles of Knowledge Representation and Reasoning, Tempe, Arizona.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. 19th International Conference on Computational Statistics. Paris, France.

Bouckaert, R.R., 1995. Bayesian belief networks: from construction to inference: Bayesiaanse belief netwerk: van constructie tot inferentie. Universiteit Uteretch.

Bouguettaya, A., Zarzour, H., Taberkit, A.M., Kechida, A., 2022. A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms. Signal Processing, p108309.

Bouvry, P., Chaumette, S., Danoy, G., Guerrini, G., Jurquet, G., Kuwertz, A., Muller, W., Rosalie, M., Sander, J., 2016. Using heterogeneous multilevel swarms of UAVs and high-level data fusion to support situation management in surveillance scenarios, 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Kongresshaus Baden-Baden, Germany, pp424–429.

Bowditch, N., 2002. American practical navigation 2002 bicentennial edition p882, National Imagery and Mapping Agency, Maryland, US.

Breejen, E.D., Breuers, M., Cremer, F., Kemp, R., Roos, M., Schutte, K., Vries, J.S.D., 1998. Autonomous forest fire detection. 6th Australian World Wide Web Conference, Glen Waverley, Australia, pp167–181.

Bregoli, A., Scutari, M., Stella, F., 2021. A constraint-based algorithm for the structural learning of continuous-time Bayesian networks. International Journal of Approximate Reasoning 138, pp105–122. https://doi.org/10.1016/j.ijar.2021.08.005.

Brier, G.W., 1950. Verification of forecasts expressed in terms of probability. Monly Weather Review 78, pp1–3.

Brosamler, G.A., 1988. An almost everywhere central limit theorem. Mathematical Proceedings of the Cambridge Philosophical Society, pp561–574.

Burov, Y., 2021. Knowledge based situation Awareness process based on ontologies. 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine.

Cabreira, T., Brisolara, L., Ferreira Jr., P.R., 2019. Survey on coverage path planning with unmanned aerial vehicles. Drones 3, 4. https://doi.org/10.3390/drones3010004.

Cabreira, T.M., Kappel, K., de Brisolara, L.B., Ferreira, P.R., 2018. An energy-aware real-time search approach for cooperative patrolling missions with multi-uavs, 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE), Joao Pessoa, Brazil, pp254–259.

Cacace, J., Finzi, A., Lippiello, V., 2014. A mixed-initiative control system for an Aerial Service Vehicle supported by force feedback, 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Illinois, US, pp1230–1235.

Cartwright, N., 1988. How to tell a common cause: generalizations of the conjunctive fork criterion, Fetzer, J.H. (Ed.), Probability and Causality: Essays in Honor of Wesley C. Salmon, Synthese Library. Springer Netherlands, Dordrecht, pp181–188.

Casbeer, D.W., Beard, R.W., McLain, T.W., Mehra, and R.K., 2005. Forest fire monitoring with multiple small UAVs, 2005 American Control Conference, Portland, US, pp3530–3535 vol. 5.

Caywood, M.S., Roberts, D.M., Colombe, J.B., Greenwald, H.S., Weiland, M.Z., 2017. Gaussian process regression for predictive but interpretable machine learning models: an example of predicting mental workload across tasks. Frontiers in Human Neuroscience 10, p647.

Chappell, D., Son, H.W., Clark, A.B., Yang, Z., Bello, F., Kormushev, P., Rojas, N., 2022. Virtual reality pre-prosthetic hand training with physics simulation and robotic force interaction. IEEE Robotics and Automation Letters 7, pp4550–4557.

Chawla, M., Duhan, M., 2018. Levy flights in metaheuristics optimisation algorithms – a review. Applied Artificial Intelligence 32, pp802–821. https://doi.org/10.1080/08839514.2018.1508807

Chawla, M., Duhan, M., 2015. Bat Algorithm: A Survey of the State-of-the-Art. Applied Artificial Intelligence 29, pp617–634.

Chen, L., Xu, J., 2004. OPtimal delaunay triangulations. Journal of Computational Mathematics 22, pp299–308.

Chiappe, D., Rorie, R.C., Morgan, C.A., Vu, K.-P.L., 2014. A situated approach to the acquisition of shared SA in team contexts. Theoretical Issues in Ergonomics Science 15, pp69–87.

Chiappe, D., Vu, K.-P.L., Rorie, C., Morgan, C., 2012. A situated approach to shared Situation Awareness. Proceedings of the Human Factors and Ergonomics Society Annual Meeting 56, pp748–752.

Choi, H., Crump, C., Duriez, C., Elmquist, A., Hager, G., Han, D., Hearl, F., Hodgins, J., Jain, A., Leve, F., Li, C., Meier, F., Negrut, D., Righetti, L., Rodriguez, A., Tan, J., Trinkle, J., 2021. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. National Academy of Sciences, US, 118. https://doi.org/10.1073/pnas.1907856118.

Choxi, H., 2007. A Distributed Constraint Optimisation approach to wireless network optimisation. 2007 Association for the Advancement of Aritificial Intelligence (AAAI), Vancouver, Canada.

Chuvieco, E., Mouillot, F., van der Werf, G.R., San Miguel, J., Tanase, M., Koutsias, N., García, M., Yebra, M., Padilla, M., Gitas, I., Heil, A., Hawbaker, T.J., Giglio, L., 2019. Historical background and current developments for mapping burned area from satellite Earth observation. Remote Sensing of Environment 225, pp45–64.

Cignoni, P., Montani, C., Scopigno, R., 1998. DeWall: A fast divide and conquer Delaunay triangulation algorithm. Computer-Aided Design 30, pp333–341. https://doi.org/10.1016/S0010-4485(97)00082-1

Consciousness, S.A.I.A., Externally Directed, Kip Smith, P. A. Hancock, 1995. Situation awareness is adaptive, externally directed consciousness. Human Factors, pp137-148.

Cooper, J.R., 2020. Optimal multi-agent search and rescue using potential field theory. AIAA Scitech 2020 Forum, American Institute of Aeronautics and Astronautics, Orlando, US.

Cortés, J., Egerstedt, M., 2017. Coordinated control of multi-robot systems: a survey. SICE Journal of Control, Measurement, and System Integration, pp495–503. https://doi.org/10.9746/jcmsi.10.495

Cox, M.T., Zhang, C., 2005. Planning as mixed-initiative goal manipulation, 25th International Conference on Automated Planning and Scheduling, ICAPS'05. AAAI Press, Monterey, California, USA, pp. 282–291.

Cummings, M.L., Bruni, S., Mercier, S., Mitchell, P.J., 2007. Automation architecture for single operator, multiple UAV command and control. International Command and Control Journal.

Cummings, M.L., Mitchell, P.J., 2008. Predicting controller capacity in supervisory control of multiple uavs. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, pp451–460.

Dama, F., Sinoquet, C., 2021. Analysis and modeling to forecast in time series: a systematic review. arXiv:2104.00164 [cs].

Danczyk, J., Eaton, R., Hutchins, R., Jenkins, M., Irvin, S., 2016. Providing Distributed Situation Awareness to human and canine tracking teams, 2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), SanDiego, US, pp73–77.

D'Aniello, G., Gaeta, A., Gaeta, M., Tomasiello, S., 2018. Self-regulated learning with approximate reasoning and situation awareness. Journal of Ambient Intelligent Human Computing 9, pp151–164. https://doi.org/10.1007/s12652-016-0423-y

D'Aniello, G., Gaeta, M., Loia, V., Orciuoli, F., Sampson, D.G., 2015. Situation Awareness enabling decision support in seamless learning, 2015 International Conference on Intelligent Networking and Collaborative Systems, Washington, US, pp440–445. https://doi.org/10.1109/INCoS.2015.59

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39, pp1–38.

Demyen, D., 2006. Efficient triangulation-based pathfinding. AAAI International Conference, Boston, USA.

Desai, J.P., Ostrowski, J., Kumar, V., 1998. Controlling formations of multiple mobile robots. IEEE Access, pp. 2864–2869 vol.4. https://doi.org/10.1109/ROBOT.1998.680621.

Di Franco, C., Buttazzo, G., 2016. Coverage Path planning for UAVs photogrammetry with energy and resolution constraints. Journal of Intelligent and Robotics System 83, pp445–462. https://doi.org/10.1007/s10846-016-0348-x

Donald Knuth, 1997. The art of computer programming, Third edition. Addison-Wesley, Boston, US.

Drew, D.S., 2021. Multi-Agent systems for search and rescue applications. Current Robotics Report 2, 189–200. https://doi.org/10.1007/s43154-021-00048-3

Durso, F.T., Sethumadhavan, A., 2008. Situation Awareness: understanding dynamic environments. Hum Factors 50, pp442–448. https://doi.org/10.1518/001872008X288448

Endsley, M.R., 2015. Situation Awareness misconceptions and misunderstandings. Journal of Cognitive Engineering and Decision Making 50, pp442–448. https://doi.org/10.1518/001872008X288448

Endsley, M.R., 2000. Direct measurement of situation awareness: Validity and use of SAGAT, Situation Awareness Analysis and Measurement. Lawrence Erlbaum Associates Publishers, Mahwah, New Jersey, US, pp147–173.

Endsley, M.R., 1999. Situation awareness in aviation systems. Handbook of Aviation Human Factors, Human Factors in Transportation. Lawrence Erlbaum Associates Publishers, Mahwah, New Jersey, US, pp257–276.

Endsley, M.R., 1995. Toward a theory of Situation Awareness in Dynamic Systems. The Journals of the Human Factors and Ergonomics. https://doi.org/10.1518/001872095779049543

Endsley, M.R., 1988. Design and evaluation for situation awareness enhancement. Human Factors Society Proceedings.

Endsley, M.R., Jones, W.M., 1996. Situation Awareness information dominance & information warfare. Logicon Technical Services Inc Dayton OH.

Erdelj, M., Natalizio, E., Chowdhury, K.R., Akyildiz, I.F., 2017. Help from the sky: leveraging UAVs for disaster management. IEEE Pervasive Computing 16, pp24–32. https://doi.org/10.1109/MPRV.2017.11

Etemadi, N., 1981. An elementary proof of the strong law of large numbers. Z. Wahrscheinlichkeitstheorie verw Gebiete 55, pp119–122. https://doi.org/10.1007/BF01013465.

Ferguson, G., Allen, J., 2007. Mixed-initiative systems for collaborative problem solving. AI Magazine, 1 p28, https://doi.org/10.1609/aimag.v28i2.2037

Feyerabend, P.K., 1959. Hans Reichenbach, The direction of time. The British Journal for the Philosophy of Science 9, pp336–337. https://doi.org/10.1093/bjps/IX.36.336

Fioretto, F., Pontelli, E., Yeoh, W., 2018. Distributed Constraint Optimisation Problems and applications: a survey. Journal of Artificial Intelligence Research, 61, pp623–698.

Fioretto, F., Yeoh, W., Pontelli, E., 2017. A multiagent system approach to scheduling devices in smart homes, 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS '17. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp981–989.

Fioretto, F., Yeoh, W., Pontelli, E., 2015. Multi-variable agents decomposition for DCOPs to exploit multi-level parallelism. AAMAS extended abstract, Volume 2, Istanbul, Turkey.

Fire Lookout History of the Santa Fe National Forest, 2017. Santa Fe National Forest.

Foushee, H.C., Helmreich, R.L., 1988. Group interaction and flight crew performance. Human Factors in Aviation, Academic Press Series in Cognition and Perception. Academic Press, San Diego, CA, US, pp. 189–227.

Fransman, J., Sijs, J., Dol, H., Theunissen, E., De Schutter, B., 2019. Bayesian-DPOP for continuous Distributed Constraint Optimisation Problems. 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 1961–1963.

FRODO: a FRamework for Open/Distributed Optimisation Version 2.17.1 User Manual PDF Version, n.d.

Gage, A., Murphy, R.R., 2004. Affective recruitment of distributed heterogeneous agents. 19th National Conference on Artifical Intelligence, AAAI'04. AAAI Press, San Jose, California, pp14–19.

Galland, S., Gaud, N., Yasar, A.-U.-H., Knapen, L., Janssens, D., Lamotte, O., 2013. Simulation model of carpooling with the janus multiagent platform. 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013) 19, pp860–866. https://doi.org/10.1016/j.procs.2013.06.115

Galland, S., Knapen, L., Yasar, A.-U.-H., Gaud, N., Janssens, D., Lamotte, O., Koukam, A., Wets, G., 2014. Multi-agent simulation of individual mobility behavior in carpooling. Transportation Research Part C: Emerging Technologies, Advances in Computing and Communications and their Impact on Transportation Science and Technologies 45, pp83–98.

Galton, A., Worboys, M., 2011. An Ontology of Information for Emergency Management. 8th International Conference on Information System for Crisis Response and Management, Lisbon, Portugal, pp1–10.

Gan, L.K., Zhang, P., Lee, J., Osborne, M.A., Howey, D.A., 2021. Data-driven energy management system with gaussian process forecasting and mpc for interconnected microgrids. IEEE Transactions on Sustainable Energy, pp695–704.

Ganoni, O., Mukundan, R., 2017. A framework for visually realistic multi-robot simulation in natural environment. Robotics. arxiv:1708.01938 [cs].

Ghamry, K., Zhang, Y., 2016. Cooperative control of multiple UAVs for forest fire monitoring and detection. 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), Auckland, New Zealand pp1–6.

Glymour, C., 1999. Rabbit hunting. Synthese 121, 55–78. https://doi.org/10.1023/A:1005229730590

Gómez, C., Green, D.R., 2017. Small unmanned airborne systems to support oil and gas pipeline monitoring and mapping. Arab Journal of Geoscience 10, 202. https://doi.org/10.1007/s12517-017-2989-x

Görtler, J., Kehlbeck, R., Deussen, O., 2019. A visual exploration of gaussian processes. Distill 4. https://doi.org/10.23915/distill.00017

Greche, L., Jazouli, M., Es-Sbai, N., Majda, A., Zarghili, A., 2017. Comparison between Euclidean and Manhattan distance measure for facial expressions classification. 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), Fez, Morocco, pp1-4. https://doi.org/10.1109/WITS.2017.7934618

Hackney, C.R., Clayton, A.I., 2015. Unmanned Aerial Vehicles (UAVs) and their application in geomorphic mapping,. Geomorphological Techniques. British Society for Geomorphology, IEEE, Madrid, pp1067–1074. https://doi.org/10.1109/IROS.2018.8593539.

Haksar, R.N., Schwager, M., 2018. Distributed deep reinforcement learning for fighting forest fires with a network of aerial robots. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, pp1067–1074.

Hale, J.Q., Zhou, E., 2015. A Model-based approach to multi-objective optimisation. 2015 Winter Simulation Conference, WSC ’15. IEEE Press, Piscataway, New Jersey, USA, pp3599–3609.

Hasegawa, G., Takemori, S., Taniguchi, Y., Nakano, H., 2012. Determining coverage area using voronoi diagram based on local information for wireless mesh networks. 9th International Conference on Information Technology, pp71–76. https://doi.org/10.1109/ITNG.2012.19

Hausman, D.M., Woodward, J., 1999. Independence, invariance and the causal markov condition. The British Journal for the Philosophy of Science 50, pp521–583. https://doi.org/10.1093/bjps/50.4.521

Heintzman, L., Hashimoto, A., Abaid, N., Williams, R.K., 2021. Anticipatory Planning and dynamic lost person models for human-robot search and rescue. 2021 IEEE International Conference on Robotics and Automation (ICRA), Shaanxi, China, pp8252–8258. https://doi.org/10.1109/ICRA48506.2021.9562070

Hendikawati, P., Subanar, Abdurakhman, Tarno, 2020. A survey of time series forecasting from stochastic method to soft computing. Journal of Physics: Conference Series 1613, p012019. https://doi.org/10.1088/1742-6596/1613/1/012019

Herzig, A., Lorini, E., Perrussel, L., Xiao, Z., 2017. BDI Logics for BDI architectures: old problems, new perspectives. Künstl Intell, 31, pp73–83. https://doi.org/10.1007/s13218-016-0457-5

Hitchcock, C., Rédei, M., 2020. Reichenbach's common cause principle. The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, California, US.

Hoang, K., 2019. Proactive Distributed Constraint Optimisation problems. Journal of Artificial Intelligence Research.

Hoang, K.D., Fioretto, F., Hou, P., Yokoo, M., Yeoh, W., Zivan, R., 2016. Proactive dynamic Distributed Constraint Optimisation. 2016 International Conference on Autonomous Agents & Multiagent System, Singapore.

Hoang, K.D., Hou, P., Fioretto, F., Yeoh, W., Zivan, R., Yokoo, M., 2017. Infinite-horizon proactive dynamic DCOPs. 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS '17. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 212–220.

Hocraffer, A., Nam, C.S., 2017. A meta-analysis of human-system interfaces in unmanned aerial vehicle (UAV) swarm management. Applied Ergonomics, pp66–80.

Hsu, P.L., Robbins, H., 1947. Complete convergence and the law of large numbers. Proc Natl Acad Sci U S A 33, pp25–31.

Huang, K., King, I., Lyu, M.R., 2003. Discriminative training of Bayesian Chow-Liu multinet classifiers. International Joint Conference on Neural Networks, 2003. pp. 484–488 vol.1. https://doi.org/10.1109/IJCNN.2003.1223394

Huang, W.H., 2001. Optimal line-sweep-based decompositions for coverage algorithms. IEEE International Conference on Robotics and Automation (ICRA), pp. 27–32 vol.1. https://doi.org/10.1109/ROBOT.2001.932525

Hutchins, E., 2001. Cognition, distributed, in: smelser, International Encyclopedia of the Social & Behavioral Sciences. Pergamon, Oxford, pp2068–2072. https://doi.org/10.1016/B0-08-043076-7/01636-3

Hutchins, E., 1995. How a cockpit remembers its speeds. Cognitive Science, pp265-288.

Ingle, L.B., 2011. Every day is fire day: a study of historic fire towers and lookout life in the great smoky mountains National Park, Masters Thesis, Clemson University, p136.

International Forest Fire News, 2006. International Forest Fire News (IFFN) No. 34.

Jensen-Nau, K.R., Hermans, T., Leang, K.K., 2021. Near-optimal area-coverage path planning of energy-constrained aerial robots with application in autonomous environmental monitoring. IEEE Transactions on Automation Science and Engineering 18, p1453–1468.

João, F., 2012. Search Patterns. Report

Jones, D.G., Endsley, M.R., 1996. Sources of situation awareness errors in aviation. Aviat Space Environ Med 67, pp507–512.

Kallmann, M., n.d. Path Planning in Triangulations. International Conference on Artificial (IJCAI) workshop on reasoning, representation, and learning in computer games, Edinburgh, Scotland.

Kanistras, K., Martins, G., Rutherford, M.J., Valavanis, K.P., 2013. A survey of unmanned aerial vehicles (UAVs) for traffic monitoring. International Conference on Unmanned Aircraft Systems (ICUAS), Geogia, US, pp221–234. https://doi.org/10.1109/ICUAS.2013.6564694

Kappel, K.S., Cabreira, T.M., Marins, J.L., de Brisolara, L.B., Ferreira, P.R., 2020. Strategies for patrolling missions with multiple UAVs. Journal of Intelligent and Robotic Systems 99, pp499–515.

Karduni, A., Markant, D., Wesslen, R., Dou, W., 2021. A Bayesian cognition approach for belief updating of correlation judgement through uncertainty visualizations. IEEE Transactions on Visualization and Computer Graphics 27, pp978–988. https://doi.org/10.1007/s10846-019-01090-2

Khan, A., Yanmaz, E., Rinner, B., 2015. Information exchange and decision making in micro aerial vehicle networks for cooperative search. IEEE Transactions on Control of Network Systems 2, pp335–347. https://doi.org/10.1109/TCNS.2015.2426771

Khan, A., Yanmaz, E., Rinner, B., 2014. Information merging in multi-UAV cooperative search. IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, pp3122–3129.

Khan, F., F., O., Sreenuch, T., Tsourdos, A., 2014. Multi-domain modeling and simulation of an aircraft system for advanced vehicle-level reasoning research and development. IJACSA 5. https://doi.org/10.14569/IJACSA.2014.050414

Kho, J., 2009. Decentralised control of wireless sensor networks. University of Southampton, PhD Thesis.

Kitchin, J., Baber, C., 2017. The dynamics of Distributed Situation Awareness. Human Factors and Ergonomics Society Annual Meeting 61, Auston, US, pp277–281.

Kitchin, J., Baber, C., 2016. A comparison of shared and distributed situation awareness in teams through the use of agent-based modelling. Theoretical Issues in Ergonomics Science, pp277–281. https://doi.org/10.1177/1541931213601551

Kluegel, W., Iqbal, M.A., Fioretto, F., Yeoh, W., Pontelli, E., 2017. A realistic dataset for the smart home device scheduling problem for DCOPs, 2017 international Conference on Autonomous Agents and Multi-agent System, Sao Paulo, Brazil, pp125-142.

Koenig, S., Liu, Y., 2001. Terrain coverage with ant robots: a simulation study. 5th International Conference on Autonomous Agents.

Kolling, A., Walker, P., Chakraborty, N., Sycara, K., Lewis, M., 2016. Human Interaction With Robot Swarms: A Survey. IEEE Transactions on Human Machine Systems, pp9–26.

Kokar, M.M., Matheus, C.J., Baclawski, K., 2009. Ontology-based situation awareness. Information Fusion, Special Issue on High-level Information Fusion and Situation Awareness 10, pp83–98.

Kumar, V., 2007. Chapman & hall/crc data mining and knowledge discovery series 64.

Kunda, Z., 1986. Prediction and the partial understanding of the law of large numbers. Journal of Experimental Social Psychology, pp339–354

Lähdesmäki, H., Hautaniemi, S., Shmulevich, I., Yli-Harja, O., 2006. Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. Signal Processing 86, pp814–834.

Larranaga, P., Sierra, B., Gallego, M.J., Michelena, M.J., Picaza, J.M., 1997. Learning Bayesian networks by genetic algorithms. a case study in the prediction of survival in malignant skin melanoma. International Conference on Artificial Intelligence in Medicine, Grenoble, France. https://doi.org/10.1007/bfb0029459

Le, T., Fioretto, F., Yeoh, W., Son, T.C., Pontelli, E., 2016. ER-DCOPs: A Framework for Distributed Constraint Optimisation with uncertainty in constraint utilities. 15th International Conference on Autonomous Agents and Multiagent Systems, Singapore.

Le, T.D., Hoang, T., Li, J., Liu, L., Liu, H., 2019. A fast PC algorithm for high dimensional causal discovery with multi-core PCs. IEEE/ACM Transactions on Computational Biology and Bioinformatics 16, pp1483–1495. https://doi.org/10.1109/TCBB.2016.2591526

Lee, K.D., Gelfand, A.E., Wiesenfeld, E., Stepnitz, B., 2007. Introduction of the hybrid inference tool (HIT). 10th International Conference on Information Fusion, pp. 1–7. https://doi.org/10.1109/ICIF.2007.4408116

Leith, D.J., Heidl, M., Ringwood, J.V., 2004. Gaussian process prior models for electrical load forecasting. International Conference on Probabilistic Methods Applied to Power Systems, pp. 112–117.

Levin, E., Pieraccini, R., Eckert, W., 2000. A stochastic model of human-machine interaction for learning dialog strategies. IEEE Transactions on Speech and Audio Processing 8, pp11–23. https://doi.org/10.1109/89.817450

Li, Cabeli, V., Sella, N., Isambert, H., 2019. Constraint-based causal structure learning with consistent separating sets. Advances in Neural Information Processing Systems. Curran Associates, Inc.

Li, F., Ding, Y., Hao, K., 2015. A dynamic leader-follower strategy for multi-robot systems. 2015 IEEE International Conference on Systems, Man, and Cybernetics, pp. 298–303.

Li, M., Yang, W., Cai, Z., Yang, S., Wang, J., 2019. Integrating decision sharing with prediction in decentralized planning for multi-agent coordination under uncertainty. 28th International Joint Conference on Artificial Intelligence, Macao, China, pp450-456.

Li, P., Zhang, L., Dai, L., Zou, Y., Li, X., 2018. An assessment method of operator's situation awareness reliability based on fuzzy logic-AHP. Safety Science p119. https://doi.org/10.1016/j.ssci.2018.08.007

Li, Y., Chen, H., Joo Er, M., Wang, X., 2011. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. Mechatronics, Special Issue on Development of Autonomous Unmanned Aerial Vehicles 21, pp876–885.

Liu, S., Brewster, C., Shaw, D., 2013. Ontologies for crisis management: a review of state of the art in ontology design and usability. 10th International ISCRAM Conference – Baden-Baden, Germany, pp. 349–359.

Lohia, P., Kannan, K., Srivastava, B., Mehta, S., 2019. Design diagrams as ontological source. 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2019. Association for Computing Machinery, New York, USA, pp 863–873.

Lopes, J., Souza, R., Gadotti, G., Pernas, A., Yamin, A., Geyer, C., 2014. An Architectural Model for Situation Awareness in Ubiquitous Computing. IEEE Latin America Transactions 12, pp1113–1119.

Lu, B., Coombes, M., Li, B., Chen, W., 2016. Improved Situation Awareness for autonomous taxiing through self-learning. IEEE Transactions on Intelligent Transportation Systems 17, pp3553–3564.

Lu, Q., Karanikolas, G., Shen, Y., Giannakis, G.B., 2020. Ensemble Gaussian processes with spectral features for online interactive learning with scalability. International Conference on Artificial Intelligence and Statistics, PMLR, pp. 1910–1920.

Lumelsky, V.J., Harinarayan, K.R., 1997. Decentralized motion planning for multiple mobile robots: the cocktail paarty model. Autonomous Robots 4, p121–135.

Madsen, A.L., Jensen, F., Salmerón, A., Langseth, H., Nielsen, T.D., 2017. A parallel algorithm for Bayesian network structure learning from large data sets. Variety and Velocity in Data Science 117, pp46–55.

Maheswaran, R., Pearce, J.P., Tambe, M., 2004. Distributed algorithms for DCOP: a graphical-game-based approach. International Conference on Parallel and Distributed Computing Systems, California, USA.

Maheswaran, R.T., Tambe, M., Bowring, E., Pearce, J.P., Varakantham, P., 2004. Taking DCOP to the real world: efficient complete solutions for distributed multi-event scheduling. 3rd International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS '04. IEEE Computer Society, Washington DC, USA, pp310–317.
Makonin, S., McVeigh, D., Stuerzlinger, W., Tran, K., Popowich, F., 2016. Mixed-initiative for big data: the intersection of human + visual analytics + prediction. 49th Hawaii International Conference on System Sciences (HICSS), pp1427–1436.
Mandt, S., Hoffman, M.D., 2017. Stochastic gradient descent as approximate bayesian inference. Journal of Machine Learning Research 5.
Marcano, M., Díaz, S., Pérez, J., Irigoyen, E., 2020. A Review of shared control for automated vehicles: theory and applications. IEEE Transactions on Human-Machine Systems, pp475–491.

Marjovi, A., Nunes, J.G., Marques, L., Almeida, A.T. de, 2009. Multi-robot exploration and fire searching. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp1929–1934.
Matthews, M.D., Beal, S.A., 2002. Assessing Situation Awareness in field training exercises. Military Academy West Point Ny Office of Military Psychology and Leadership.
McCabe, T.J., 1976. A complexity measure. IEEE Transactions on Software Engineering SE-2, pp308–320.
McCaskey, U., von Aster, M., O'Gorman, R., Kucian, K., 2020. Persistent differences in brain structure in developmental dyscalculia: a longitudinal morphometry study. Frontiers in Human Neuroscices 14, p272.
McLain, T.W., Chandler, P.R., Rasmussen, S., Pachter, M., 2001. Cooperative control of UAV rendezvous, 2001 American Control Conference. (Cat. No.01CH37148), pp. 2309–2314 vol.3.
Meloni, A., Ripoli, A., Positano, V., Landini, L., 2009. Mutual Information preconditioning improves structure learning of bayesian networks from medical databases. IEEE Transactions on Information Technology in Biomedicine, pp984–989.
Merino, L., Caballero, F., Dios, J.R.M., Ferruz, J., Ollero, A., 2006. A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires. Journal of Field Robotics, pp165–184.
Merino, L., Caballero, F., Martínez-de-Dios, J.R., Maza, I., Ollero, A., 2010. Automatic forest fire monitoring and measurement using unmanned aerial vehicles. 2010 International Conference on Forest Fire Research, Viegas.
Minsky, M., 1987. The society of mind, First. edition. William Heinemann Ltd, London, UK.
Mohd Daud, S.M.S., Mohd Yusof, M.Y.P., Heo, C.C., Khoo, L.S., Chainchel Singh, M.K., Mahmood, M.S., Nawawi, H., 2022. Applications of drone in disaster management: A scoping review. Science and Justice 62, pp30–42.
Monesi, A., Imbriaco, G., Mazzoli, C.A., Giugni, A., Ferrari, P., 2022. In-situ simulation for intensive care nurses during the covid-19 pandemic in italy: advantages and challenges. Clinical Simulation in Nursing, pp52–56.
Morgan, M.G., Henrion, M., 1993. Uncertainty: a guide to dealing with uncertalnty in quantitative risk and policy analysis. Journal Devoted to the Problems of Capital Investment, 38(4).

Muñoz, J., López, B., Quevedo, F., Monje, C.A., Garrido, S., Moreno, L.E., 2021. Coverage strategy for target location in marine environments using fixed-wing UAVs. Drones 5, p120.

Murray, B., Perera, L.P., 2022. Ship behavior prediction via trajectory extraction-based clustering for maritime situation awareness. Journal of Ocean Engineering and Science 7, pp1–13.

N. v., K., G. a., L., M. v., Y., R. v., Y., 2017. Hidden attractors in dynamical models of phase-locked loop circuits: Limitations of simulation in MATLAB and SPICE. Communications in Nonlinear Science and Numerical Simulation, pp39–49.

Nasir, J., Islam, F., Malik, U., Ayaz, Y., Hasan, O., Khan, M., Muhammad, M.S., 2013. RRT*-SMART: A rapid convergence implementation of RRT*. International Journal of Advanced Robotic Systems, 10, p299. https://doi.org/10.5772/56718.

Nasirian, B., Mehrandezh, M., Janabi-Sharifi, F., 2021. Efficient coverage path planning for mobile disinfecting robots using graph-based representation of environment. Frontiers in AI and Robotics 8. https://doi.org/10.3389/frobt.2021.624333.

Nath, P., Saha, P., Middya, A.I., Roy, S., 2021. Long-term time-series pollution forecast using statistical and deep learning methods. Neural Computing and Application. https://doi.org/10.1007/s00521-021-05901-2

Neapolitan, R.E., 1990. Probabilistic reasoning in expert systems: theory and algorithms. John Wiley & Sons, Inc., USA.

Nebel, B., Bolander, T., Engesser, T., Mattmüller, R., 2019. Implicitly coordinated multi-agent path finding under destination uncertainty: success guarantees and computational complexity. Journal of Artificial Intelligence Research, 64, pp497–527. https://doi.org/10.1613/jair.1.11376.

Nguyen, D.T., Yeoh, W., Lau, H.C., Zilberstein, S., Zhang, C., 2014. Decentralized multi-agent reinforcement learning in average-reward dynamic DCOPs, 28th AAAI Conference on Artificial Intelligence, Quebec, Canada, pp.1447–1455.

Nguyen, T., Lim, C.P., Nguyen, N.D., Gordon-Brown, L., Nahavandi, S., 2019. A review of situation awareness assessment approaches in aviation environments. IEEE Systems Journal, pp3590–3603.

Noreen, I., Khan, A., Habib, Z., 2016. A comparison of RRT, RRT* and RRT*-smart path planning algorithms. International Journal of Computer Science and Network Security, 8.

Norstein, E., Sharma, A., Jungefeldt, S.S.M., Nazir, S., 2019. Distributed situation awareness in a demanding maritime operation: A case study of the subsea segment. International Journal of Marine Navigation and Safety of Sea Transportation, pp811-822.

Nurzaman, S.G., Matsumoto, Y., Nakamura, Y., Koizumi, S., Ishiguro, H., 2009. Biologically inspired adaptive mobile robot search with and without gradient sensing, IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, St. Louis, MO, USA, pp142–147.

Ozkan, O., Kilic, S., 2022. UAV routing by simulation-based optimisation approaches for forest fire risk mitigation. Annals of Operation Research.

Papastefanopoulos, V., Linardatos, P., Kotsiantis, S., 2020. COVID-19: A comparison of time series methods to forecast percentage of active cases per population. Applied Sciences , p3880.

Park, C.Y., Laskey, K.B., Costa, P.C.G., Matsumoto, S., 2016. A process for human-aided Multi-Entity Bayesian Networks learning in Predictive Situation Awareness, 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, pp2116–2124.

Park, C.Y., Laskey, K.B., Costa, P.C.G., Matsumoto, S., 2013. Multi-Entity Bayesian networks learning for hybrid variables in situation awareness. 16th International Conference on Information Fusion, Istanbul, Turkey, pp1894–1901.

Patil, C., Baidari, I., 2019. Estimating the optimal number of clusters k in a dataset using data depth. Data Science and Engineering, pp132–140.

Pavlin, G., de Oude, P., Maris, M., Nunnink, J., Hood, T., 2010. A multi-agent systems approach to distributed bayesian information fusion. Information Fusion, Agent-Based Information Fusion 11, pp267–282.

Pearl, J., 1988. Probabilistic Reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Pearl, J., 1978. An economic basis for certain methods of evaluating probabilistic forecasts. International Journal of Man-Machine Studies, 10, pp175–183. https://doi.org/10.1016/S0020-7373(78)80010-8

Pearson, R., Donnelly, M.P., Jun Liu, Galway, L., 2016. Generic application driven situation awareness via ontological situation recognition. IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), San Diego, Canada, pp131–137.

Perera, S., Barnes, N., 2011. A Simple and practical solution to the rigid body motion segmentation problem using a RGB-D camera. International Conference on Digital Image Computing: Techniques and Applications(DICTA), Queensland, Australia.

Peter Hirschberger, 2016. Forests ablaze: Causes and effects of global forest fires. 2017 Edition. Deutchland, Berlin.

Qi, X., Fan, X., Wang, H., Lin, L., Gao, Y., 2021. Mutual-information-inspired heuristics for constraint-based causal structure learning. Information Sciences 560, pp152–167. https://doi.org/10.1016/j.ins.2020.12.009.

Quintin, F., Iovino, S., Savvaris, A., Tsourdos, A., 2017. Use of co-operative uavs to support/augment UGV situational awareness and/or inter-vehicle communications. IFAC-PapersOnLine, 20th IFAC World Congress 50, Toulouse, France, pp8037–8044.

Rabinovich, S., Curry, R.E., Elkaim, G.H., 2018. Toward dynamic monitoring and suppressing uncertainty in wildfire by multiple unmanned air vehicle system. Journal of Robotics. https://doi.org/10.1155/2018/689215

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian processes for machine learning, adaptive computation and machine learning. MIT Press, Cambridge.

Rathbun, D., Kragelund, S., Pongpunwattana, A., Capozzi, B., 2002. An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments. 21st Digital Avionics Systems Conference, Irvine, US.

Raymundo, C.R., Costa, P.D., Almeida, J.P.A., Pereira, I., 2014. An infrastructure for distributed rule-based situation management. IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), San Antonio, US, pp. 202–208.

Revach, G., Greshler, N., Shimkin, N., 2017. Planning for cooperative multiple agents with sparse interaction constraints. Thesis 9.

Reynolds, C.W., 1987. Flocks, Herds and Schools: A distributed behavioral model. 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87. ACM, New York, USA, pp25–34.

Rocha, A.M., Casau, P., Cunha, R., 2022. A control algorithm for early wildfire detection using aerial sensor networks: modeling and simulation. Drones, p44.

Romanycia, M., 2019. Netica-J Reference Manual. Norsys Software Corp. Canada

Rosário, C.R., Amaral, F.G., Kuffel, F.J.M., Kipper, L.M., Frozza, R., 2021. Using Bayesian belief networks to improve distributed situation awareness in shift changeovers: a case study. Expert Systems with Applications 188, p116039. https://doi.org/10.1016/j.eswa.2021.116039

Salerno, J., Boulware, D., Cardillo, R., 2005. Knowledge representation requirements for situation awareness. 7th International Conference on Information Fusion, Stockholm, Sweden.

Salmon, P.M., Plant, K.L., 2022. Distributed situation awareness: from awareness in individuals and teams to the awareness of technologies, sociotechnical systems, and societies. Applied Ergonomic, 98, p103599. https://doi.org/10.1016/j.apergo.2021.103599

Salmon, P.M., Stanton, N.A., Jenkins, D.P., 2009. Distributed Situation Awareness: theory, measurement and application, First Edition, Ashgate, Farnham, UK.

Salmon, P.M., Stanton, N.A., Walker, G.H., Baber, C., Jenkins, D.P., McMaster, R., Young, M.S., 2008. What really is going on? review of situation awareness models for individuals and teams. Theoretical Issues in Ergonomics Science, 9, pp297–323. https://doi.org/10.1080/14639220701561775

Salmon, P.M., Stanton, N.A., Walker, G.H., Baber, C., McMaster, R., Jenkins, D., Beond, A., Sharif, O., Rafferty, L., Ladva, D., 2006. Distributed Situation Awareness in command and control: a case study in the energy distribution domain. Human Factors and Ergonomics Society Annual Meeting, 50, pp260–264. https://doi.org/10.1177/154193120605000311

Salmon, P.M., Walker, G.H., Stanton, N.A., 2015. Broken components versus broken systems: why it is systems not people that lose situation awareness. Cognition, Technology and Work, 17, pp179–183. https://doi.org/10.1007/s10111-015-0324-4

Salmon, W.C., 1984. Scientific explanation and the causal structure of the world. Princeton University Press, New Jersey, US.

Sanz-Pena, I., Blanco, J., Kim, J.H., 2021. Computer Interface for real-time gait biofeedback using a wearable integrated sensor system for data acquisition. IEEE Transactions on Human-Machine Systems, 51, pp484–493. https://doi.org/10.1109/THMS.2021.3090738

Sarter, N.B., Woods, D.D., 1991. Situation Awareness: A critical but ill-defined phenomenon. The International Journal of Aviation Psychology 1, pp45–57. https://doi.org/10.1207/s15327108ijap0101_4.

Sathe, S., Papaioannou, T.G., Jeung, H., Aberer, K., 2013. A survey of model-based sensor data acquisition and management. Managing and Mining Sensor Data. Springer US, Boston, US, pp9–50.

Sauter, J.A., Matthews, R., Van Dyke Parunak, H., Brueckner, S.A., 2005. Performance of digital pheromones for swarming vehicle control. 4th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '05. Association for Computing Machinery, New York, USA, pp903–910.

Scanagatta, M., Salmerón, A., Stella, F., 2019. A survey on Bayesian network structure learning from data. Programming and Artificial Intelligence 8, pp425–439. https://doi.org/10.1007/s13748-019-00194-y

Schloss, R., Usceda-Sosa, R., Srivastava, B., 2014. The D-SCRIBE process for building a scalable ontology. 25th AAAI Conference on Artificial Intelligence, Sansfrancisco, US.

Schurz, G., 2017. Interactive causes: revising the markov condition. Philosophy of Science, 84, pp456–479. https://doi.org/10.1086/692143

Schwab, J.D., Kühlwein, S.D., Ikonomi, N., Kühl, M., Kestler, H.A., 2020. Concepts in Boolean network modeling: what do they all mean? Computational and Structural Biotechnology Journal 18, pp571–582. https://doi.org/10.1016/j.csbj.2020.03.001

Schwaighofer, A., Tresp, V., Yu, K., 2004. Learning Gaussian process kernels via hierarchical Bayes. In Advances in Neural Information Processing Systems, pp1209–1216.

Scutari, M., 2015. Bayesian network constraint-based structure learning algorithms: parallel and optimised implementations in the bnlearn r package. Journal of Statistical Software 77, p2.

Scutari, M., Graafland, C.E., Gutiérrez, J.M., 2019. Who learns better bayesian network structures: accuracy and speed of structure learning algorithms. International Journal of Approximate Reasoning 115, pp235-253.

Shannon, C.E., 1959. A mathematical theory of communication. The Bell System Technical Journal 27, pp379 -423.

Shumway, R.H., 1984. Some Applications of the EM algorithm to analyzing incomplete time series data. Time Series Analysis of Irregularly Observed Data, Lecture Notes in Statistics. Springer, New York, pp290–324.

Skotarczak, E., Dobek, A., Moliński, K., 2018. Entropy as a measure of dependency for categorized data. Biometrical Letters, 55, pp233–243. https://doi.org/10.2478/bile-2018-0014

Smith, D.M., 2017. Sustainability and wildland fire the origins of forest service wildland fire research. Rocky Mountain Research Station 128, Wasginton DC, p20.

So, A.M.-C., Ye, Y., 2005. On solving coverage problems in a wireless sensor network using voronoi diagrams. Internet and Network Economics, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp584–593.

Spirtes, P., Glymour, C.N., Spirtes, P., Glymour, C., 1991. An algorithm for fast recovery of sparse causal graphs. Social Science Computer Review, pp584–593. https://doi.org/10.1007/11600930_58

Stanton, N.A., 2016. Distributed situation awareness. Theoretical Issues in Ergonomics Science, 17, pp1–7. https://doi.org/10.1080/1463922X.2015.1106615

Stanton, N. A, Chambers, P.R.G., Piggott, J., 2001. Situational awareness and safety. Safety Science, 39, pp189–204. https://doi.org/10.1016/S0925-7535(01)00010-8

Stanton, N.A, Chambers, P.R.G., Piggott, J., 2001. Situational awareness and safety. Safety Science, 39, pp189–204. https://doi.org/10.1016/S0925-7535(01)00010-8

Stanton, N.A., Salmon, P.M., Walker, G.H., Jenkins, D.P., 2009. Genotype and phenotype schemata as models of situation awareness in dynamic command and control team. International Journal of Industrial Ergonomics 39. pp480-489. https://doi.org/10.1016/j.ergon.2008.10.003

Stanton, N.A., Salmon, P.M., Walker, G.H., Salas, E., Hancock, P.A., 2017. State-of-science: situation awareness in individuals, teams and systems. Ergonomics, 60, pp449–466. https://doi.org/10.1080/00140139.2017.1278796

Stanton, Stewart, R., Harris, D., Houghton, R.J., Baber, C., McMaster, R., Salmon, P., Hoyle, G., Walker, G., Young, M.S., Linsell, M., Dymott, R., Green, D., 2006. Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. Ergonomics 49, pp1288–1311.

Stefanidi, Z., Margetis, G., Ntoa, S., Papagiannakis, G., 2022. Real-time adaptation of context-aware intelligent user interfaces, for enhanced Situational Awareness. IEEE Access, Multidisplinary, pp23367–23393.

Stewart, R., Stanton, N., Harris, D., Baber, C., Salmon, P., Mock, M., Tatlock, K., Wells, L., Kay, A., 2008. Distributed Situation Awareness in an airborne warning and control system. Application of Novel Ergonomics Methodology. Cognition, Technology and Work, 10. pp221–229. https://doi.org/10.1007/s10111-007-0094-8

Stranders, R., Farinelli, A., Rogers, A., Jennings, N.R., 2009. Decentralised coordination of continuously valued control parameters using the max-sum algorithm. 8th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), Budapest, Hungary.

Suhail, S., Malik, S.U.R., Jurdak, R., Matulevičius, R., 2022. Towards situational aware cyber-physical systems: a security-enhancing use case of blockchain-based digital twins. Computers in Industry 141.

Sulistyawati, K., Wickens, C.D., Chui, Y.P., 2011. Prediction in Situation Awareness: confidence bias and underlying cognitive abilities. The International Journal of Aviation Psychology 21, pp153–174. https://doi.org/10.1080/10508414.2011.556492

Sutantyo, D.K., Kernbach, S., Nepomnyashchikh, V.A., Levi, P., 2011. Multi-Robot searching algorithm using levy flight and artificial potential field. 8th IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR-2010), Bremen, Germany.

Sutantyo, D.K., Kernbach, S., Nepomnyashchikh, V.A., Levi, P., 2011. Multi-robot searching algorithm using levy flight and artificial potential field. 8th International Workshop on Safety, Security, and Rescue Robotics, Bremen, Germany..

Tandon, H., Ranjan, P., Chakraborty, T., Suhag, V., 2020. Coronavirus (COVID-19): ARIMA based time-series analysis to forecast near future. Quantitative Biolog, Springer.

Tecuci, G., Boicu, M., Cox, M.T., 2007. Seven aspects of mixed-initiative reasoning:an introduction to this special issue on mixed-initiative assistants. AI Magazine 28. p11-12.

Teichmann, M., Motus, L., 2021. Situation Awareness, Mental Models and Understanding. 11th IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA), Tallin, Estonia, pp86–93.

Thorndike, R.L., 1953. Who belongs in the family. Psychometrika 18, pp267–276.

Turpin, M., Michael, N., Kumar, V., 2014. Capt: Concurrent assignment and planning of trajectories for multiple robots. The International Journal of Robotics Research 33, pp98–112. https://doi.org/10.1177/0278364913515307

Ucgun, H., Yuzgec, U., Bayilmis, C., 2021. A review on applications of rotary-wing unmanned aerial vehicle charging stations. International Journal of Advanced Robotic Systems 18. https://doi.org/10.1177/17298814211015863

Uma Pavan Kumar Kethavarapu, S. Saraswathi, 2016. Concept based dynamic ontology creation for job recommendation system. Procedia Computer Science 85, pp915–921. https://doi.org/10.1016/j.procs.2016.05.282

Vagale, A., Oucheikh, R., Bye, R.T., Osen, O.L., Fossen, T.I., 2021. Path planning and collision avoidance for autonomous surface vehicles: a review. Journal of Marine Science and Technology 26(2), pp1292–1306.

Van Wilgen, B.W., 2009. The evolution of fire management practices in savanna protected areas in South Africa. South African Journal of Science, 105, pp343–349.

Varty, Z., 2017. Simulated Annealing Overview. Lancaster University, Lancaster UK.

Vásárhelyi, G., Virágh, C., Somorjai, G., Tarcai, N., Szörényi, T., Nepusz, T., Vicsek, T., 2014. Outdoor flocking and formation flight with autonomous aerial robots. 2014 International Workshop on Intelligent Robots and System, Chicago, USA.

Vasile, M., Zuiani, F., 2011. Multi-agent collaborative search : an agent-based memetic multi-objective optimisation algorithm applied to space trajectory design. Journal of Aerospace Engineering 225, pp1211–1227.

Verfaillie, G., Jussien, N., 2005. Constraint solving in uncertain and dynamic environments: a survey. Constraints, 10, pp253–281. https://doi.org/10.1007/s10601-005-2239-9.

Vincent, P., Rubin, I., 2004. A framework and analysis for cooperative search using UAV swarms. 2004 ACM Symposium on Applied Computing, SAC ’04. ACM, New York, USA, pp. 79–86.

Wagberg, J., Zachariah, D., Schon, T., Stoica, P., 2017. Prediction performance after learning in Gaussian process regression. Artificial Intelligence and Statistics (PMLR), pp1264–1272.

Waharte, S., Trigoni, N., Julier, S., 2009. Coordinated search with a swarm of UAVs. 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, Rome, Italy, pp1–3. https://doi.org/10.1109/SAHCNW.2009.5172925

Walshe, N., Ryng, S., Drennan, J., O’Connor, P., O’Brien, S., Crowley, C., Hegarty, J., 2021. Situation awareness and the mitigation of risk associated with patient deterioration: A meta-narrative review of theories and models and their relevance to nursing practice. International Journal of Nursing Studies.

Wang, J., Xu, Z., 2014. Bayesian Inferential Reasoning Model for Crime Investigation. Frontiers in Artificial Intelligence and Applications 262, pp59-67.

Wang, Q., Bu, S., He, Z., Dong, Z.Y., 2021. Toward the prediction level of situation awareness for electric power systems using CNN-LSTM network. IEEE Transactions on Industrial Informatics, 17, pp6951–6961. https://doi.org/10.1109/TII.2020.3047607

Wang, S., Li, H., Niu, S., 2021. Empirical research on climate warming risks for forest fires: a case study of grade I forest fire danger zone, Sichuan province, China. Sustainability *13*(14), 7773; https://doi.org/10.3390/su13147773

Wei, H., Chen, X.-B., 2020. Flocking for multiple subgroups of multi-agents with different social distancing. IEEE Access, Multidisciplanary, pp164705–164716.

Weick, K.E., 1995. Sensemaking in organizations. University of Michigan, Michigan, US, Volume 3.

Weick, K.E., 1993. The collapse of sensemaking in organizations: the Mann Gulch disaster. Administrative Science Quarterly 38, pp628–652. https://doi.org/10.2307/2393339

Williamson, J., 2001. Bayesian networks for logical reasoning. AAAI Technical Report Compilation.

Wiltshire, T.J., Butner, J.E., Fiore, S.M., 2018. Problem-solving phase transitions during team collaboration. Cognitive Science 42, pp129–167. https://doi.org/10.1111/cogs.12482

Wittenburg, L., Zhang, W., 2003. Distributed breakout algorithm for Distributed Constraint Optimisation Problems. 2nd International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS ’03. ACM, New York, USA, pp1158–1159.

Xuan, G., Zhang, W., Chai, P., 2001. EM algorithms of Gaussian mixture model and hidden Markov model. 2001 International Conference on Image Processing, pp. 145–148, Thessaloniki, Greece. https://doi.org/10.1109/ICIP.2001.958974

Yan, Z., Jouandeau, N., Chérif, A.A., 2011. Multi-robot decentralized exploration using a trade-based approach. 8th International Conference on Informatics, Automation, and Robotics. Noordwijkerhout, Netherlands.

Yang, X., Song, C., Xu, C., Hao, M., 2022. A survey of the estimation and fusion methods for battlefield situation awareness. 7th Asia Pacific Conference on Optics Manufacture (APCOM 2021), Singapore, pp. 785–793.

Yang, X., Suash Deb, 2009. Cuckoo search via Lévy flights. 2009 World Congress on Nature Biologically Inspired Computing (NaBIC), Coimbatore, India, pp 210–214.

Yang, X.-S., 2012. Bat Algorithm for Multi-objective Optimisation. International Journal of Bioinspired Computing.

Yang, X.-S., 2010. Firefly algorithm, Lévy flights and global optimisation. Research and Development in Intelligent Systems XXVI. Springer London, pp209–218.

Yang, X.-S., Karamanoglu, M., Ting, T.O., Zhao, Y.-X., 2014. Applications and analysis of bio-inspired eagle strategy for engineering optimisation. Neural Computing and Application 25, pp411–420.

Yang, Y., Polycarpou, M.M., Minai, A.A., 2007. Multi-UAV cooperative search using an opportunistic learning method. Control 129, pp716–728. https://doi.org/10.1115/1.2764515

Yanmaz, E., Kuschnig, R., Quaritsch, M., Bettstetter, C., Rinner, B., 2011. On path planning strategies for networked unmanned aerial vehicles, pp212–216. https://doi.org/10.1109/INFCOMW.2011.5928811

Yeoh, W., Varakantham, P., Sun, X., Koenig, S., 2011. Incremental DCOP search algorithms for solving dynamic DCOPs. 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3, AAMAS ’11. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 1069–1070.

Yoon, Y., Kim, Y.-H., 2013. An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks. IEEE Transactions on Cybernetics 43, pp1473–1483. https://doi.org/10.1109/TCYB.2013.2250955

Yu, W.-C., Yang, C.-Y., Su, K.-H., Tu, Y.-H., 2014. Dynamic path planning under randomly distributed obstacle environment. 2014 CACS International Automatic Control Conference (CACS 2014), pp138–143. https://doi.org/10.1109/CACS.2014.7097177

Zadeh, R.B., Zaslavsky, A., Loke, S.W., MahmoudZadeh, S., 2021. Multi-UAVs for bushfire Situational Awareness: a comparison of environment traversal algorithms. 2021 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing Communications (GreenCom) and IEEE Cyber, Physical Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), Melbourne, Australia, pp. 107–114.

Zhang, L., Rodrigues, L.O., Narain, N.R., Akmaev, V.R., 2020. bAIcis: a novel bayesian network structural learning algorithm and its comprehensive performance evaluation against open-source software. Journal Computational Biology, 27, pp698–708. https://doi.org/10.1089/cmb.2019.0210

Zhang, S., Wang, F., Tan, S., Wang, S., Chang, Y., 2015. Novel monitoring strategy combining the advantages of the multiple modeling strategy and gaussian mixture model for multimode processes. Industrial Engineering Chemistry Research, p11866–11880.

Zhang, T., Kaber, D., Zahabi, M., 2021. Using situation awareness measures to characterize mental models in an inductive reasoning task. Theoretical Issues in Ergonomics Science, pp1–24.

Zhang, W., Wang, G., Xing, Z., Wittenburg, L., 2005. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimisation problems in sensor networks. Artificial Intelligence, pp55–87.

Zhao, Y., Wang, T., Li, Z., Li, P., Hu, W., Cheng, X., 2021. Overview of research on distribution network fault prediction based on Situation Awareness. IEEE 5th Conference on Energy Internet and Energy System Integration (EI2), pp1436–1440.

Zhou, X., Wang, W., Tao, W., Xiaboo, L., Tian, J., 2018. Continuous patrolling in uncertain environment with the UAV swarm. PLOS One.

Zivkovic, Z., 2004. Improved adaptive Gaussian mixture model for background subtraction. 17th International Conference on Pattern Recognition 2004, Cambridge, UK, pp. 28-31. https://doi.org/10.1109/ICPR.2004.1333992

Appendices

Appendix A

Delaunay-centric Algorithm

Input: seed waypoints (W)

Output: Waypoints plan (π)

For all $a_i \in A$ do // i.e. for all simple agents

assigned to a picture compiler

Find $\pi_1 \in \Pi$ find

$\pi_i \in \text{argmin/max}_{\pi \in \Pi^i} U^{\pi}$ (U is the optimised

set of waypoints policies as defined by

equation 3) using

While count(τi) ≤ 2

τi: $a_i \rightarrow w_i$ Select the agents' first-layer

waypoints as seed points, e.g., using longest

non-crossed jumps of Figure 9 etc.

8:	Triangulate the seed waypoints using Delaunay triangulations.
9:	Find the centre of each triangle at every layer and mark it as the seed for the upper layer.
10	$w_1 \rightarrow \tau_1$ // Add waypoint to layer
11	Repeat the process //go to 6
12	$\tau_1 \rightarrow \pi_i$ // Add the layers to the plan.
13	Return π_i

Supplemental Files

Step-by-step instructions and sample code to apply the proposed algorithm in any team of agents search problem, e.g., the described forest fire searching, missing person finding, etc., can be found in https: //www.dropbox.com/s/1ebdr20b5zwzr6n/Supplimental%20Documents.zip?dl=0.