Intelligent and Autonomous Multi-UAVs (Multiple Drones) Swarm Monitoring for Effective Surveillance and Situation Awareness in the Nigerian Telecommunication Industry

Final Report

By

Sagir Yusuf Muhammad(Team Lead)

Ibrahim Abdullahi

Adamu Bappi

Summary of the Project Outcomes

This project took a direction of formalising the concept of multi-UAVs (multi-drones) swarm coordination to conduct surveillance mission especially for the purpose of monitoring telecommunication industry properties, services monitoring, and Situation Awareness management. The primary focus points are: (i) development of algorithms and methods to effectively coordinate the swarm activities; and (ii) development of models to effectively manage Situation Awareness (SA) of the UAVs (Unmanned Aerial Vehicle) and overall telecommunication industry data. The first challenge was addressed by developing a model that create effective (resources utilised) surveillance route plan for the UAVs swarm. This is modelled as a Distributed Constraint Optimisation Problem (DCOP) and the results proved agents resources optimisation. The second challenge is addressed by applying Bayesian Belief Network (BBN) to modelled the system Situation Awareness. Algorithms and methods were developed to describe how UAVs data can be converted to system information, how to merge experts contributions, and how to make predictions and handle uncertainties in telecommunication industries data using expectation-maximisation algorithm, gradient descent algorithm, and counting algorithm. In all cases, practical application to telecommunication industry problems (e.g., surveillance, quality of service prediction, missing data estimation, radio planning, signal monitoring, signal booster detection, etc.) were applied as the use case and the results proved better performance over the existing strategies. Summarily, the following achievements were recorded:

i. Development of novel, intelligent, and autonomous algorithms and methods to coordinate swarm of UAVs for surveillance and Quality of Service (QoS) monitoring in telecom industries
ii. Application of Bayesian Belief Network (BBN) to model the concept of Situation Awareness for telecommunication industry data acquired through the swarm of UAVs operation
iii. Development of the Nigeria's first intelligent and autonomous multi-UAVs swarm surveillance system. This has been implemented at the Gombe State University security services under the sponsorship of this grant. This involves a series of training and a donation of twelve (12) intelligent surveillance UAVs (drones) and mobile tablet to the

University. Note that, this system can be applied in the telecommunication industry Quality of Service monitoring mission (e.g., the signal booster detection system of iv below, as already tested) or used by the Nigerian security forces to improve our national security (i.e., by assisting with a cheap and intelligent multi-UAV surveillance services e.g., to monitor train tracks, sensitive areas (e.g., Kaduna- Abuja high way, etc).

iv. Signal boosting devices detecting system: in order to assist NCC with their telecom regulation mandates, we developed a system that utilises drones in detecting signal boosters installation.

v. Journal articles publication in the internationally recognised, reputable, and high-index journals. The first journal was the IEEE transactions on Human Machine Systems (HMS) with the article entitled "Formalising Distributed Situations Awareness in Multi-agent networks". The Second article was published in the Frontiers for AI and Robotics with the title " DIMASS: A Delaunay-inspired, Hybrid Approach to a Team of Agents Search Strategy"

vi. Publication of two conference papers. The first paper entitled "Towards Autonomous Multi-UAVs Surveillance Mission: A Study of Nigerian Telecommunication Masts Surveillance" was presented at the 1st International Conference on Multi-disciplinary Engineering and Applied Sciences, Abuja. The second conference paper was presented at the International Conference on Adaptive Techniques, Analysis, and Algorithms, London, United Kingdom with the title "Autonomous Heterogeneous Multi-agent Coordination for Effective Surveillance Mission in Dynamic Environment".

vii. Best presentation award at the International Conference on Adaptive Techniques, Analysis, and Algorithms, London, United Kingdom.

viii. Training of over 1000 Nigerian on multi-UAVs (multi-drones) opportunities. This includes students (i.e., by creating awareness on recent research areas within the field), security personnel (i.e., training on surveillance), surveyors from the Nigerian Association of Surveyors and general public (i.e., awareness on jobs available within the field of multi-UAVs e.g., photography, filming, etc).

ix. A total of six (6) workshops, three national webinars, and seven (7) practical Multi-UAVs training sessions were conducted.

x. Creating awareness among students and lecturers on research areas within the field of multi-UAVs. This lead to the production of many undergraduates, masters, and PhD projects within the field of multi-UAV mission for telecommunication industry. Some of the students receive partial funding from this grant.

xi. Establishment of the multi-UAV section of the artificial intelligence laboratory of Gombe State University which receives some funding from this project.

xii. Two conference papers drafts to be submitted in both national and international conferences. However, this is subject to financial availability because our budget was not executed as planned due to the rise in the price of dollar.

xiii. Provision of Multi-UAV hardware. A total of thirty-six (36) drones, three (3) tables, and one (1) mobile phones, and sensors were purchased and made available for the host institution research

xiv. Local drones maintenance e.g., battery boosting, battery construction, propellers constructions, arduino programming, etc.

xv. Locally made drones spare parts and tools.

xvi. Anti-drone spy system

In terms of challenges, we faced some financial issues due to the rise in the price of dollar which generates large variation between the planned budget and the current markets price. We overcome the challenge by utilising local materials were necessary and reduction of items quantities.

This report is segmented into two pillars. The first pillar deals with the issue of multi-UAV swarm coordination for effective surveillance. The second pillar addresses the issue of Situation Awareness maintenance in telecommunication industry Quality of Services (QoS) assurance. This is in line with addressing the commission’s comments on our previous reports.

Chapter 1: Multi-UAVs (Multi-Drones) Swarm Coordination for Effective Surveillance and Quality of Service and Situation Awareness Management in Telecom Industry

Introduction

At its simplest level, coordination of multiple UAVs can be characterized as flocking [1]–[3]. In this, individual UAV SA is a matter of detecting one's neighbors and adjusting activity accordingly. Extending this to DSA, we might assume that the flock is coordinating its activity towards a common goal, say, to catch prey. Here, there needs to be some allocation of function between entities. That is, members of the flock need to perform different actions in different locations, e.g., herding the prey, ensuring that the prey is kept with a given region, attacking the prey etc., and these actions rely on different information to support them. In this case, coordination involves the 'flock' understanding the environment and interpreting this in terms of a common goal. Rather than local heuristics, flock-level understanding can be reflected in an ontology by which the flock (as a holistic system) appreciates a common goal. Thus, the flock (rather than a single entity) maintains the swarm coordination. For example, Figure 1 describes a multi-UAV surveillance problem for telecommunication industry.

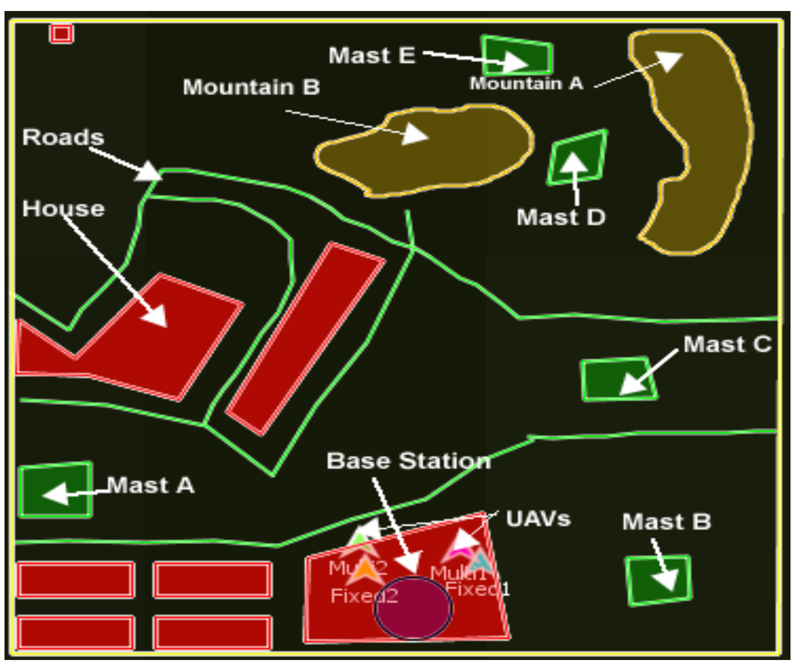


Figure 1: Simulation of Multi-UAVs Surveillance Problem for Telecommunication Industry

The task of the UAVs in Figure 1 is to ensure constant surveillance of the telecommunication antennas spread across the selected village. The UAVs mission also needs to utilise the UAVs resources (batteries, communication bandwidth, computational power etc.). We termed the problem as Multi-Agent Planning under Destination Uncertainty and Limited Resources (MAP/DULAR). The problem (MAP/DULAR) is the variation of multi-agent pathfinding under destination uncertainty (MAFP/DU) of [4] and multi-agent planning under sparse interaction of [5]. We believe that our constraints are tough, and the expected outcome needs to be simple which ease realistic application implementation on real devices. We are now to define the basic components for MAP/DULAR problem.

Definition 1. Agents (A): is an autonomous entity capable of executing its own plans, e.g., navigation, computations, and so on. An example of an agent is the unmanned aerial vehicles (UAVs) in figure 1. The agents can be imposed on different constraints which affect the complexity and difficulty of the MAP/DULAR problem. For example, simple agents may not be communicating with one another during mission execution. This version of multi-agent planning problem is very difficult to solve than its centralised or decentralised versions with agents communication. Therefore, we assume the following constraints (i) no communication among simple agents during mission execution (iii) agents have very limited resources (such as energy, memory, and computational capacity).

Definition 2. Agent parameters (U) are the set of attribute variables for the agent, e.g., energy, speed, computational capacity, etc, which define the utility cost of its action. These parameters will be used in evaluating the cost effectiveness of the coordination planning algorithms.

Definition 3. Environment (E) is the space to be explored by the agents—for example, the rectangular space in figure 1. The environment E can be modelled as the set of waypoints V connected with edges E. Therefore, the environment is the graph E = (V,E). We assume that our agents are aerial robots operating at high altitude as such our environment is obstacle-free. We also assume that our environment has changing variables which affect the targets' mobility. For example, wind direction will be affecting fire (targets O1 and O2) movement in figure 1. The factors affecting targets' movement (e.g., wind speed, wind direction, etc.) were assumed to be predictable with high degree of accuracy.

Definition 4. Target (T) is any static or moving object of interest within the environment (E). Example of targets are the fires (O1 and O2) in figure 1.

Definition 5. Waypoint (V): is a particular geodetic point within the searching environment i.e., the set of waypoints $v_i \in V$ within the environment E. The Euclidean distance between waypoints in MAP/DULAR plan is $e = ||v_i - v_{i+1}||$ for all $e_i \in E$. The agents waypoints generation function λ is the injective function λ:A→V i.e., one-to-one waypoints assignment function.

Definition 6. Coverage (σ) of MAP/DULAR: is the agents' sensed area on all axes, i.e.,

$$\sigma = \sum_{d=1}^{d=n+1} \sum_{i=1}^{i=n} \left(\int_{r_v^i}^{r_v^{i+1}} \left(\left\| e_i^{d=1} - e_{i+n}^{d=2} \right\| \right) + \cdots + \int_{r_v^j}^{r_v^{j+1}} \left(\left\| e_i^{d=n} - e_{i+n}^{d=n+1} \right\| \right) \right) \quad (1)$$

Where e_i is the edge i for the sensed area, d is the environment dimension, r_v^i and r_v^{i+1} are the agent's lower and upper sensing range. Therefore σ:A→V^d is a function for mapping agents' waypoints with their covered areas dimension. Another important variable related to coverage is the diversity of MAP/DULAR, which is the related way of how the covered area spread across the environment. For example, covering 20KM^2 within the middle of the forest could have higher diversity than covering at one side of the forest. The reason behind that is, if we can estimate the expected target's size and mobility rate, we can easily conclude its presence or absence within the the environment.

Figure 2 described an example of diversity variation for two MAP/DULAR plans with the same coverage and length of edges. The plan in picture 'B' is more diverse than the one in picture 'A'.

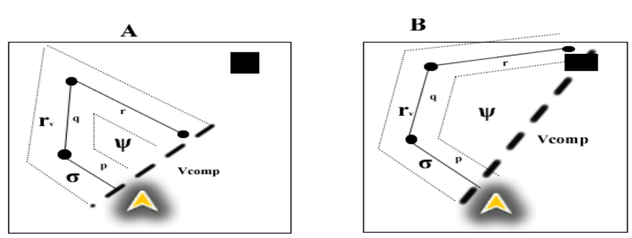


Figure 2: Multi-agent Planning Diversity Variation

Definition 7. Multi-agent planning under destination uncertainty and limited resources (MAP/DULAR): is the resources optimised (best utility) set of waypoints at individual and global agents planning problem.

Definition 8. The utility of MAP/DULAR is the set of cost variables to be optimised while solving the planning problem. It can be the combination of agents parameters and global mission objectives to be optimised, for example, the multi-objective variables: energy, coverage, redundant search, mission time, etc. The best utility function U_{best} can be described as the function.

$$U_{best}(\lambda) = \mathrm{argmin/max}_{C}[\sum_{t=0}^{T} \sum_{ci}(\vec{U}_{\mathrm{i}}(\mathrm{A}_{\mathrm{i}}^{\mathrm{t}} \backslash \lambda))]^{T} \quad (2)$$

Where V, λ is the set of waypoints and their assignment function (from definition 5). $\vec{U}_{\mathrm{i}}$ is the vector containing the best individual agents utility cost. In other words, U_{best} is the function that gives the best utility at every situation.

Definition 9: Redundant search: is the continuous exploration of a space within a short period of time, and it can be categorised into intra-agent (redundancy within agent's self waypoints) and inter-agent redundancy (redundancy among other agents' waypoints). **Intra-agent redundancy** in MAP/DULAR is the variable $\pi = \sum_{i}^{n}(\sigma_{i}^{t^{i}} \cap \sigma_{i+n}^{t^{i+n}})$ where t_{i+1}-t_i < $t_{threshold.}$. i.e., the summation of all multiple covered areas within a minimum time and space threshold. **Inter-agent redundancy** in MAP/DULAR of two agents i and j is $\Pi = \sum_{i}^{n}(\sigma_{i}^{t^{i}} \cap \sigma_{j}^{t^{j}})$ where t_{i+1}-t_i <$t_{threshold}$ $i \neq j$. i.e., the summation of all redundant coverage among agents. Therefore, the entire MAP/DULAR redundancy is the summation of intra and inter agents redundancy define by $M_{red} = \sum_{i}^{n}\left|\left|\pi_{i} - \pi_{i+n}\right|\right| + \sum_{j}^{n}\left|\left|\Pi_{j} - \Pi_{j+1}\right|\right|$

Therefore, achieving efficient MAP/DULAR given the constraints that agent are not to be communicating and negotiating while conducting their mission is all about the creation of effective policies to be controlling agents' behaviours during their mission execution. At this junction, let us state this important point, our work is proposing the use of smart, simple, and intelligent policies to control agents' individual activities during path planning generation. That is, rather than looking for strategies to control agents reasoning and communication during mission execution, which at the end make the agents to be performing complex activities (e.g., sensor data acquisition, logics, information processing, delay controls, security, etc.) why not make the policies (rules) controlling the agents' activities very intelligent and utilise the simplest (poorest) agent to accomplish the task. By doing so, we can save a lot of agents resources (e.g., energy, computational capacity, memory, etc.), and achieve a secured, robust, and scalable distributed multi-agent planning solution.

Definition 10. K-previous assessment is the concept of analysing agent's self k-previous, where k = 1,2,3,…,n, activities (e.g., selected waypoints) by itself or with other agents. For example, an agent may be saving its k-previous waypoints and check for redundant search. This aspect can be used in obtaining reduced intra-agent and inter-agent redundancy while solving MAP/DULAR.

Definition 11. MAP/DULAR Policy (β): is the set of rules that controls the overall agents' behaviours and optimises the mission utility costs describe by the tuple $\beta = (A, R_t, U)$, where A is the set of agents, R_t is the set of rules at a given time (more especially when dealing with dynamic systems).

Definition 12. MAP/DULAR Layer (τ): Refers to the set of waypoints at the same level. $\tau: A \rightarrow v$ for all $\tau, v \in V$.

Obviously, making the policies simple will be an added advantage to the agents' resources consumption. The essence of definitions 1-10 is to be able to have a thoritical basis on how the policies could be created. Next section describes the model.

The Model

We modelled the multi-agent planning problem as a multi-objective distributed constraints optimisation problem (MO-DCOP) described by the tuple T = $\{A_{i,j}, S, T, E, \alpha, k, C, I\}$ where A_{ij} is the set of agents i of type j, mainly the low and middle level agents.

S is the set of agents states over time period T, i.e., $S_i = \{S_1 \times S_2 \times S_3 \times \ldots \times S_i\}$

T is the defined mission time space $T = \{T_1, T_2, T_3, \ldots, T_n\}$ n = 1,2,3,…,n. e.g., T_1=5 minutes, T_2 =10minutes,..etc.

E is the set of environmental conditions such that $E \triangleq \{S, R, T\}$ where R is the environment's dynamic features. That is, the environment condition is defined by current environmental dynamic features e.g., fire, wind direction, wind speed, etc., agent situation, and current operating time.

α is the set of actions space across agents, such that $\alpha_i = \{\alpha_1 \times \alpha_2 \times \alpha_3 \times \ldots \times \alpha_i\}$ factored across agents.

k is the set of constraints imposed on the agents, e.g., as described above.

V is the set of policies for choosing variables v e.g., waypoint to be visited for the agents which define the cost for the agent's situations and actions i.e., V_i: $A_i \times S_i \times \alpha_i \rightarrow c$, where $c_i \in C$ that is, the optimised individual cost (c_i) contribute in the mission cost optimisation (C).

I is the agent's interactions define by the tuple $I = (C, A_{i,j}, \alpha_i)$

C is the set of utility cost functions, such that utility cost for single agent i is define by c_i: $A_i \times V \times S \times \alpha_i \rightarrow \mathbb{R}$, i.e., set of agent's actions, at a given state, mapped to a real number cost value. The essence of the real number value is to be able to grade the mission cost. For example, if an agent 'A' finishes exploring point 'X' and now has the option of going to point 'Y' or 'Z', the choice of 'Y' or 'Z' depends on the current situation, e.g., 'Z' may be a redundant waypoint with an overall energy cost of -10 and why in not a redundant waypoint with a cost of +15, then surely, 'Y' point will be visited.

Therefore, the central problem is to obtain an optimised group of policy π' such that M^{π} is minimal or maximal base on the designated agent parameters, i.e., $\pi' \in \mathrm{argmin/max}_{\pi \in \Pi^i} M^{\pi}$ where $\pi = (\pi_1, \pi_2, \pi_3, \ldots, \pi_i)$ is the set of optimised policies for each agent state-action transitions, M is the global mission cost function in all situations over time define by $M = \sum_{i,j=1}^{i,j=n} \sum_{T=0}^{T=t} \vec{C}(S_{t,i}, A_{t,i,j}, V_{t,i}, \alpha_{t,i})$. That is, to develop a policy creation algorithm that generates policies for coordinating the agents' activities during the planning task under the imposed constraint with minimal resources. Each of the agents will have a copy of the policy creation algorithm, and thus, at the pre-mission planning, the simple UAVs negotiate with each other in order to share starting points, which after mission has started, communication will not be possible among them and as such, they will be using the policy-based algorithm to optimise their activities, e.g., make a decision on places to visit. Next section introduces the multi-agent planning algorithms that solves MAP/DULAR problem through the use of simple policies.

Existing Solutions

A comparative analysis of related works and methodologies was done in this project. The analysis reviewed and identified gaps to be filled in the domain of surveillance mission. Different strategies were suggested for surveillance mission operations since the early 19th century, which started from the use of wardens on horses, tree climbing, building towers, satellite imaging systems, helicopter patrol, camera surveillance, and UAVs missions[6]–[11]. In our approach, we modelled the multi-UAV surveillance (searching problem) as a Distributed Constraint Optimization Problem (DCOP). We use a team of heterogeneous Unmanned Aerial Vehicles to conduct area exploration by optimising the searching time and energy as the cost functions. The aim of the UAVs is to detect telecommunication signal strength across the search space under imposed constraints such as minimal communication, mountains blocks, and so on. The task is to detect the signal and report the value to the base station for Quality of Service (QoS) control purposes. UAVs were assumed to be mounted with various signal detecting sensors (i.e., presence of heterogeneous UAVs).

In summary, the following key strategies were compared, which eventually led to the fine-tuning of the developed model for the requirements of this project from [10], [12]–[17].

i. Levy Flight: The continuous and random exploration of a single location within minimum time and space. Both naturally inspired and artificial methods presented limitations of difficulty in controlling agents (swarm) activities due to randomness in generating waypoints[13], [15]. This is mostly inspired by animals' mode of foraging. Here agents, generate random waypoints within the searching space.
ii. Geometric-based search patterns: Different geometric-based shape search patterns exist in surveillance missions such as parallel track, creep lining, Zamboni search, grid-base, sector search, expanding square search etc. [9]–[17]. These approaches provide a fixed pattern for conducting space exploration and perform better in a well-known and static environment. The main challenge of the geometric-based approaches is inflexibility, i.e., difficulty to control for various purposes.

These strategies were reviewed and adopted in the simulation done in the works to identify which method best fits the telecommunications surveillance problem in terms of complex terrains as well as network distribution. A novel model based on the Delaunay triangulation of the search space was developed as described by Algorithm 1. The approach generated systematic waypoints (search area) then triangulated them and visit each centre of the triangle continuously.

Layers of waypoints are generated based on the number of Delaunay triangles (derived from the Delaunay triangulation policies).

Algorithm evolution

Some experiments were carried out to validate the model developed, having analysed the potential strategies. Papers 1 & 2 (as attached with this report) described a problem space and performed modelling to implement and compare strategies, then adopted an improved solution based on results obtained. From paper 1, it was identified that Levy flight had the most prospects to solve and apply the surveillance strategy for a telecommunications industry domain, and so paper 2 adopted that and performed well under the constraints and scenarios used.

We modelled a forest with mountains, telecommunication mast, and buildings inside and tasked four UAVs of two different types (multirotor and fixed-wing) to detect the telecommunication signal strength as described in Figure 1 using Aerospace Multi-agent Simulation Environment (AMASE). We implement both parallel rack and Lèvy flight and set up the costs functions as the searching time and energy used.

Figure 2 describes a simulated search space. It contains buildings, roads, masts, and the base station. The task of the UAVs is to search for the fire using the least time and energy. We applied both parallel track searching and Lèvy flight, as shown in Figures 3 and 4. The experiments were run 30 times, and the average of the values was used.

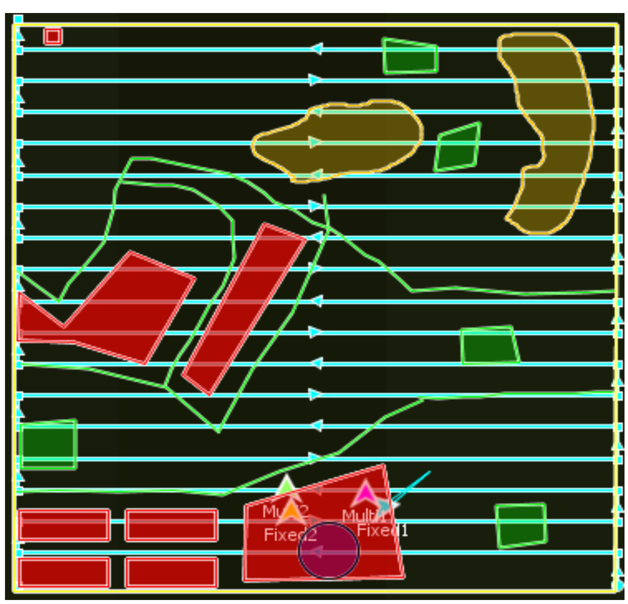


Figure 3: Parallel Track Search.

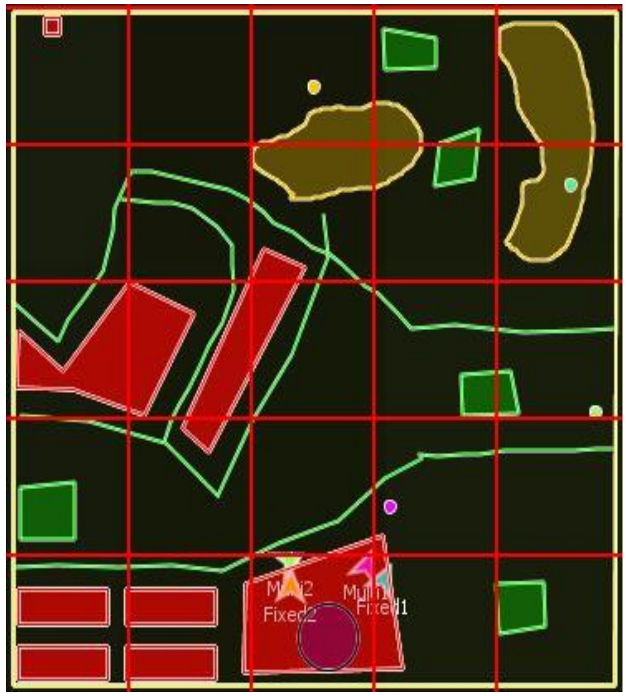
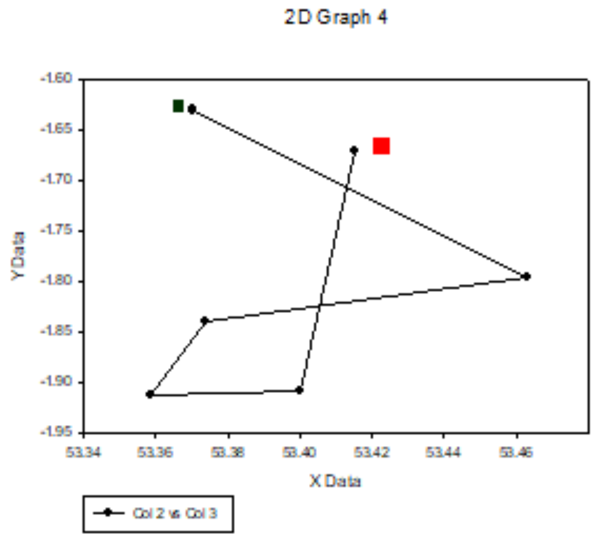


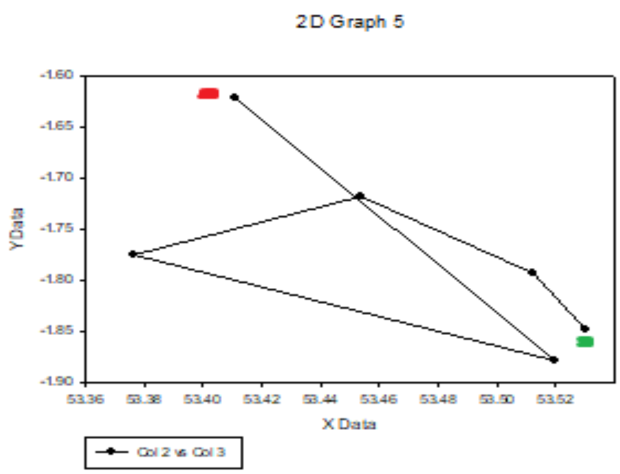
Figure 4: Lèvy Flight Searching.

Figure 3 describes the parallel track sweeps by a single UAV to demonstrate the structure of the parallel track search strategy. Thus, it is challenging to get the size of the sweep in parallel track searching. The performance of the search is affected by the position of the targets. Therefore it is not robust and scalable. On the other hand, Lèvy flight follows a Lèvy distribution and generates well-diverse waypoints using equations 1-3. As described in Figure 4, agents generated a waypoint (the same as the agent's colour) to search for fire; if not found, they regenerate another one continuously until a target is found or energy is about to finish. Figures 5a and 5b describe the sample trajectories of two different UAVs performing Lèvy search (green and red dots represent starting and ending waypoints). The performance of searching approaches was presented in Figures 6 and 7. The search strategies were evaluated based on the space they covered using the assigned search task. Hence, the evaluation is based on how waypoints were distributed across the search space.

A clear challenge to the levy flight strategies is the concern on how to control the UAVs operations. That is, how to control who does what among the agents due to the randomness in waypoints generation. We attempt to change the random number of seeds ranges. Surprisingly, this has no effect on the control aspect of the levy flight, as described in Table 1.



a. Example of UAV 1 Levy Flight Plan



b. Example of UAV 2 Levy Flight Plan

Figure5: Fixed-wing 1 Conducting Searching Task using Lèvy Flight.

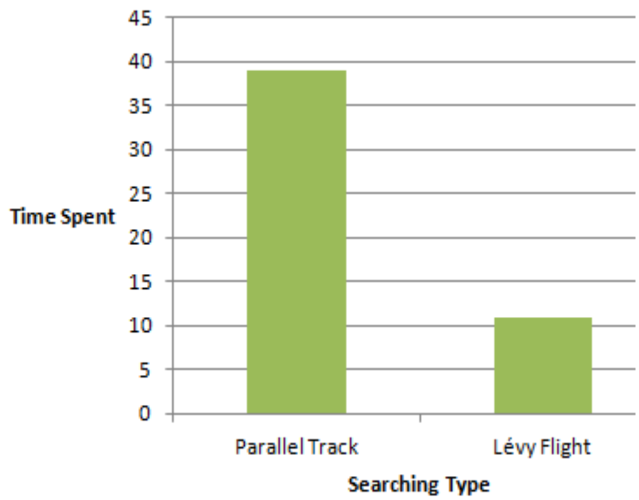


Figure 6: Searching Time Performance Comparison.

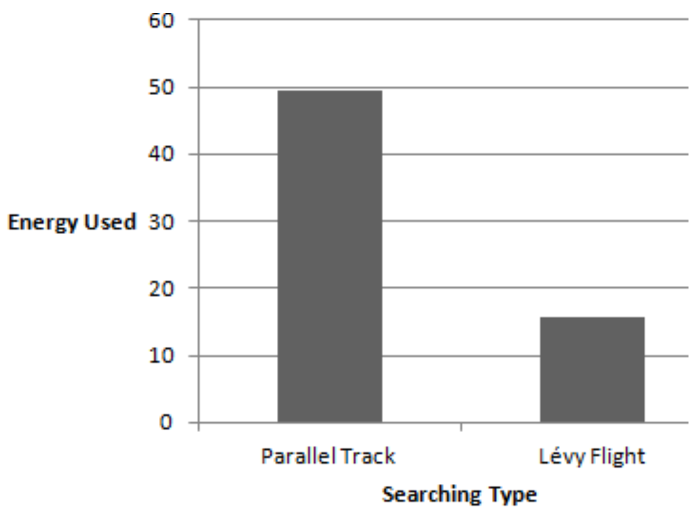


Figure 7: Energy Used Performance Comparison.

Table 1:Metrics Used

#	Area Covered by the UAV Mission(Km^2)	Levy Distribution Random Number Range
1	273.58465	1-5
2	288.39134	1-10
3	260.3449	1-15

4	275.85979	1-20
5	193.87027	1-25
6	293.285	1-30

Figures 6 and 7 describe the performance of the parallel track and Lèvy flight searching approaches. Lèvy flight saves more time and energy. It also provides a robust and scalable multi-agent searching technique. That is, it is independent of the structure.

In summary, to map these results with the milestones of this project, the following points highlight what has been achieved. Concept of Operations: A concept of operation for the Autonomous UAV was designed and validated using AMASE to specify the potential behaviour of the drones when deployed on the field. As such, resources usage and constraints were factored Such as energy, time, computation, and network. Secondly, Environment dynamics were factored to facilitate the surveillance in complex terrains such as mountains and valleys in remote areas. Algorithm 1 described the proposed algorithm, which is more effective, robust, and scalable and proved by the submitted journal. Swarm Drones Algorithm has been discussed in the previous model.

Telecommunication Industry Application (Signal Surveillance): In terms of the designated multi-agent surveillance for signal strength detection, the levy flight demonstrates a more effective solution based on the results in Figures 6 and 7. Additionally, it demonstrates a certain level of flexibility (waypoints spread across the search space). The flexibility and randomness could suit the dynamic behaviour of the network signal fluctuations due to external environmental factors, e.g., wind, mountains blockage etc. In terms of coordination control, the levy flight is difficult to control due to the random described by the average of 30 experiments in Table 1. The random number seeds variation demonstrates no effect on the searching task control. This aspect has been defined in detail from our submitted report 2. Future work focuses on real UAVs implementation of algorithm 1.

Our Solution: Multi-UAVs Planning Problem under Destination Uncertainty and Limited Resources: A Policy-based Approach

The policy-based algorithm evolved by applying the Delaunay triangulation strategy. The Delaunay triangulation of a set of waypoints occurs by inscribing triangles of the waypoints in a circle, such that no waypoints in any of the circumcircle of any triangles [26]. The algorithm starts by generating waypoints for the first layer using designated strategy e.g., making longest non-crossed jumps, zigzag jumps, etc., then perform the Delaunay triangulations of these seeds waypoints, visit the centre of each triangle, then perform the Delaunay triangulation of the centre of the first layer triangles as the second layer, repeat the process until the number of centre waypoints is less than or equal to 2. Algorithm 1 describes the first version of the Delaunay triangulation-base algorithm, and figure 3 describes the path generated by one of the agents in Figure 1.

Algorithm 1. The Delaunay Triangulation Based MAP/DULAR Algorithm

```
Input: seed waypoints
Output: MAP/DULAR waypoints
For all ai ∈ A do
 For all π ∈ Π finds
     π_i ∈ argmin/max_{π ∈Π^i} M^π   finds using
τ_1: a_i →v_i Select the first layer waypoints as seed points using appropriate team negotiation e.g., longest non-crossed jumps of figure 3.
Triangulate the seed waypoints using Delaunay triangulations
Find the centre of each triangle at every layer and mark it as the seed for upper layer
```

Repeat until number of waypoint is less than or equal to 2

Return π_i

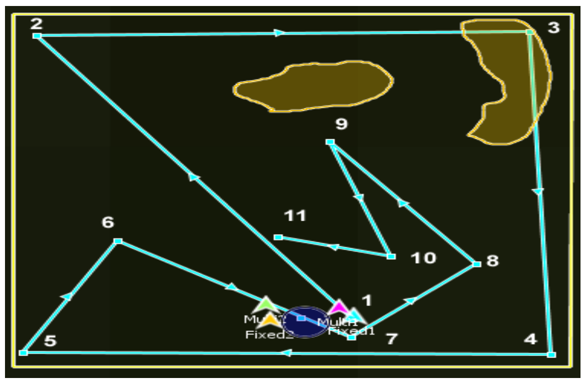


Figure 3. The Delaunay Triangulation Based Planning for One Agent

Figure 3 describes a single agent's planning solution. The waypoints labelled 1,2,3, and 4 are the longest non-crossed seed waypoints for the first layer. The second layer waypoints are the waypoints 6,7,8, and 9, which are the centres of the triangles of first layer's triangulation. Layer 3 waypoints are 10 and 11. Based on our definition 6, the MAP/DULAR planning algorithm in Algorithm 1 gives a well diverse searching waypoint which exhibits the best features of flexible and fixed-pattern approaches. That is, it looks predictable because agents' future waypoints can be predicted if the initial point and the policies used for subsequent waypoint generation are known. In addition to that, it looks pseudo-random (by given high diverse pattern) as such it can be controlled and be predictable which eases agents knowledge understanding, analysis, and prediction (i.e., Distributed Situation Awareness [27]). In terms of complexity, algorithm 1 is of $O(n^2 logn)$ complexity, i.e., $O(nlogn)$ for performing the Delaunay triangulation [26] and the $O(n)$

for finding the centre of each triangle. For multiple agents, the seeds waypoints can be varied and unique searching waypoints would be obtained because the Delaunay triangulations is unique [26], [28]–[31]. Although the path planning approach can solve MAP/DULAR with less resource and been predictable, we still propose an idea of simplifying and making it more robust using algorithm 2.

Algorithm 2 works by inspiring algorithm 1, when an agent obtains the seeds waypoints 1 to 5 in figure 3, instead of triangulating them, it will be using distinct policies (rules) to generate waypoints in the second layer. For example, the waypoints 6, 7, 8, and 9 in figure 4 were obtained by projecting in angle $\Theta = 180^{o}/n$ where n is the number of upper layer's waypoints. The quadrant projection uses different sequences depending on the number of agents and unique paths needed, for instance, first, third , second, and fourth quadrants, another agent may have third, fourth, second, and first quadrant and so on. Therefore, we now have different policies for generating projection angles and directions in various layers. The number of waypoint per layer utilises the Delaunay triangulations number of triangles theorems (i.e., number of triangles and edges of the Delaunay triangulation are 2n-2-k and 3n-3-k respectively, where n is the total number of waypoints and k is the number of convex waypoints). Therefore the policies for generating seeds waypoint and each layers waypoints at the PC's or host level control the agents' overall planning. The selection of the first layer waypoints will be given by the PC after checking all agents joins with other agents during the pre-mission planning. Algorithm 2 describes the pseudocode for the proposed MAP/DULAR algorithm that inspired algorithm 1, and figure 4 describes an example of the path generated using algorithm 2 for a single agent.

Algorithm 2: Generating MAP/DULAR using Delaunay-inspired Algorithm

Input: seed waypoint
Output: MAP/DULAR waypoints
For all ai $\in$ A do For all $\pi \in \Pi$ finds $\pi_i \in \text{argmin/max}_{\pi \in \Pi^i} M^{\pi}$ finds using

Select the first layer waypoints as seed points using appropriate team negotiation e.g., longest non-crossed jumps of figure 3.

Select the first layer waypoints as seed points using appropriate team negotiation e.g., longest non-crossed jumps of figure 3.

Get the seeds and use projection angle and edge policies (rules) to generate each layer's waypoints

Use the Delaunay triangulation rules to generate waypoints for each layer and repeat the process until number of waypoints is less than or equals to 2.

For all $v_i \in V$

Allocate $\lambda: A_i \rightarrow v_i, \forall v_i \in V$ s.t. $U_{best}(\lambda) = \text{argmin/max}_C [\sum_{t=0}^{T} \sum_{ci} (\vec{U}_i(A_i^t \backslash \lambda))]^T$

Return π_i

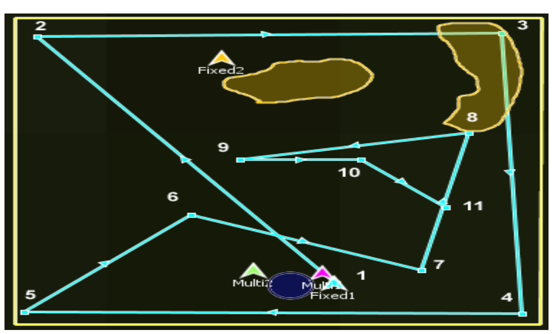


Figure 4. The Delaunay Triangulation Inspired Strategy Solution.

Figure 4 describes an example of the policy-based agent planning for single UAV(the blue one). The seed waypoints are waypoints labelled 1-5 same as figure 3. Waypoint 6's projection angle was obtained by taking $\Theta = 45^{o}$ ($\Theta = 180^{o}/4$ projected in the first quadrant and the number of expected waypoints in the layer is 4 base on the Delaunay triangulation rules), and the edge is the half of the opposing edge (i.e., half of the distance between waypoints 5 and 4). Waypoint 7,8,9 were obtained by following the same process but projected in different quadrants, and the edges are the half of the distance of the opposing edge. Waypoints 10 and 11 have the projection angles of $\Theta = 30^{o}$(i.e., $\Theta=180^{o}/4+2$) for the third layer two waypoints. Note that, this is one of the possible paths as other agents may choose different angles of projection, quadrants, and edges length when solving the whole MAP/DULAR at the PC level as describe by algorithm 3.

Algorithm 3: Complete MAP/DULAR Solution for Multiple Agents

Input: seed waypoint

Output: MAP/DULAR waypoints for n agents

Compute the number of agents needed

For all ai ∈ A do

For all π ∈ Π finds

$\pi_i \in \text{argmin/max}_{\pi \in \Pi^i} M^{\pi}$ //use algorithm 1 or 2

For all $v_i \in V$

Allocate $\lambda : A_i \rightarrow v_i, \forall v_i \in V$ s.t. $U_{best}(\lambda)$ $= \text{argmin/max}_{C} [\sum_{t=0}^{T} \sum_{ci} (\vec{U}_i(A_i^t \backslash \lambda))]^T$

For all ai ∈ A find

	$\pi_k^* \in \text{argmin/max}_{\pi \in \Pi^i} M(\pi_i)$
13:	Return $\pi^* = \{\pi_1^*, \pi_2^*, \pi_3^*, \ldots, \pi_k^*$

For the purpose of layer selection, the following concepts will be very important.

Definition 13. A waypoint is called **seed waypoint** when it serves as the initial (starting) point of the agent mission plan. For example, the waypoints 1 to 5 of figure 4.

Definition 14. Two waypoints X_{ij} and Y_{ij} , with dimension i and j, with upper and lower boundaries P_{ij}, Q_{ij} where i, j $\in R^n$ in MAP/DULAR plan are said to be reflected if and only if

$Y_i = Q_i-(X_i-P_i)$ or $Y_j = Q_j - (X_j-P_j)$ where i,j $\in R^2$ (4)

Where n is the dimension of the space.

Definition 15: Two waypoints X_{ij} and Y_{ij} in a MAP/DULAR environment with bounds P_{ij}, Q_{ij} where i,j $\in R^n$ are said to be refracted waypoints if and only if

$Y_{ij} = Q_{ij}-(X_{ij}-P_{ij})$ where i,j $\in R^2$ (5)

Definition 16: Displaced waypoint (V_{disp}) with edge e_i can be categorised as displaced in MAP/DULAR plan if there exists a waypoint in opposing direction.

For multiple agents, the initial waypoints can be reflected or refracted, and the agents will have different policies for generating layers' waypoints; for instance, figure 5 describes MAP/DULAR solution for the problem in figure 1 using algorithm 3 with individual agents waypoint selection using algorithm 2.

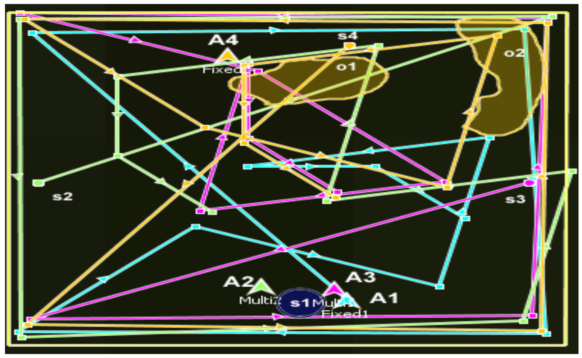


Figure 5. The MAP/DULAR Solution of Problem in Figure 1 using Algorithm 2.

Figure 5 describes the MAP/DULAR solution for multiple agents (four agents) using algorithm 2, agent 1 (A1) and agent 4 (A4) have refracted seed waypoint s1 and s4, likewise agent 2 (A2) and agent 3 (A3) with seeds s2 and s3. Each individual agent has a unique path according to its colour because they are using different policies. Even if the targets change position over time, the path can detect the target because they have different locations over time as designed by their respective policies.

Based on these concepts, we can generate theorems, for example, theorem 1.

Theorem 1:

Two different agents with different policies of generating layers waypoints have different MAP/DULAR solutions. Proof: inspired by the uniqueness of the Delaunay triangulations.

Theorem 2:

Convex MAP/DULAR using algorithm 2 have more coverage than the concave counterpart with the distance of the same edges.

Proof.

Let n be the number of points, and the convex of n be Conv(n). If m out of n points is concave, then the convex of n waypoints is Conv(n-m), i.e., using Graham's Scan algorithm. From Euler's formula the number of triangles T_{convex} formed for convex polygon will be:

T_{convex} = 2+E-n

While for a concave polygon it will be

$T_{concave}$ = 2+E – (n-m)

It means $T_{convex} > T_{concave}$ because n > n-m for m>0.

This proved that the number of triangles for a convex polygon is greater than the number of triangles for concave polygons while using Graham's Scan convex hull algorithm and Euler's formula. To proof, the area of convex is greater than the area of concave formula. Let Edisp be the concave edge of polygons $M_{concave}$, and let T_{disp} be the triangle carrying the displaced edge such that T_{disp} = Δ(n-1,Edisp,n+1). Let E be the E_{disp} with length λ when not concave and T be the area of the triangle Δn-1, λE, n+1. Then the total area of the polygon Mconvex = $\sum_i^n T_i$ and Mconcave = $\sum_i^n T_i - T_{disp}$ thus Mconvex>Mconcave.

The creations of more theorems for these algorithms is beyond the capacity of this paper and therefore mark as future work. Based on the various positions of the agents, one can see how the agents will swarm as if as they are communicating with one another and negotiating on the waypoints to visits as describe in the video at appendix section. Removing too much communication in MAP reduces the agents' resource consumption, for example, energy, memory, and computational capacities needed for sending and receiving messages, processing them, security maintenance were all saved in addition to the robustness and scalability improvement. Another interesting part of the policy-based coordination approach is that, it can be implemented on simplest agents possible base on the low resource demand. The agents will be executing policies on their memories, for instance, turn left after 2 minutes, turn right after 4 minutes and so on. The global policies of the agents make their behaviours seems to be intelligent. Hence, we are after the perfection of the agents' policies rather than making the agents very intelligent and complex as oppose to the existing works. Additionally, the waypoints distribution could be modelled as vectors and perform all linear algebra operation in finding satisfactory waypoint.

We believe that using simple agents to achieve the task will reduce the costs and technical efforts needed to solve MAP problem under the outlined hard constraints as well as increase the scalability, robustness, and security of the agents' swarm. For example, removing communication while solving MAP/DULAR enhances the security because radio signal jamming or hijacking will not be used in stopping the agents' from their mission as such improves the safety of the application of MAP/DULAR in defence surveillance and so on. Additionally, rules can be defined for approximating the number of agents needed given the sensor range and the environment size. Our observation shows that the number of agents needed for a given environment and sensor range can be approximated using the function in equation 6.

$$K(r_v) = \begin{cases} C_{uc} = 0 \; if \; r_v \; \geq 6.25\% \; of \; E \; i.e., sn = 4 \\ otherwise \;\; C_{uc} \sim \mathrm{f(sn)} \end{cases} \tag{6}$$

$$\mathrm{f(sn)}: \mathrm{Sn} \rightarrow \mathbb{Z}_o^+$$

The function K in equation 6 gives the approximated number of uncovered cells C_{uc}, when the environment is segmented into grid cells of a square size equivalent to the sensor range r_v. The variable sn means the square root of the total number of grids. For example, if the grids are 100 in number then sn = 10, when sn = 4 C_{uc} =0. If sn>4 the number of uncovered cells C_{uc} is approximated to positive prime numbers $\mathbb{Z}_o^+$ sequentially, i.e., if sn=5, then C_{uc} =2, etc. There is also possibilities for interesting flexibilities in the inter-agents' policies control which make this approach a more flexible one.

In order to show that, our approach also gives good search pattern, we simulate a forest fire searching mission on Aerospace Multi-UAVsSimulation Environment (AMASE) [32] using unmanned aerial vehicles (UAVs) as agents. Section 4 describes our results.

Practical Application to the Telecommunication Industry Problem

Summaryly, the practical application aimed at developing a swarm of UAVs to perform surveillance mission (remember, the surveillance can be utilised to address signal booster detection, properties monitoring, location security, etc.). The demonstration addresses two issues i.e., surveillance and signal boosting system. Both cases utilises the developed algorithm and the purchased drones.

The implementation of DIMASS on real UAVs is simple. The process starts by selecting the seeds waypoints (e.g., the longest non-cross waypoints of Figure 3), then a function can be developed to generate the remaining waypoints by taking some parameters, e.g., using the function *generateWaypoint*($L_x, L_y, e, q, \theta, h, n$), where L_x, L_y are the longitude and latitude of the current waypoint, e is the formula for the edge length of the opposing layer (from Algorithm 1), q is the projecting quadrant (i.e., first to fourth quadrants), 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 MAP/DULAR solution (i.e., based on the Delaunay triangulations formular discussed at the beginning of Section 3). In other words, the function *generateWaypoint*($L_x, L_y, e, q, \theta, h, n$) produces a waypoint based on current waypoint, opposing edge, angle(θ), number of waypoints in a layer, and the implemented rules as described in section 3. Waypoints latitudes and longitudes distance differences (i.e., for e) can be computed using the Haversine formula or Euclidean distance for planar coordinates. This can be implemented in any programming language (e.g., Java as in the supplemental documents).

The generated plan can be transferred easily to the UAVs using the respective UAVs' (drone's) mobile application (i.e., downloaded from either Google play store or Appstore), e.g., DJI GO, DJI pilots, FreeFlight 6, FreeFlight Pro, etc. for DJI and Parrots drones. Before testing the proposed algorithm on real UAVs, the researchers acquired the operator and flyer identification numbers (drone flying licenses) from the Civil Aviation Researcherity (CAA) of the United Kingdom. The flying took place in one of the areas of Birmingham, the United Kingdom, as shown in Figure 10, i.e., the real UAV flying experiment took place in the United Kingdom, whereas the fire spread experiment was conducted in Nigeria. Figure 10 shows the implementation of DIMASS on the DJI pilot mobile app to explore a search space (forest) starting from waypoint S to 11 (just like Figure 3). The plan creation and modification (i.e., quadrants, edges, and angles configurations) occurs simply by clicking and dragging of waypoints. The displayed distance between waypoints helps in defining edges length (as described in Figure 10). Additionally, the mobile apps allow plan saving and deletion (using the bin icon). Thus, the plan saving will allow routine lookout planning for the team of UAVs (i.e., waypoints plans can be saved and utilized for routine area searching). Alternatively, waypoints can be sent to the UAVs via python code for the programmable drones, e.g., DJI Tello Edu python Application Programming Interface (API). For larger UAVs, e.g., DJI matrices 100, onboard computers can be mounted to perform information analysis and other complex tasks. The researchers flew three UAVs, Parrot Bebob 1 and 2 and DJI Phantom 3. Each UAV has a controlling app running on a tablet (one tablet per UAV). The outcome is similar to the simulation results as described in Figure 6 to 8. Thus, the proposed DIMASS is easy to implement on real devices.

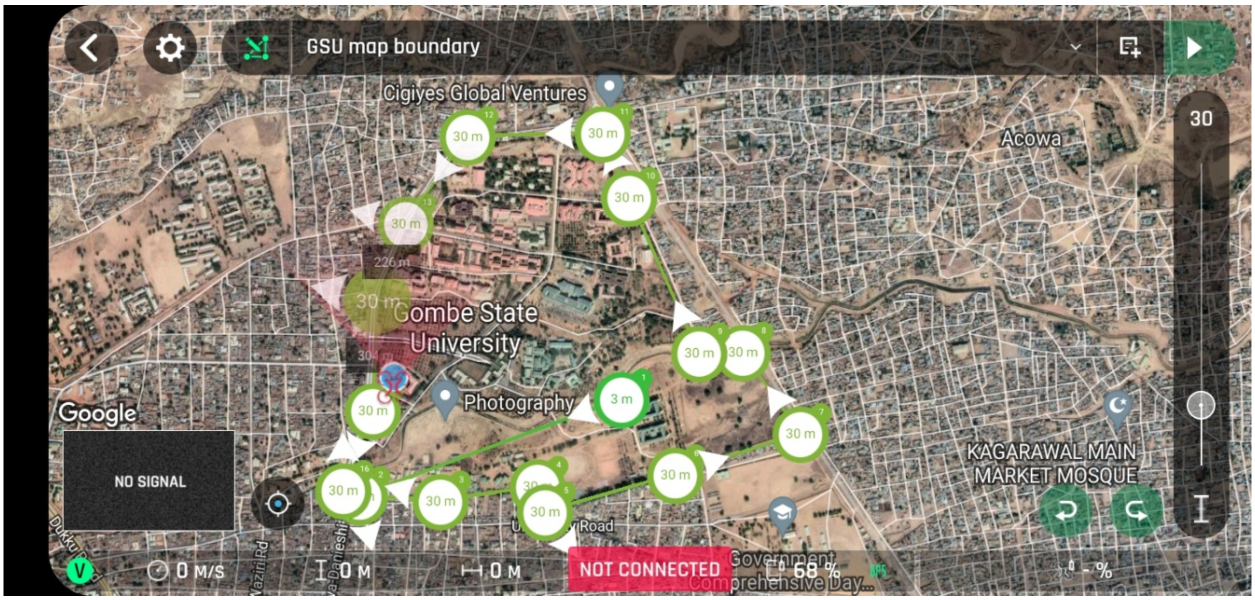


Figure 6: Drone 1 Path (Boundry Monitor)

Figure 7: Drone 2 Path

Figure 8: Drone 3 Path

Evaluation

This section gives the resources optimisation performance comparison of the proposed algorithms with other existing approaches against coverage, redundant search, energy, and overall complexity. The experiment involves the use of two UAVs to solve the MAP/DULAR problem in Figure 1, given the same time and energy. We assumed that the UAVs receive the initial point and policies from their PC. The area covered was measured by summing all the areas of the Delaunay triangles of the MAP/DULAR waypoints (i.e., by assuming the same sensor coverage). Table 3 describes the coverage, energy, and mission time performance comparison.

Table 3. Coverage Performance Comparison

Searching Strategy	**Covered Area (km^2)**	**Mission Time (minutes)**	**Energy Used**
Algorithm 1	834.8314	34	100%
Algorithm 2	962.3321	34	100%

Lévy Flight	288.39134	34	100%
Parallel Track	128.0284	34	100%
Creeping Line	252.6444	34	100%
Expanding Square Search	172.2770	34	100%
Zamboni Search	518.0589	34	100%

Table 4. Intra-Agent (r) and Inter-Agent (R) Redundant Search versus Sensor Range (r_v) Comparison

r_v	**Algorithm 1**		**Algorithm 2**		**Lévy flight**	
	r	**R**	**r**	**R**	**r(std for 15 values)**	**R(std for 15 values)**
5%	0	0	0	1	4 (2)	4.7 (2.11)
10 %	4	1	2	2	9 (4.76)	9.8(4.80)
15 %	5	2	3	3	7.9 (3.75)	9.8 (4.13)
20 %	5	3	4	4	9.4 (3.37)	10.9 (3.11)
25 %	5	3	4	4	11.2 (2.48)	13.3 (2.54)
30 %	6	3	5	4	11.5 (3.84)	14.2 (4.09)

35 %	6	4	6	4	14.9 (6.89)	18.2 (8.04)
40 %	6	5	6	4	12.6 (3.17)	15.3 (3.89)
45 %	6	5	6	5	17.6 (3.60)	18.26 (2.77)
50 %	6	6	6	7	18 (2.83)	21.6 (3.89)

Table 4 describes the number of intra-agent and inter-agent redundant waypoints comparison versus sensor range. For the Lévy flights, an average value of 15 experiments was taken; the values in the brackets are the standard deviations. This signifies that predicting agents' activities in random strategies like the Lévy flight is very difficult and hence not good for the DSA achievement process. Table 5 shows the comparison of the algorithms' quantitative and qualitative measures.

Algorithm	**Cyclomatic complexity**	**Big O Complexity**	**Line of Code**
Algorithm 1	19	$O(n^2 \log n)$	268
Algorithm 2	2	$O(1)$	763
Lévy flight	3	$O(1)$	327
Parallel track	7	$O(n^2)$	532
Creep lining	9	$O(n^2)$	300
Expanding square	7	$O(n^2)$	244
Zamboni Search	7	$O(n^2)$	241

The results in Tables 3 and 4 show that the proposed algorithms have higher coverage, redundant search reduction, agents' energy minimisation, mission time reduction, and agents negotiation reduction with the main limitation of memory consumption due to policies checking. This might not be an issue because it can be implemented on a UAV with a memory capacity of at least 298KB (assumed 50 characters per line of code).

Limitations

While we have demonstrated good performance of DIMASS, both in simulation and real UAVs, there is a number of limitations to explore in further work:

• Seed waypoints (first layer waypoints) need to be defined systematically. In the case of a large number of UAVs, the seed waypoints definition function must be systematic. For example, using four UAVs (Figure 4) is less challenging than 50 UAVs unless rules grouping is applied (i.e., a group of UAVs will be using different control protocols with others). Thus, in a large team of UAVs, seed waypoints control will be challenging. Therefore, layers configuration and control policies will be difficult. Although this scalability issue can be solved by using waypoints altitudes variations, Definition 3.2, Definition 3.3, Proposition 1, and seed waypoints variation, any non-organized set of plans could cause team disorganization (poor coordination). Thus, we agree that the type of search task used for our use case would be best performed by a team of around 4-7 UAVs which are controllable by a single SME (Baber et al., 2011).

• For a large number of layers and agents (e.g., UAVs), searching for the best solution could demand large computational resources, which must be bounded, e.g., by setting a number of iterations to avoid plan delay. For instance, configuring the best edges for 10 UAVs given 50 waypoints could require a lot of information exchange among UAVs. Therefore, large consumption of memory, communication bandwidth, and processing power will occur at the preparation stage.

• Policies for controlling plan updates need to be defined. As such, in the case of a large number of UAVs, this will be quite challenging to control.

• Collision avoidance requires Detect and Avoid (DAA) techniques using sensors or some form of the organization, e.g., stop and pass rules (i.e., stop and wait rules to avoid collision), waypoints altitudes variation, etc. Again, implementing effective collision avoidance for a large number of UAVs and waypoints will be difficult.

Chapter 2: Application of Bayesian Belief Network (BBN) to Model Situation Awareness in Telecommunication Industry

In this chapter, we will describe how BBN can be utilised to managed SA and data in telecom industry.

Definition of Situation Awareness in Telecommunication Industry and its Effect on Quality of Service Monitoring

Focusing on the project aim, our definition of Situation Awareness in the telecommunication industry refers to the ability to obtain a model or framework that allows easy detection, understanding and projection (prediction or estimation of missing variables), which will improve decision making and mode of operations. This is applicable in many operations of the regulatory agency (Nigerian Communications Commission), such as signal strength monitoring, quality of service monitoring (e.g., complaints from customers and response etc.), radio planning etc. with regards to the swarm of autonomous drones, we focus on the ability to obtain an autonomous algorithm that will coordinate a swarm of drones to conduct surveillance missions for the telecommunication industry's activities. For example, assuming a swarm of drones to monitor radio signal emission or radio planning, the effectiveness of the mission relied on how these drones (Unmanned Aerial Vehicles) coordinate themselves (based on the monitoring algorithm) to extract information, submit it to the comprehension model and supports decisions, whereas, reports three focus on the algorithms for autonomous swarm coordination. Note that the application of the research can be applied not only to the telecommunication industry but also to other areas within the country, e.g., the security sector, surveying, farming, etc. This report focuses on information understanding and training (using machine learning algorithms). In summary, at its simplest and more generic level, Situation Awareness in the Nigerian Telecommunication industry requires the ability to gather data, understand it, and makes effective decisions on plausible future system situations.

Knowledge Presentation

A clear challenge to SA is how to present the collected information (either from the autonomous swarm of drones surveillance or other operations, e.g., complaints from mobile phone users to NCC) and to logically understand it to make a decision. Existing works focus on the application of concepts maps, propositional networks, ontology, and fuzzy logic[27], [33]–[38]. Our model proposes the use of Bayesian Belief Network (BBN) due to its ability to handle uncertainties, measure beliefs, and makes situational-based predictions [39], [40].

A BBN is a directed graph that reflects a configuration matrix, C_{ij}. This matrix defines the probabilities of links between nodes (where the nodes represent concepts). Probabilities can be specified using a weight matrix, W_{ij}. Thus, in a BBN graph G: $C_{ij} \rightarrow W_{ij}$. An advantage of using BBN for DSA is that the probabilities can adapt to the situation. As the situation unfolds, then the BBN must be updated (either by introducing new nodes or by altering links between nodes) to reflect changes in perception or comprehension of the agents in that situation. Figure 1 describes an example of BBN for Quality of Service monitoring signal strength by multiple of drones (UAVs) by generating waypoints. For example, Figure 1 describes a NETICA developed BBN for signal monitoring using a swarm of UAVs. Thus, the task of the network is to present the information gathered by the swarm of the UAVs.

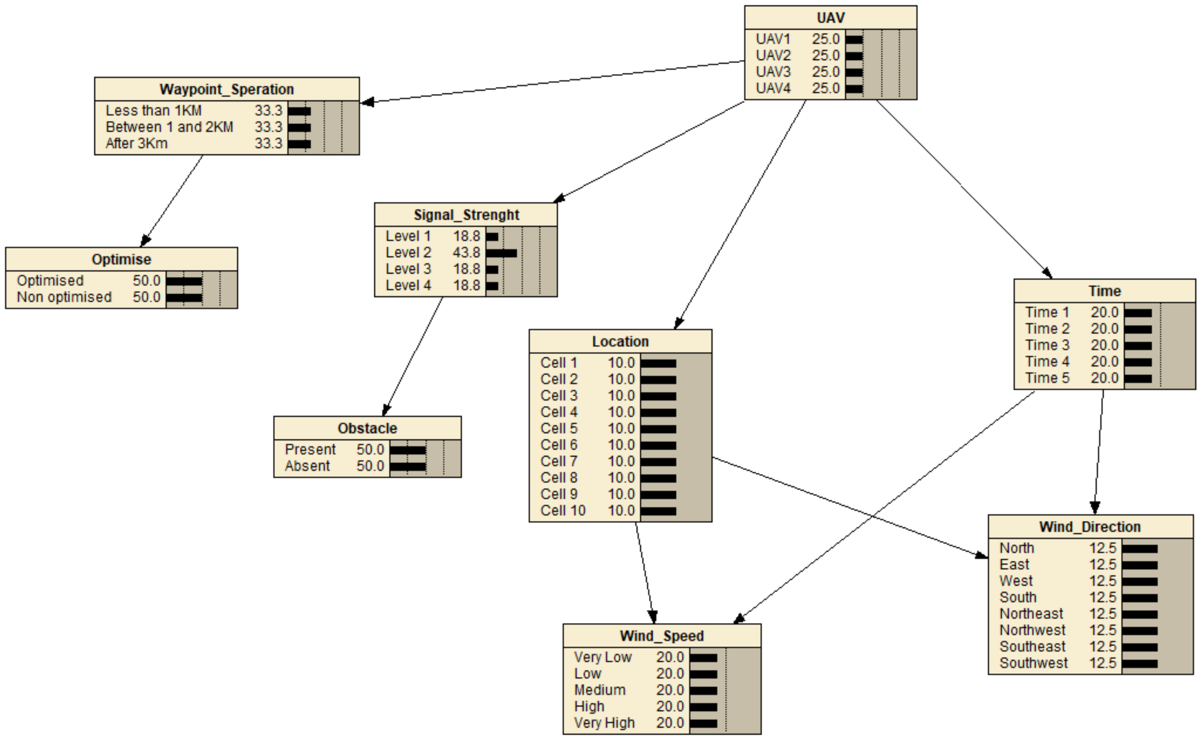


Figure 1: Example Bayesian Belief Network for Signal Monitoring using Swarm of Four UAVs

Figure 1 describes an example of BBN responsible for pressing multiple UAV information. The UAV nodes indicate the detecting UAV, and the time UAV indicates the time frame. The waypoint separation denotes how far the current waypoint is with respect to other waypoints, whereas optimise nodes are responsible for deciding whether the waypoint is optimised or not. The signal strength decides on the strength of the network and whether the detection has an obstacle, wind speed, etc., that could interfere with the value[41], [42]. Thus, the task of the swarm of UAVs is to explore the space and gather the outlined information.

BBN Definition and Construction

The following steps describe steps for BBN construction

Step 1: Select your nodes (situations):

In this step, you need to select the information needed for understanding the situation for the mission. For this guide, the mission goal is to understand signal strength as described in Figure 1. Based on the Standard Operating Procedure (SOP) of signal in the Nigerian telecommunication industry. The signal strengths change randomly more especially in the presence of obstacle or interference[43], [44]. Therefore, the following situation nodes are extracted as a sample: {optimisation, signal strength, location, time, UAV, waypoint difference, and obstacle}

Step 2: Identify your nodes states

Based on the SOP knowledge, the state of every situation node needs to be identified. Continuous variables need to be discretised to allow effective learning. For example, the node signal strength could be a range of signal strength values to determine being strong or weak across various levels.

Step 3: Identify the Relationship among your Information Nodes

Just like the propositional network or concept map, sketch the relationships (what influences or causes what?) among your information nodes as described in Figure 1. The output is the BBN for the outlined operation.

1.1 BBN Nodes Types

This section describes various types of nodes that can be used for BBN development and Situation awareness.

a. Cost (utility) nodes: defines the cost implication of agents’ action. For example, From Figure 1, the “Optimisation” measures how optimal a particular waypoint is to the mission.

   Thus, the Bayesian Belief Network describes not only how agents can make decisions on variables in DSA but also paves an easy way of measuring the success of the mission. That is, by summing the optimisation grades probabilities. Additionally, human agents' contributions can be incorporated to control the agents' behaviours in more sensible ways, i.e., given the Conditional Probability Tables (CPTs, i.e., a table of probability for each node entry of the dependent node), the agent could control all its actions effectively by searching for high-grade optimisation (i.e., high probability CTP entries). For instance, the optimisation node could be kept at an optimised state whenever possible.

b. Awareness nodes: represent how various nodes states could be joined to understand a particular situation. For example, consider the "UAV" node of Figure 1; it aims to understand the information gathered by various UAVs, e.g., the location, time, signal strength, etc.

c. Situation node: represents the ordinary perception node belief. The probabilities of the situation node could be updated using the following process. Having established a high-level BBN for the situation, the next step is to apply this to a Mission. To manage the change in weight due to incoming information, we apply parametric learning. To illustrate this, assume a simple UAV has a sensor that defines the obstacle presence of Figure 1 (as 1 for 'present' or 0 if the information does not exceed a threshold for 'present'). On initiation, the UAV will have a belief, *B*, for its ‘obstacle’ element. Assume a cold start weight with equal probability for present / absent (this could change, depending on the SME weight or feature of the situation). 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 an obstacle present. From this, *p*(fire = present) increases from 0.5 to 0.75, i.e., *p*(fire = present) = $B_1 * I_1 + 1 / I_2 (I_1 + 1) = 0.5 * 1+1 / 1*(1+1) = 1.5 / 2 = 0.75$. On the second report, its sensors make another positive report. In this case, *p*(fire = present) increases; $(0.75 * 2 +1) / 3 = 0.83$. Alternatively, highly critical states need to be updated rapidly. For example, if an obstacle is detected from a reliable sensor (say, an experienced human), the belief will be rapidly increased to 1. To support this, an intensifying factor, *k*, can be applied.

Thus, based on the nodes categorisation, the transition of agent’s SA states of perception, comprehension, and projection state at every situation while solving the swam goal in the DSA system could be maintained using the CPTs entries.

1.2 BBN Prediction and Uncertainty Handling in Situation Awareness using Machine Learning

One of the main advantages of using BBN for knowledge presentation is the ability to apply machine learning algorithms to make predictions and handle mission variables. For this task, we applied the Expectation-Maximisation (EM) algorithm. The EM algorithm utilises the following steps.

Step 1. (Expectation State)

Predict the target state (S_i) using the Bayes rule for the previous trends data (R_r).

$$P(S_s|R_r) = P(R_r|S_s)P(S_s)/P(R_r) \qquad (1)$$

That is, the algorithm will compute the probability of each state (S_s as defined in Equation 3) given related states priors (R_r). The state with a maximised value of likelihood, $P(R_r|S_s)$, will be selected. Note that the priors and likelihood for this computation will be generated from entities sensor states using the algorithm described in the above section. Thus, mostly occurring states have higher chances of being selected which indicates a Situation-Based prioritisation. At the maximisation state(Step 2), the likelihood ($P(R_r|S_s)$) will be maximised based on previous values.

Step 2: Maximisation

This step maximises the likelihood of a chosen future state using an iteration process.

$$P(S_i) = argmax_{S_i} P(S_i|R_r) \qquad (2)$$

At each iteration, the algorithm replaces the selected value with the revised value (i.e., revised by going through the previous values). Iteration continues until convergence (i.e., the difference between the previous value and the current value is negligible or maximum iteration is reached as specified by the human SMEs). An intuitive feature of this algorithm is that during each iteration, the predicted value likelihood is greater than or equal to the previous value [45]. That is, the current belief values at each iteration satisfied Jensen's inequality [45] conditions of Equations (4) and (5) [45] (i.e., prediction/estimation value increases at every iteration). To formalise this with DSA, a state probability is either active (increasing) or in-active (decreasing) knowledge of the system (Table V). Thus, both likelihoods conditions are captured using Equations 3 and 4.

Belief increment:

$$P(\int_{l_1}^{l_{1+1}} zB(R_r|S_s))\, dS \leq \int_{l_1}^{l_{1+1}} zP\big(B(R_r|S_s)\big)dS \qquad (3)$$

Belief decrement:

$$P(\int_{l_1}^{l_{1+1}} zB(R_r|S_s))\, dS \geq \int_{l_1}^{l_{1+1}} zP(B(R_r|S_s))dS \qquad (4)$$

where B(Ss) and B(Rr) are the probabilities of the querying and related states of the BBN, and z is the attached weight factor for human SME critical Weight (human assigned weight) and/or related nodes influence measure on current goal.

Equations (3) and (4) indicates when a situation belief (probability value) is increasing or decreasing, respectively. Thus, the objective of the iteration process at every time-space T of the EM algorithm is to find the maximised prediction/estimation of the belief of the state with a high likelihood value for the future time, i.e., $P\big(B(S_i)\big)^{t+1}$.

$$P\big(B(s)\big)^{t+1} = argmax_{t \prod_{t=1}^{t=n} P(B(S_i)^t}. \quad (5)$$

Where $P\big(B(S_s)\big)^{t+1}$ *and* $P\big(B(S_s)\big)^{t}$ *is the BBN state probability estimation before and after an iteration.*

Finally, changes in projection accuracy are important in determining how the projection/estimation activities in the DSA system is performing over time the period.

The outcome of the learning process using the EM algorithm can be evaluated using the error rate (number of incorrect predictions) and other metrics such as the logarithm loss, Brier score, and surprise rate (the number of times the EM algorithm made a prediction with X% confidence and the prediction becomes wrong)[46], [47]. For example, consider the swarm of four UAVs data for the BBN in Figure 1 (the autonomous swarm algorithms are described in report 3). Figure 2 describes the excerpt of the data.

NumCases	Location	UAV	Obstacle	Waypoint_Speration	Optimise	Signal_Strenght	Time	Wind_Speed	Wind_Direction
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Very_Low	North
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Very_Low	East
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Very_Low	West
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Very_Low	South
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Very_Low	Northeast
0.00260417	?	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Very_Low	Northwest
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Very_Low	Southeast
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Very_Low	Southwest
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	?	North
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Low	East
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Low	West
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Low	South
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Low	Northeast
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Low	Northwest
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Low	Southeast
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Low	Southwest
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Medium	North
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Medium	East
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Medium	West
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Medium	South
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Medium	Northeast
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Medium	Northwest
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Medium	Southeast
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	Medium	Southwest
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	High	North
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	High	East
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	High	West
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	High	South
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	High	Northeast
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	High	Northwest
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	High	Southeast
0.00260417	Cell_1	UAV1	Present	Less_than_1KM	Optimised	Level_1	Time_1	High	Southwest

Figure 2: Excerpt of UAVs Sensor Data

Figure 2 describes an example of mission data (which is demonstrated by the swarm of UAVs mission, but could be applicable to other operations, e.g., customer complaints). Missing information can be identified using "?" or "*". We claim that this approach could suit most of the NCC operations because it accepts data in the form of excel documents(.xls, .csv, etc.) or text. The learning output will bring the outcome of the prediction, e.g., a 10% error rate. Results on this are under publication with the IEEE journals (so the report just describe the idea, but we are ready to apply it to any of the NCC's operation).

The versatility of the Model to Telecommunication Industry

The proposed machine learning model based on BBN is applicable to most of the operations of the Nigerian Telecommunication Industry Quality of Service (QoS) monitoring. For example, the records of customer complaints can be modelled, and predictions and missing variables can be performed. This would support the agency's decision on best performing service, forecast the future, and take all the necessary measures before the occurrence of the events. For example, the customers' complaints data from [48] can be predicted using the proposed BBN model, which could reduce the number of complaining by making predictions and allowing early action-taking. A journal and conference paper on this specific matter will be published, subjected to the consent of the NCC.

Our progress achieved the following:

1. We developed a framework that could provide any of the NCC's operation prediction, and uncertainty handling. This could support the commission's decision-making process by allowing predictions and uncertainty handling. For example, Figures 3 and 4 described an example of the machine learning result to predict a telecommunication signal strength using BBN.

```
Read 4 cases, and used 4 of them to test net.

For Signal_Strenght:
-------------------

Confusion:
  ...........Predicted..........
  Level_  Level_  Level_  Level_    Actual
  ------  ------  ------  ------    ------
       0      19       0       0    Level_1
       0      44       0       0    Level_2
       0      19       0       0    Level_3
       0      19       0       0    Level_4

Error rate = 56.25%

Scoring Rule Results:
  Logarithmic loss = 1.303
  Quadratic loss   = 0.7031
  Spherical payoff = 0.5449

Calibration:
  Level_1    0-20:     18.8 |
  Level_2    0-50:     43.8 |
  Level_3    0-20:     18.8 |
  Level_4    0-20:     18.8 |
  Total      0-20:     18.8 | 20-50:    43.8 |

Times Surprised (percentage):
               .................Predicted Probability...................
  State        < 1%             < 10%             > 90%             > 99%
  -----        ----             -----             -----             -----
  Level_1      0.00  (0/0)      0.00  (0/0)       0.00  (0/0)       0.00  (0/0)
  Level_2      0.00  (0/0)      0.00  (0/0)       0.00  (0/0)       0.00  (0/0)
  Level_3      0.00  (0/0)      0.00  (0/0)       0.00  (0/0)       0.00  (0/0)
  Level_4      0.00  (0/0)      0.00  (0/0)       0.00  (0/0)       0.00  (0/0)
  Total        0.00  (0/0)      0.00  (0/0)       0.00  (0/0)       0.00  (0/0)
```

Figure 3: Signal Strength Prediction of Figure 1 using Test Cases

```
Sensitivity of Test:
  Level_1    0     100    | 20    0      | 100   0      |
  Level_2    0     100    | 50    0      | 100   0      |
  Level_3    0     100    | 20    0      | 100   0      |
  Level_4    0     100    | 20    0      | 100   0      |
```

Figure 4. Sensitivity Test for Each State of the Node of Signal Strength Node.

Note that the proposed machine learning using EM algorithm can be applied to all of the NCC's operations, such as complaints monitoring, radio planning, etc.

Chapter 3: Signal Boosting Testing System using Swarm of UAVs

The signal boosting detecting system follows these simple states

1. Mounting frequency measuring device on the drone: the frequency measuring device detects higher frequency for signal boosting presence and announce that using high pitch beep sounds. This is then transferred to the monitors (Quality of Service Monitors) via a microphone
2. Attached microphone: the microphone will then receives the detection beep sounds and transmit it to the base station's speaker.
3. Broadcasting speaker: the broadcasting speaker announces the signal boosting detection.

Figure 1 describes our implementation in Gombe.

Figure 1: Signal Boosters Installation Detection using Drones

Figure 2: Signal Frequency Detector

Figure 3: Microphone

Figure 4: Speaker

A clear limitation of this approach is the payload on the drone. As such, in our future work we will utilised microcontrollers and locally developed items to reduce the payload.

Chapter 4: Anti-Drone System

We develop an anti-drone system (in case of surveillance intrusion) based football nets and a clip. The idea is to be dropping the net on top of the intruding drone. Figure 1 to 4 describes the whole idea. We provide the video of the system in our repository folder (https://bit.ly/3o1wKvb).

Figure 1: Anti-Drone System Setup (The Scene was Captured During NCC Staffs Visit)

Figure 2: Anti-drone Scene 2

Figure 3: Anti-drone System Scene 3

Figure 4: Anti-drone System (Drone Catching Scene)

Future work will look at fully automated system enhance with computer vision. That is, the drones detection will be vision-based obtained using machine learning.

Conclusion and Future Work

We applied the concept of multi-UAVs (multi-drones) swarm to the various aspect of surveillance and quality of service monitoring telecommunication industry. Novel algorithms were developed to handle the agents' coordination task with minimal resources. In terms of agents' data and Situation Awareness (SA) management, Bayesian Belief Network (BBN) were applied. The outcome shows a robust way of managing the acquired data and potentials for machine learning which could assist in predictions and uncertainty handling. The project further developed signal boosting and anti-drones systems on top of the agents coordination and SA management work. This is in addition to the journals and conference papers publications. We also utilised the grant funds in training more than One Thousand (1000) Nigerians on the opportunities in multi-drones research field. Future work will focus on further of the developed systems.

Appendices

Appendix A: Expenditures

Appendix B: Training Galleries

[1] A. Zulu and S. John, "A Review of Control Algorithms for Autonomous Quadrotors," *Open Journal of Applied Sciences*, vol. 04, no. 14, pp. 547–556, 2014, doi: 10.4236/ojapps.2014.414053.

[2] T. M. Cabreira, K. Kappel, L. B. de Brisolara, and P. R. Ferreira, "An Energy-Aware Real-Time Search Approach for Cooperative Patrolling Missions with Multi-UAVs," in *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, Nov. 2018, pp. 254–259. doi: 10.1109/LARS/SBR/WRE.2018.00054.

[3] H. Ucgun, U. Yuzgec, and C. Bayilmis, "A review on applications of rotary-wing unmanned aerial vehicle charging stations," *International Journal of Advanced Robotic Systems*, vol. 18, no. 3, p. 17298814211015864, May 2021, doi: 10.1177/17298814211015863.

[4] A. A. A. Alkhatib, "A Review on Forest Fire Detection Techniques," *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, p. 597368, Mar. 2014, doi: 10.1155/2014/597368.

[5] X. Zhou, W. Wang, W. Tao, L. Xiaboo, and J. Tian, "Continuous patrolling in uncertain environment with the UAV swarm," 2018.

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0202328 (accessed Aug. 18, 2019).

[6] A. Jahn, R. J. Alitappeh, D. Saldaña, L. C. A. Pimenta, A. G. Santos, and M. F. M. Campos, “Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments,” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 273–278, 2017, doi: 10.1109/ICRA.2017.7989035.

[7] K. S. Kappel, T. M. Cabreira, J. L. Marins, L. B. de Brisolara, and P. R. Ferreira, “Strategies for Patrolling Missions with Multiple UAVs,” *Journal of Intelligent & Robotic Systems*, vol. 99, no. 3–4, pp. 499–515, Sep. 2020, doi: 10.1007/s10846-019-01090-2.

[8] P. Bouvry *et al.*, “Using heterogeneous multilevel swarms of UAVs and high-level data fusion to support situation management in surveillance scenarios,” in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2016, pp. 424–429. doi: 10.1109/MFI.2016.7849525.

[9] E. D. Breejen *et al.*, “Autonomous Forest Fire Detection,” in *Proceedings of AusWeb 2000, the Sixth Australian World Wide Web Conference*, 1998, pp. 167–181.

[10] L. B. Ingle, “Every Day Is Fire Day: A Study of Historic Fire Towers and Lookout Life in the Great Smoky Mountains National Park,” p. 136, 2011.

[11] L. Merino, F. Caballero, J. R. M. Dios, J. Ferruz, and A. Ollero, “A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires,” *Journal of Field Robotics*, vol. 23, no. 3–4, pp. 165–184, 2006, doi: 10.1002/rob.20108.

[12] D. Zhang and H. Duan, "Identification for a reentry vehicle via Levy flight-based pigeon-inspired optimisation," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 232, no. 4, pp. 626–637, Mar. 2018, doi: 10.1177/0954410016682274.

[13] X. Yang and Suash Deb, “Cuckoo Search via Lévy flights,” in *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, Dec. 2009, pp. 210–214. doi: 10.1109/NABIC.2009.5393690.

[14] S. G. Nurzaman, Y. Matsumoto, Y. Nakamura, S. Koizumi, and H. Ishiguro, “Biologically inspired adaptive mobile robot search with and without gradient sensing,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, Oct. 2009, pp. 142–147. doi: 10.1109/IROS.2009.5353998.

[15] M. Chawla and M. Duhan, “Levy Flights in Metaheuristics Optimization Algorithms – A Review,” *Applied Artificial Intelligence*, vol. 32, no. 9–10, pp. 802–821, Nov. 2018, doi: 10.1080/08839514.2018.1508807.

[16] A. Karduni, D. Markant, R. Wesslen, and W. Dou, "A Bayesian cognition approach for belief updating of correlation judgement through uncertainty visualisations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 978–988, Feb. 2021, doi: 10.1109/TVCG.2020.3029412.

[17] O. Ganoni and R. Mukundan, “A Framework for Visually Realistic Multi-robot Simulation in Natural Environment,” *arXiv:1708.01938 [cs]*, Aug. 2017, Accessed: Feb. 19, 2019. [Online]. Available: http://arxiv.org/abs/1708.01938

[18] G. Bevacqua, J. Cacace, A. Finzi, and V. Lippiello, “Mixed-initiative Planning and Execution for Multiple Drones in Search and Rescue Missions,” in *Proceedings of the Twenty-Fifth International Conference on International Conference on Automated Planning and Scheduling*, Jerusalem, Israel, 2015, pp. 315–323. Accessed: Feb. 19, 2019. [Online]. Available: http://dl.acm.org/citation.cfm?id=3038662.3038706

[19] D. W. Casbeer, R. W. Beard, T. W. McLain, and and R. K. Mehra, “Forest fire monitoring with multiple small UAVs,” in *Proceedings of the 2005, American Control Conference, 2005.*, Jun. 2005, pp. 3530–3535 vol. 5. doi: 10.1109/ACC.2005.1470520.

[20] J. D. Griffith, M. J. Kochenderfer, R. J. Moss, V. V. Misic, V. Gupta, and D. Bertsimas, “Automated Dynamic Resource Allocation for Wildfire Suppression,” 2017.

[21] F. João, “Search Patterns,” 2012, [Online]. Available: https://paginas.fe.up.pt/~ee07245/wp-content/uploads/2012/04/pattern.pdf

[22] A. Marjovi, J. G. Nunes, L. Marques, and A. T. de Almeida, “Multi-robot exploration and fire searching,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1929–1934, 2009, doi: 10.1109/IROS.2009.5354598.

[23] G. Paul, “Search patterns.” [Online]. Available: https://www.instructortoolkit.co.uk/app/download/5847990162/Search_Patterns-RIBInt%27l.pdf?t=1390308881

[24] B. Saicharan, R. Tiwari, and N. Roberts, "Multi Objective optimisation based Path Planning in robotics using nature inspired algorithms: A survey," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Jul. 2016, pp. 1–6. doi: 10.1109/ICPEICES.2016.7853442.

[25] P. Vincent and I. Rubin, “A Framework and Analysis for Cooperative Search Using UAV Swarms,” in *Proceedings of the 2004 ACM Symposium on Applied Computing*, New York, NY, USA, 2004, pp. 79–86. doi: 10.1145/967900.967919.

APPENDIX A

Photo A1: Gombe State Surveyors Trainees during our Workshops

Photo A2: Group Picture with a set of

Photo A3: Cross-section of Workshop

More gallery and videos can be found here: https://bit.ly/3o1wKvb