sensors

MDPI

Article

# Internet-Of-Things in Motion: A UAV Coalition Model for Remote Sensing in Smart Cities

**Adiel Ismail** *,†, **Bigomokero Antoine Bagula** † and **Emmanuel Tuyishimire** †

Department of Computer Science, University of the Western Cape, Private Bag X17, Bellville 7535, South Africa; abagula@uwc.ac.za (B.A.B.); tuyinuel@gmail.com (E.T.)

* Correspondence: aismail@uwc.ac.za; Tel.: +27-21-959-3879

† These authors contributed equally to this work.

Received: 18 June 2018; Accepted: 2 July 2018; Published: 6 July 2018

**Abstract:** Unmanned aerial vehicles (UAVs) or drones are increasingly used in cities to provide service tasks that are too dangerous, expensive or difficult for human beings. Drones are also used in cases where a task can be performed more economically and or more efficiently than if done by humans. These include remote sensing tasks where drones can be required to form coalitions by pooling their resources to meet the service requirements at different locations of interest in a city. During such coalition formation, finding the shortest path from a source to a location of interest is key to efficient service delivery. For fixed-wing UAVs, Dubins curves can be applied to find the shortest flight path. When a UAV flies to a location of interest, the angle or orientation of the UAV upon its arrival is often not important. In such a case, a simplified version of the Dubins curve consisting of two instead of three parts can be used. This paper proposes a novel model for UAV coalition and an algorithm derived from basic geometry that generates a path derived from the original Dubins curve for application in remote sensing missions of fixed-wing UAVs. The algorithm is tested by incorporating it into three cooperative coalition formation algorithms. The performance of the model is evaluated by varying the number of types of resources and the sensor ranges of the UAVs to reveal the relevance and practicality of the proposed model.

**Keywords:** smart cities; Internet-of-Things; multi-drone task allocation; unmanned aerial vehicles; path planning; Dubins curves; particle swarm optimization

## 1. Introduction

Drone technology has recently moved from a niche area mainly controlled by the American Department of Defense into a technology that can be accessed off-the-shelf for different purposes and tasks. Presently, drones are increasingly used for tasks that are either too dangerous, expensive or difficult for human beings [1]. Drones are also used in cases where a task can be performed more economically or more efficiently than if done by humans. Target search is a common application widely used in missions involving rescue [2,3], monitoring [4,5] or destruction [6,7]. Area coverage is also a popular application for multiple purposes, which include environment mapping [8,9], surveillance [4,10–14], sensor deployment [15,16], aerobiological sampling [17] or acting as communications hubs for immobile wireless sensor networks [15,18]. UAVs can also be used to provide data muling services in applications such as smart parking [19,20] and drought mitigation [21]. Other applications include fire detection, weather forecasting, space exploration, traffic surveillance, environmental clean-up, agricultural monitoring and interior surveillance of buildings. New and unforeseen applications also continue to surface. In Israel and Australia, UAVs are being used to search for groundwater. More recently, commercial applications using drones have emerged in niche areas where these vehicles have been found much more practical and more economically sound than traditional applications. These include cases where drones are being used commercially: (1) as marketing gimmicks, (2) to deliver pizza by Domino's pizza in the U.S., (3) to deliver flowers on Valentine's Day and (4) to deliver beer during a festival. Drones equipped with cameras have been deployed to help combat the scourge of rhino-poaching on game farms in Africa, while the Australian defense force has started using UAVs to monitor its vast coastline. Facebook and Google have recently acquired aerospace companies Ascenta and Titan Aerospace, respectively, and are investing in drones for supplying remote areas on Earth with broadband and Internet connectivity. It is envisaged that their solar-powered drones will be capable of staying in the air for months and beam WiFi signals to inaccessible areas of the world. In 2014, Alec Momont, in the Netherlands, developed an ambulance drone that delivers an automatic external defibrillator to patients via air, much faster than a standard ambulance [22]. This invention could bring relief to many of the approximately 360,000 patients who experienced cardiac arrest in the United States last year. These applications along with the potential for drones to deliver life-saving medicines and supplies to isolated communities in rural and disaster zones when overland access is not an option make the multi-drone task allocation an interesting research area that may benefit both urban and rural areas of the world. The use of a team of drones instead of a single drone is possible when the mission to be performed can be decomposed into independent tasks, i.e., the mission 'is inherently distributed in terms of space, time and functionality' [23]. However, path finding and resources allocation, both aerial and ground-based, are two key challenging issues that need to be addressed in order to provide a team of drones with the necessary autonomy to achieve a task cooperatively.

### *1.1. Related Work*

In [24], Dubins presented a solution for the shortest path between two positions given a car's direction or heading at both positions. His solution consists of all combinations of arcs of minimal turning radius of the car and a straight line segment joining the two arcs. The shortest path solution provided by Reeds and Shepp in [25] included both forward and backward movement of the vehicle. In [26], Kavraki et al. constructed a probabilistic road map of all feasible paths between the source and target, and the path with the lowest cost was then selected as a solution. In [27], a swarm of UAVs with airborne sensors cooperated in path planning by using Dubins curves in conjunction with splinegons, which are generalizations of polygons with sides of constant curvature, to detect the boundaries of contaminant clouds in order to model and track their movement. In [28], a path generator was proposed that finds the shortest path for a fixed-wing UAV from its current position and direction of flight to a new position and direction. The generated path was traversed using their Lyapunov path-following scheme. In [29], a path planning approach for mobile robots was modeled as an optimization problem with the objective of achieving a smooth path linking several waypoints by using fifth order Bézier curves. In [30], a path planning model was proposed for many UAVs to cooperatively collect data at different locations. A particle swarm optimization (PSO) based distributed coalition auction algorithm for real-time task assignment that uses a dynamic bidding strategy to address target assignment conflicts between UAVs, which resulted in shorter mission completion times, while still maintaining a high mission completion rate, was proposed in [31] by Lin et al. In [32], Fargeas et al. used a heuristic algorithm to plan the paths of a team of UAVs patrolling a network of roads and pursuing intruders detected by unattended ground sensors where UAV paths were scheduled using revisit deadlines that resulted in most of the intruders being identified. In [33], Kwok et al. used PSO to determine drive commands, speed and turning, to coordinate the motion of construction vehicles for task assignment by minimizing the distance and difference in orientation between the current positions of the vehicles and the positions of the targets. Collisions between vehicles were avoided using a behavioral-based reactive approach in conjunction with a dynamical indexing schedule that reduced dead-lock events. Bellingham et al. in [34] solved large problems involving many obstacles and waypoints by partially decoupling the task allocation and trajectory design aspects of the fleet coordination problem. Completion times for the various allocation scenarios were efficiently estimated and then passed on to the allocation problem that was solved using mixed-integer linear programming. In [35], Tong et al. successfully applied a cooperative control strategy based on Voronoi diagrams and discrete PSO to cooperative-trajectory planning problems with timing constraints. Path planning and task allocation are crucial aspects in the application of a fleet of UAVs to remote sensing. UAVs collecting data directly from targets close to each other often result in non-optimal flight paths due to the limitation of a UAV's minimum turning radius. In [36], a UAV path planner was presented that collected sensing data using a minimal distance by exploiting the sensor footprint and by pruning a tree of feasible paths, which was based on the learning real-time A* search algorithm. Road network search was defined as a graph in [37] and first solved as a Chinese postman problem and then modified to address the physical constraints of UAVs before using a multiple multidimensional knapsack in conjunction with Dubins paths to find the shortest flight path. In [38], task allocation was decentralized, and three complementary algorithms based on swarm intelligence and multi-agent systems were presented, which outperformed current approaches. In [39], a UAV's flight path was adapted based on fluctuations detected for several performance parameters, and the path planner calculated an optimal flight path by considering: avoidance of obstacles using a probabilistic risk model, the limitation of the maneuverability of a UAV and the velocity and flight time to a target.

### 1.2. Contributions and Outline

This paper proposes a collaborative strategy and mechanisms to control multiple UAVs and sensor networks with the objective of surveillance and achieving remote sensing in a city. A potential deployment scenario for the model presented in this paper consists of remote sensing in a city where different environment parameters need to be collected by a team of drones at separate locations of interest. The environment sensing parameters may differ from location to location, and different drones can be equipped with different sensing resources, thus leading to a model where drones need to form coalitions in order to meet the sensing requirements of different locations. The remote sensing services may consist, for example, of (i) pollution monitoring of different pollutants to be measured at different locations of the city and (ii) visual sensing by a number of drones coming from different angles to capture a view of a scene of a given location as depicted in Figure 1. In this scenario example, the pollution monitoring parameters and visual information captured at different locations will be ferried to a processing place where the different portions of the scene are stitched together to reproduce the whole scene of the location, and levels of pollution are evaluated at the location using machine learning techniques.

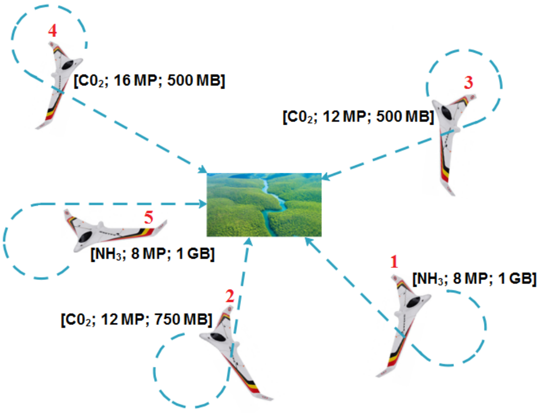


**Figure 1.** Visual sensing and pollution monitoring scenario.

The main contributions of this paper include:

1. An algorithm to calculate the Dubins longest path using elementary geometry is presented.

2. For successful collection of environment parameters from various locations, agents need to have sufficient storage capacity that can accommodate all the parameters. Agents in the experiments in this paper are equipped with sufficient storage capacity that can accommodate all the environment parameters collected from the various locations.
3. The effect of the increase of the number of types of environment parameters on the performance of the algorithms is investigated.
4. The effect of the increase of the sensor range on the performance of the algorithms and mission completion time are also investigated.
5. The energy gain of the proposed approach in (2) is quantified.

This paper borrows from Manathara et al. in [40] the three coalition formation algorithms, but uses a different model where the mission completion time is used as cost for minimization, while Manathara et al. used total coalition path length as the cost. In addition, the model introduced in this paper uses agents that were initialized with sufficient resources that met all the targets' resource requirements, would lead to complete prosecution of all the targets. The path generation was introduced in [41].

The remainder of this paper is organized as follows. The coalesced city's remote sensing model is described in Section 2. Section 3 describes the problem statement and algorithms used in this paper. The approach to calculate the Dubins longest path is described in Section 4. Experiments and results are reported in Section 5. The paper is concluded in Section 6.

## 2. The Coalesced City's Remote Sensing Model

### 2.1. Model Formulation

Let us consider a hybrid sensor network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L}, \mathcal{S}, \mathcal{Q})$, where $\mathcal{N}$ is the set of cluster sensor nodes, $\mathcal{L}$ is the set of locations of interest in a given region, $\mathcal{S}$ is the set of the UAV base stations, while $\mathcal{Q}$ is the set of UAVs. Consider the case where each location has to be visited only once by a set of UAVs at time $r(i)$ and a set of UAVs forming a coalition $C_{oal}(i)$ spend an amount of time $t(i)$ to arrive at the location $i$ with a risk $\epsilon(i)$ of arriving earlier than planned or after the expected arrival time. Assume that $\mathcal{D}(\mathcal{G}) = \{\mathcal{D}_n\}$ is the set of the city's remote sensing configurations associated with the hybrid network $\mathcal{G}$.

### 2.2. The City's Remote Sensing Problem

The coalesced city's remote sensing task is based on some rules, which can be expressed in terms of constraints and/or objectives to be met by the coalition of UAVs and resources needed at the specific locations. These rules are described as follows:

- **Visiting time at locations:** An early visit to a location will be penalized, and a late visit to a location is also penalized. Therefore, a visit by a UAV $k(i) \in \mathcal{Q}$, which is part of a coalition $C_{oal}(i)$ at location $i$, will be constrained by:

$$0 \leq r(i) - t(i, k(i)) < T_1, \;\; \forall \;\; k(i) \in C_{oal}(i) i \text{ and} \tag{1}$$

$$T_2 < r(i) - t(i, k(i)) < 0, \;\; \forall \;\; k(i) \in C_{oal}(i), \tag{2}$$

where $t(i, k(i))$ is the time taken by UAV $k(i)$ to arrive at location $i$. $T_1$ and $T_2$ are the allowed early and late thresholds, respectively. The model considers that there is a penalty assigned to the UAV if it arrives early $t(k(i)) < r(i)$ and late $t(k(i)) > r(i)$.
- **Resource requirements at locations:** The sum of resources required by a UAV coalition at location $i$ is constrained by:

$$\sum_{i \in C_{oal}(i)} R^c_{src}(k(i)) \geq R^c_{eq}(i), \;\; \forall \;\; c \in \{1, 2, \ldots, C\}, \tag{3}$$

where $R^c_{src}(k(i))$ is the resource of type $c$ carried by UAV $k$ and $C$ the total number of resources at the location of interest.

- **UAV teams' coalition:** A data collection model is proposed where a coalition of UAVs $C_{oal}(i)$ collects data at a given location $i$. Each network configuration $\mathcal{D}_n$ is defined by a partition of the set of UAVs $\mathcal{Q}$ and made up of a set of UAV coalitions at different locations as expressed by:

$$\bigcup_{i \in \mathcal{L}} C_{oal}(i) = \mathcal{Q} \text{ and}$$
$$\mid C_{oal}(i) \mid = \sum_{k \in \mathcal{Q}} y(i, k(i)),$$

where $\mid C_{oal}(i) \mid$ is the size of a coalition at node $i$ and $y(i, k(i))$ defined by:

$$y(i, k(i)) = \begin{cases} 1, & \text{if } k(i) \in C_{oal}(i), \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

This coalition process is constrained by the fact that each UAV can be either part of only one coalition as expressed by:

$$C_{oal}(i) \cap C_{oal}(j) = \emptyset, \forall\, i, j \in \mathcal{L}, \tag{5}$$

or of more than one coalition if it has enough resources as expressed by:

$$C_{oal}(i) \cap C_{oal}(j) \neq \emptyset, \text{if } R_{src}(k(i)) \geq R_{eq}(i \cup j). \tag{6}$$

- **The city's remote sensing problem** consists of finding a cost-efficient city remote sensing configuration $\mathcal{D}_{cef}$, such that:

$$\begin{aligned} \hat{\tau}_{cst}(\mathcal{D}_{cef}) &= \min_{\mathcal{D}_n \in \mathcal{D}(\mathcal{G})} \tau_{cst}(\mathcal{D}_n), \qquad (7) \\ &= \min_{\mathcal{D}_n \in \mathcal{D}(\mathcal{G})} \sum_{i \in \mathcal{L}} y(i, k(i)) ( \max_{C_{oal}(i) \in \mathcal{D}_n} D_{ist}(k(i))), \,\forall\, k(i) \in C_{oal}, \end{aligned}$$

subject to the constraints:

$$0 \leq r(i) - t(i, k(i)) < T_1, \;\; \forall \;\; k(i) \in C_{oal}(i) \tag{8}$$
$$T_2 < r(i) - t(i, k(i)) < 0, \;\; \forall \;\; k(i) \in C_{oal}(i) \tag{9}$$
$$k \in C_{oal}(i) \cap C_{oal}(j) \text{ if } R_{src}(k(i)) \geq R_{eq}(i \cup j) \tag{10}$$
$$\bigcup_{i \in \mathcal{L}} C_{oal}(i) = \mathcal{D}_{cef} \tag{11}$$
$$\sum_{i \in C_{oal}(i)} R^c_{src}(k(i)) \geq R^c_{eq}(i), \;\; \forall \;\; c \in \{1, 2, ..., C\} \tag{12}$$
$$y(i, k(i)) = \begin{cases} 1, & \text{if } k(i) \in C_{oal}(i), \\ 0, & \qquad \text{otherwise,} \end{cases} \tag{13}$$
$$R_{eq}(i) = \sum_{s \in \mathcal{S}} R_{eq}(k(i), s), \tag{14}$$

where $D_{ist}(k(i), i)$ is the distance covered by UAV $k(i)$ when forming part of coalition $C_{oal}(i)$ to reach location $i$. Note that the constraints (8) and (9) are timing constraints defining respectively the arrival requirements at location $i$ and the city's remote sensing scheduling at the same location. Equations (10)–(13) are resource allocation constraints. Note also that the constraints (10), (11) express the fact that one UAV can participate in different coalitions if it has remaining resources after a first coalition. These constraints could be relaxed to restrict a UAV to participate in only one coalition as expressed by:

$$k \in C_{oal}(i) \cap C_{oal}(j) \text{ if } R_{src}(k(i)) \geq R_{eq}(i \cup j) \tag{15}$$

$$\bigcup_{i \in \mathcal{L}} C_{oal}(i) = \mathcal{D}_{cef} \tag{16}$$

### *2.3. The City's Remote Sensing Process*

At the start of the process, the positions of the locations of interest and the resources on-board the UAVs are provided. The solution presented in Figure 2 consists of two steps,

1. Resource allocation or coalition formation:
   Each UAV may contain three types of resources, i.e., a pollutant sensor, a camera and on-board storage. If all the targets or locations of interest have been visited, then execution details, such as the coalition and mission times, are reported, otherwise proceed and find any of the remaining locations of interest. If the combined resources of the available UAVs that are not assigned to coalitions fulfil the resource requirement of the location of interest, then proceed to form a coalition of minimal distance each time, by including from the available UAVs the one closest to the location of interest in the potential coalition until the resource requirement of the location of interest is met.
2. Path planning:
   Once a coalition is formed, the radii of the coalition members are increased to ensure that their Dubins distances are all equal to that of the UAV furthest from the location of interest under consideration. The coalition members proceed to fly to the location of interest. After collecting the sensor data at the location of interest, the UAVs become available to form further coalitions should they have any resources remaining.

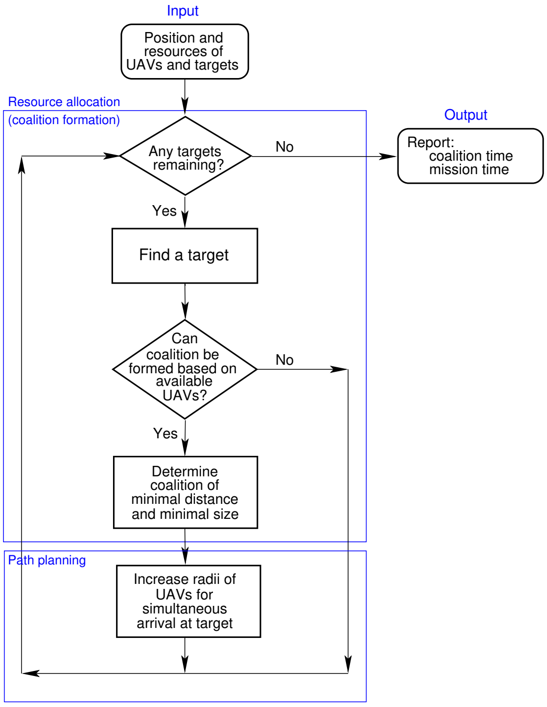

Input
Position and resources of UAVs and targets
Resource allocation (coalition formation)
Any targets remaining?
No
Output
Report: coalition time mission time
Yes
Find a target
Can coalition be formed based on available UAVs?
No
Yes
Determine coalition of minimal distance and minimal size
Path planning
Increase radii of UAVs for simultaneous arrival at target

**Figure 2.** Diagram depicting solution steps.

## 3. Problem Statement and Algorithms

A number of fixed-wing UAVs are required to collect different types of environment parameters from various locations. Each location may contain different amounts of each of the different types of environment parameters. Each UAV is capable of storing each type of environment parameter, but in different quantities. For example, a UAV may contain a sensor to detect the levels of pollutants such as $CO_2$ or $NH_3$, a camera with picture quality ranging from 8–16 megapixels (MP) and on-board storage ranging from 500 megabytes (MB) to two gigabytes (GB). See the illustration in Figure 1. The UAVs do not know the positions of the locations of interest and fly in search of it and can only detect a location's position once it is within the UAVs' sensor range. All UAVs can communicate with each other. Once the position of a location of interest is detected, a single-member coalition or multiple-member coalition can potentially be formed to collect the environment data using Dubins longest path. After collecting the data at a location, the UAVs that formed part of the coalition are released and continue to search for positions of the remaining locations of interest and may participate in further coalitions if they have any data storage space available. The objective of coalition formation is to complete a mission in minimum time, by forming a coalition for each location of interest that (a) reaches the position of a location in minimum time, (b) uses a minimum number of UAVs and (c) collects the environment data simultaneously.

In [40], the combined resources of all UAVs may not be sufficient to prosecute all the targets. This will be the case if for each resource type, the sum of the quantities of the resource under consideration of all UAVs is not equal to, or does not exceed, the sum of the quantities of the same type of resource of all the targets. No test is performed initially to check whether the combined resources of all UAVs are sufficient to prosecute all the targets. Instead of initiating a mission that may not end in success, one can establish if the combined storage capacity of all UAVs in a mission is adequate to accommodate the environment data of all locations of interest before commencing a mission. For the data used in the experiments reported in this paper, it was ensured that the combined storage capacity of all the UAVs was sufficient to accommodate the environment data at all locations of interest.

Further, in Manathara et al.'s paper, the positions of the fixed position targets were also not known in the polynomial time coalition formation algorithm (PTCFA) and optimal coalition formation algorithm (OCFA). If the targets are dynamic and moving all the time, one could understand the need for searching for them. However, if the targets are fixed and static, then the actual positions of the targets can be communicated to the UAVs at the start of the mission. This is a more realistic representation of problems that involve targets at fixed positions. The PTCFA and OCFA solutions proposed by Manathara et al., where targets are searched for, can easily be remodeled as a solution where the positions of the targets are known to the UAVs at the start of the execution of the algorithm by ensuring that the sensor range is large enough to cover the entire fly zone. The length of the diagonal of the square search area with each side 1000 m long is 1414 m. Fitting the UAVs with sensors that have a sensor range of 1500 m will ensure that the entire search area is covered and that all targets or the positions of environment data locations are detectable by the UAV sensors when the coalition formation algorithm is executed.

In [40], the mission completion time is minimized by selecting coalition paths of minimal length to each target using a coalition of minimal size. Manathara et al. described the Dubins longest path using three angles, $\theta_1$, $\theta_2$, $\theta_3$, and the radius, as indicated in Section 4. The $\theta$'s are computed using the initial position of the UAV, the position of the target and the initial direction of flight of the UAV. Finding accurate values for $\theta_3$ may pose a problem as indicated in Section 4, and a different and more robust approach is adopted in this paper to calculate the Dubins longest path based on the following four attributes, (1) the initial position of the UAV, (2) the size of the angle traversed during the circular path, (3) the exit point where the UAV leaves the circular path and (4) the incoming angle of the UAV when approaching the target.

Coalition formation is considered using three approaches as described below.

The Dubins longest path was included in three algorithms developed by Manathara et al. as described in [40]. The algorithms are:

1. polynomial time coalition formation algorithm (PTCFA),
2. optimal coalition formation algorithm (OCFA) and
3. particle swarm optimization (PSO).

In all three algorithms, it is assumed that all UAVs fly at a constant speed and at a constant altitude. A short description of each algorithm appears below.

### 3.1. PTCFA

The PTCFA consists of two parts. First, each UAV is in search of a target and is equipped with a sensor with limited range capability. As soon as a target is detected, its position and resource requirements are communicated to all UAVs. The UAV that detects the target is referred to as the coalition leader. All UAVs that are not in a coalition and that contain any of the resources required by the target calculate their estimated time of arrival (ETA) or cost and communicate this information together with the amount of resources on board the UAV to the coalition leader. The coalition leader sorts the ETAs received from all responding UAVs in ascending order and then adds the UAV closest to the target to the potential coalition and accumulates its resources. If the accumulated resources do not meet the target's resource requirements, then the next closest UAV is also included in the coalition and its resources accumulated, as well. This process of adding a UAV with the next smallest ETA to the potential coalition continues until the resources accrued satisfy the resource requirements of the target. In the second part of the PTCFA, the coalition leader sorts the ETAs of the UAVs in the potential coalition in ascending order. Again starting with the UAV closest to the target, the coalition leader removes the resources of this UAV from the resources accumulated. If the remaining accumulated resources after removal still satisfy the target's resource requirements, then the UAV under consideration is removed from the potential coalition; otherwise, its resources are added back to the accumulated resources. The UAV removed is redundant since its resources are not required for prosecution. The removal from the coalition of the next closest UAV is then considered. This process ends after all UAVs in the coalition have been considered for removal. The coalition leader then informs all remaining coalition members of (a) their inclusion in the coalition and (b) the ETA of the UAV furthest from the target. Coalition members then calculate revised radii that increase their ETAs at the target to match the ETA of the UAV furthest from the target.

### 3.2. OCFA

The first part of the OCFA is the same as the first part of the PTCFA. In the second part, the size of the coalition is minimized using mixed integer linear programming such as MATLAB's bintprog. The coalition leader then informs all coalition members of (a) their inclusion in the coalition and (b) the ETA at the target of the UAV furthest from the target. Coalition members then increase their radii appropriately so that their distances to the target match that of the UAV furthest from the target.

### 3.3. PSO

Particle swarm optimization (PSO) has been used in [40] to select an optimal set of UAVs for a single known target, given a set of UAVs and their resources. PSO is a nature-based stochastic optimization algorithm that emulates the swarm behaviors of bird flocks [42]. PSO searches for an optimum solution by merely drawing the position of each particle in the swarm toward its own historical best position and toward the position of the historical best particle in a defined neighborhood [43]. The position and velocity of each particle are updated over time. Each particle in the swarm is $N \times M$-dimensional, where $M$ denotes the number of targets and $N$ the number of UAVs. Each particle is a potential solution to the optimization problem. The position vector of a particle is constructed as follows. For each UAV, there is an $M$-dimensional vector where each element stores the preferred target to be visited by the UAV, e.g., vector (5,3,2,1) indicates the preference of the UAV to visit targets in the order 5, 3, 2 and 1. The $N$ vectors, each of dimension $M$, of the UAVs are concatenated to form an $N \times M$ position vector. The position vector of each particle contains the preferences of all $N$ UAVs in visiting the various targets. Each value in the position vector of a particle indicates the target to be visited and potentially prosecuted by a specific UAV, e.g., if element (2,3) in the position vector has a value of four, then it indicates that UAV Number 2 prefers Target Number 4 to be visited during the third sequence. To check whether a particle can successfully prosecute a target, say Target 4, during Sequence 2, proceed by identifying all UAVs that containing a value of four as their second vector element. All the UAVs identified form a potential coalition to prosecute Target 4. Next, accumulate the resources of all the UAVs in the potential coalition, i.e., the UAVs that prefer Target 4. If the combined resources of all the UAVs in the potential coalition satisfy the resource requirements of Target 4, then a coalition is possible, and the coalition time is calculated as the time taken by the UAV with the largest ETA to reach the target. If all targets can be prosecuted by a particle, then the sum of the coalition distances of all coalitions will be returned as the cost of the prosecution by the particle. A fitness function evaluates the cost of a particle. The cost of a successful mission is simply the sum of the coalition time required to prosecute each target. If all targets are not prosecuted by a particle, then a large cost, say infinity, is assigned as the particle's fitness value. For more details of the fitness function, refer to [40]. To kick start the process, the position vector of the swarm is initialized to random values in the range [0, $M$], where zero indicates a search and a non-zero integer of the prospective target to be prosecuted; targets being numbered from 1–$M$. The PSO process ensures that the position vector yielding the best fitness and thus lowest coalition time is returned as the minimum or optimal value.

In the experiments reported in this paper, the resources referred to in Manathara et al. were replaced with environment data and the targets with locations of interest from whence environment data must be collected.

The next section discusses the Dubins longest path.

## 4. Calculating the Dubins Longest Path

Dubins curves can be used to calculate the shortest path to a target for a fixed-wing UAV. A Dubins curve consists of all combinations of arcs of minimal turning radius and a straight line segment that joins the two arcs. Six possible configurations were considered, namely LSL, LSR, RSL, RSR, RLR and LRL, where L and R represent a left and right turn, respectively, and S a straight line segment. In many search missions, fixed-wing UAVs are required only to fly a circular path followed by a straight line path to reach a target. No second arc is required. Hence, the original Dubins curves that consist of three components can be reduced to a curve with two components, in which case only two possible cases need to be considered, i.e., LS and RS.

Given a drone's current position and a target, two types of Dubins paths can be traversed to reach the target, i.e., the Dubins shortest path (DSP) and Dubins longest path (DLP) (Figure 3a). A target may lie either to the left or to the right of the straight line path of a UAV. In the case of DSP, the UAV performs a turn in the same direction as the direction of the target relative to the UAV's straight line path of flight. In the case of DLP, the UAV turns in the opposite direction of the direction of the target relative to a UAV's position.

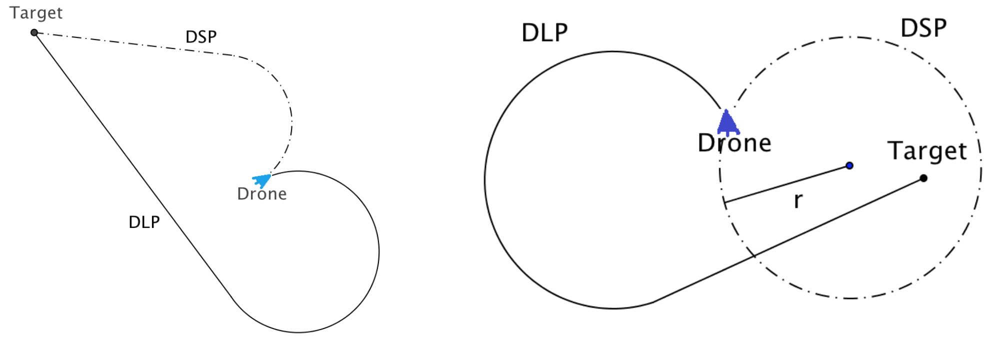
When the distance between the UAV and the target is smaller than the minimum turning radius of the UAV, then the Dubins shortest path cannot reach the target, as shown in Figure 3b. In this case, the UAV will simply circle around the target, never reaching it.

With the Dubins longest path, this problem does not exist, since the target is always reachable, irrespective of how close a target is to the UAV, as shown in Figure 3b.

To calculate the distance of the Dubins longest path, the following two cases are considered,

1. the target lies to the right of the straight line path of the UAV or
2. the target lies to the left of the straight line path of the UAV.

In Case (1), the UAV has to turn left or counterclockwise, while in Case (2) the UAV has to turn right or clockwise to reach the target.



(**a**) (**b**)

**Figure 3.** (**a**) Dubins shortest and longest paths (DSP and DLP, respectively) where the target lies to the left of the path of the drone; (**b**) DSP and DLP where the target lies to the right of the path of the drone.

In 'Multiple UAV Coalitions for a Search and Prosecute Mission' by Manathara et al., the estimated time of arrival of a UAV reaching its target based on the Dubins longest path for the two cases was defined in terms of three angles $\theta_1$, $\theta_2$ and $\theta_3$ (see Figure 4), where $\theta_1$ denotes the angle between the line that joins the center of the circle with the target and the line that joins the center of the circle with the exit point of the UAV from the arc, $\theta_2$ denotes the angle between the line that joins the initial position of the UAV with the center of the circle and the positive $x$-axis, measured in a positive direction, i.e., anti-clockwise, while $\theta_3$ denotes the angle between the line that joins the center of the circle with the target and the positive $x$-axis measured in a positive direction.

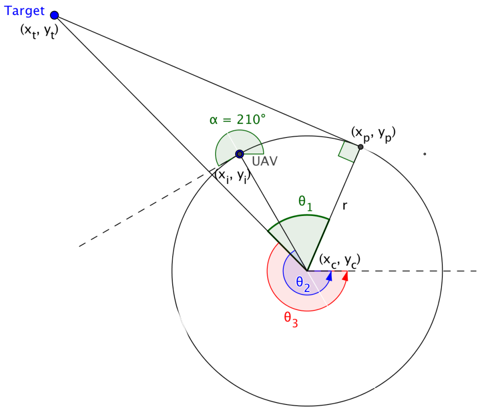


**Figure 4.** Dubins longest path.

The angles are given by:

$$\theta_1 = \arccos\left(\frac{\sqrt{(x_t - x_c)^2 + (y_t - y_c)^2}}{r}\right), \tag{17}$$

$$\theta_2 = \frac{\pi}{2} - \alpha, \quad \text{and} \tag{18}$$

$$\theta_3 = \arctan\left(\frac{y_t - y_c}{x_t - x_c}\right). \tag{19}$$

In Case (1), the estimated time of arrival (ETA) was defined as,

$$D = \frac{r(2\pi - (\theta_1 - (\theta_2 - \theta_3))) + r\tan\theta_2}{v_i}, \tag{20}$$

and in Case (2),

$$D = \frac{r(2\pi - (\theta_1 - (\pi - \theta_3) - \theta_2)) + r\tan\theta_2}{v_i}, \tag{21}$$

where $v_i$ is the ground speed, $(x_c, y_c)$ is the center of the arc traversed by the UAV, $(x_t, y_t)$ is the position of the target and $\alpha$ the angle of direction of flight of the UAV measured from the positive $x$-axis.

Calculating the Dubins longest path using Equations (10) and (11) may pose a problem in certain cases, as reflected in Figure 5.

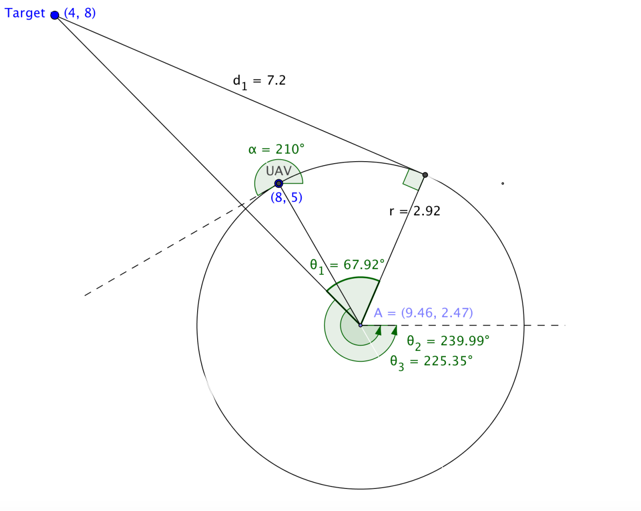


**Figure 5.** Calculating $\theta$'s based on Manathara et al.

Substituting the radius of the arc, the coordinates of the target's position, the center of the arc and the angle of the flight of the UAV in Equations (7)–(18) yield values for $\theta_1$, $\theta_2$ and $\theta_3$ as 67.92°, $-120.0°$ and $-45.65°$, respectively, while values computed using the mathematical tool Geogebra for $\theta_1$, $\theta_2$ and $\theta_3$ are 67.92°, 239.99° and 225.35°, respectively. This discrepancy resulted in an alternative approach to be explored to determine the Dubins longest path.

An alternative approach to calculate the Dubins longest path is proposed next. If the UAV's initial position, $(x_i, y_i)$, the angle of flight of the UAV, $\alpha$, the target's position $(x_t, y_t)$ and the minimum turning radius of the UAV, $r_{min}$, are provided, then the distance of the circular path and the straight line path of Dubins longest path can be calculated based on the following data:

- center of the circular path formed when the UAV turns,
- the exit point of the UAV from the circular path,
- the central angle traversed by the UAV while turning and
- the angle of the direction of flight of the UAV when approaching the target along the straight line path.

The distance of the Dubins longest path is then simply computed as the distance traversed along the arc plus the distance covered by the straight line path that joins the exit point on the arc with the target, as shown in Equation (38). In the next section, each of the data values referred to above will be calculated.

### *4.1. Calculating the Coordinates of the Center of the Circle*

The center of the circle is calculated for each of the two cases referred to above.

If the target lies to the right of the straight line path of the UAV, then the center of the circle, $(x_c, y_c)$ is calculated as follows,

In triangle CDI , in Figure 6,

$$\begin{aligned} \frac{x_i - x_c}{r} &= \sin\alpha_2 \text{ and} \\ x_c &= x_i - r\cdot\sin\alpha_2, \end{aligned} \tag{22}$$

and,

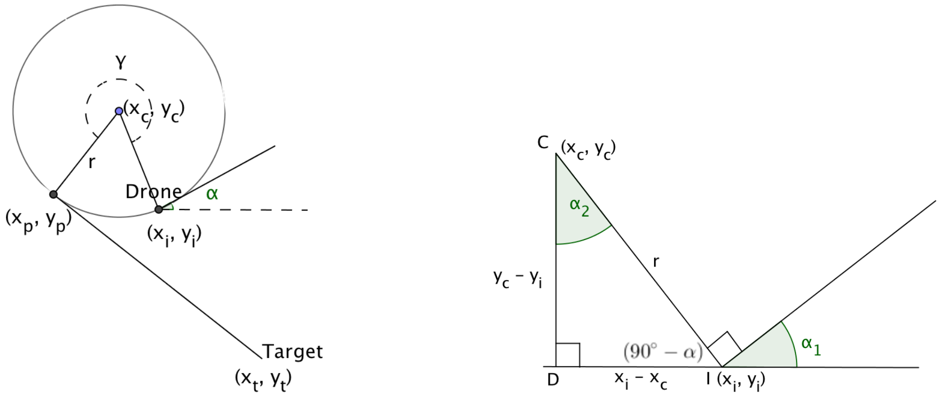
$$\begin{aligned} \frac{y_c - y_i}{r} &= \cos\alpha_2 \text{ and} \\ y_c &= y_i + r\cdot\cos\alpha_2. \end{aligned} \tag{23}$$

When the target lies to the left of the straight line path of the UAV, then it can be shown that,

$$x_c = x_i + r\cdot\sin\alpha \text{ and} \tag{24}$$

$$y_c = y_i - r\cdot\cos\alpha, \tag{25}$$

where $r$ is the radius of the arc and $\alpha$ denotes the angle of the direction of the UAVs initial flight path.



(**a**) (**b**)

**Figure 6.** (**a**) Calculating the center of the circle when the target lies to the right of the path of the drone; (**b**) Calculating the center of the circle given the initial position, angle of the drone and the radius.

### *4.2. Calculating the UAV's Exit Point on the Arc*

Let $m_k$ denote the gradient of line L$_k$ in Figure 10, then:

$$m_2 = \tan\ \alpha_2. \tag{26}$$

If line L$_2$ is not parallel to the $y$-axis, then its equation is,

$$y = m_2\cdot x + c_2. \tag{27}$$

Since $(x_t, y_t)$ lies on line L$_2$, $(x_t, y_t)$ can be substituted in Equation (27) to calculate the constant $c_2$, i.e.,

$$c_2 \quad = \quad y_t - m_2 \cdot x_t. \tag{28}$$

Equation (27) can now be written as,

$$y \quad = \quad m_2 \cdot x + (y_t - m_2 \cdot x_t). \tag{29}$$

Since line L$_3$ is perpendicular to line L$_2$, this implies:

$$\begin{aligned} m_3 \cdot m_2 &= -1, \\ m_3 &= \frac{-1}{m_2} = cot\ \alpha_2, \\ &= \frac{y - y_c}{x - x_c}, \\ y - y_c &= m_3 \cdot (x - x_c). \end{aligned} \tag{30}$$

The equation of L$_3$ is,

$$y \quad = \quad y_c + m_3 \cdot (x - x_c). \tag{31}$$

The point of tangency is the intersection of lines L$_2$ and L$_3$ in Figure 10 and is calculated as,

$$(x_p, y_p) \quad = \quad (\frac{y_c - m_3 \cdot x_c - c_2}{m_2 - m_3}, m_2 \cdot x_p + c_2). \tag{32}$$

If line L$_2$ is parallel to the $y$-axis, then line L$_3$ (that is perpendicular to L$_2$) is parallel to the $x$-axis; their corresponding equations are,

$$\begin{aligned} \mathrm{L}_2: &\quad x = x_t, \\ \mathrm{L}_3: &\quad y = y_c \end{aligned}$$

and the point of tangency is,

$$(x_p, y_p) \quad = \quad (x_t, y_c). \tag{33}$$

Similarly, if line L$_2$ is parallel to the $x$-axis, then line L$_3$ is parallel to the $y$-axis; the corresponding equations are,

$$\begin{aligned} \mathrm{L}_2: &\quad y = y_t, \\ \mathrm{L}_3: &\quad x = x_c \end{aligned}$$

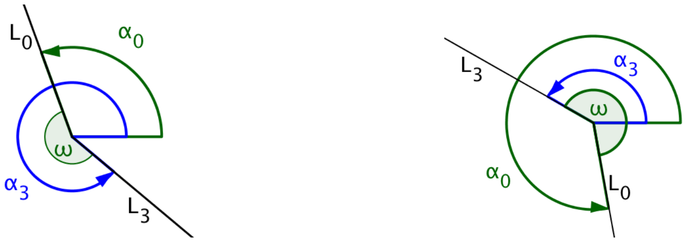
and the point of tangency is,

$$(x_p, y_p) \quad = \quad (x_c, y_t). \tag{34}$$

The exit point of the UAV from the circular path is given by $(x_p, y_p)$.

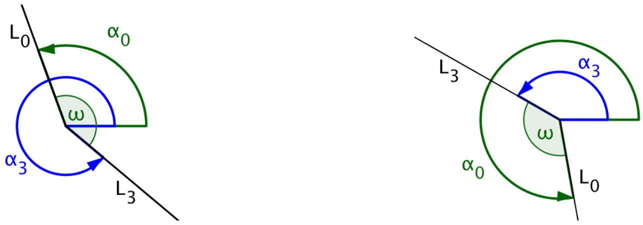
### *4.3. Calculating the Angle Traversed by the UAV While Turning*

If L$_0$ denotes the line that joins the UAV's initial position $(x_i, y_i)$ with the center of the circle $(x_c, y_c)$ and L$_3$, the line that joins the exit position of the UAV on the arc $(x_p, y_p)$ with the center of the circle, then the angle traversed by the UAV is the angle measured from line L$_0$ to line L$_3$ in the same direction as that of the UAV's circular direction of flight. In Figure 7, $\alpha_0$ and $\alpha_3$ denote the angles formed between the $x$-axis and lines L$_0$ and L$_3$, respectively. The size of the angle traversed by the UAV for the circular path is the angle $\omega$ measured from L$_0$ to L$_3$, as shown in the diagrams in Figures 7 and 8.



(**a**) (**b**)

**Figure 7.** (**a**) UAV flies counterclockwise where $\alpha_0 < \alpha_3$; (**b**) UAV flies counterclockwise where $\alpha_0 > \alpha_3$.



(**a**) (**b**)

**Figure 8.** (**a**) UAV flies clockwise where $\alpha_0 < \alpha_3$; (**b**) UAV flies clockwise where $\alpha_0 > \alpha_3$.

The central angle is calculated as indicated in Algorithm 1.

**Algorithm 1:** Calculating the central angle, $\omega$.

```
if (UAV turns left) then
  if ($\alpha_3 > \alpha_0$) then
    $\omega \leftarrow \alpha_3 - \alpha_0$
  else
    $\omega \leftarrow 360° + (\alpha_3 - \alpha_0)$
  end if
else
  if (UAV turns right) then
    if ($\alpha_0 > \alpha_3$) then
      $\omega \leftarrow \alpha_0 - \alpha_3$
    else
      $\omega \leftarrow 360° + (\alpha_0 - \alpha_3)$
    end if
  else
    $\omega \leftarrow 0°$
  end if
end if
```

### 4.4. Calculating the Incoming Angle of the UAV

In the case where the target lies to the right of the straight line path of the UAV and the UAV has to turn anti-clockwise as reflected in Figure 9, let the new angle of the direction of the flight of the UAV be denoted by $\alpha_2$. If $\alpha_1$ is the angle between the extended hypotenuse and the positive $x$-axis and $\beta$ is the angle between the hypotenuse extended and the tangent to the circle at $(x_p, y_p)$ extended, then $\alpha_1$ and $\beta$ can be calculated as follows,

$$\alpha_1 \quad = \quad \arctan 2(y_t - y_c; x_t - x_c), \tag{35}$$

$$\beta \quad = \quad \arcsin(\frac{r}{\sqrt{(x_t - x_c)^2 + ((y_t - x_c)^2}}), \tag{36}$$

where $\arctan 2(\cdot, \cdot)$ returns the angle measured between the positive $x$-axis and the line that joins the two points $(x_t, y_t)$ and $(x_c, y_c)$. If the resulting value is negative, then simply add 360° to yield the angle measured counterclockwise from the positive $x$-axis.

In Figures 9 and 10, $\mathrm{L}_1$ denotes the line that joins the center of the circle with the target, while $\mathrm{L}_2$ denotes the line that joins the UAVs exit point, after turning, with the target. $\mathrm{L}_3$ denotes the line that joins the center of the circle with the exit point.

In Figure 9, where the target lies to the right of the straight line path of the UAV resulting in the UAV turning left, the new incoming angle of the UAV is calculated as,

$$\alpha_2 \quad = \quad \alpha_1 + \beta. \tag{37}$$

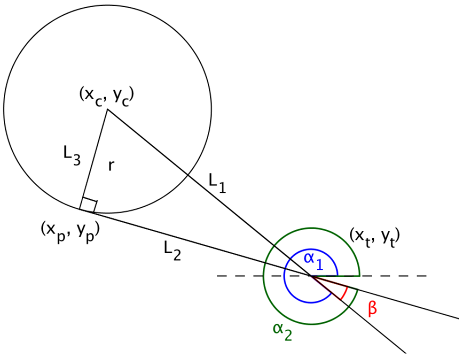


**Figure 9.** Target lies to the right of the UAV's straight line path.

In the case where the target lies to the left of the straight line path of the UAV, as shown in Figure 10, the new angle of the UAV, $\alpha_2$, is calculated as,

$$\alpha_2 \quad = \quad \alpha_1 - \beta. \tag{38}$$

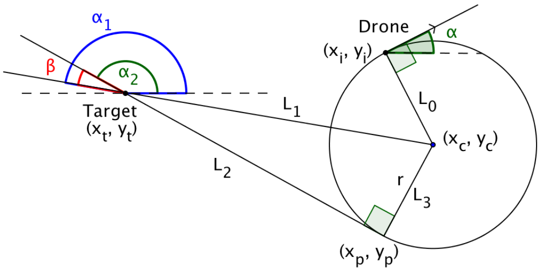


**Figure 10.** Target lies to the left of the UAV's straight line path.

Finally, the distance of the Dubins longest path, $d$, is the distance measured along the arc from the UAV's initial position to the exit point on the circular path, plus the distance of the straight line path from the exit point to the target's position, i.e.,

$$d \quad = \quad \omega \cdot r_{min} + \sqrt{(x_t - x_p)^2 + (y_t - y_p)^2}. \tag{39}$$

This concludes the discussion on the calculation of the Dubins longest path based on elementary geometry.

## 5. Experiments and Results

For each of the OCFA, PTCFA and PSO algorithms, 100 simulations were performed on a square search area with sides of length 1000 m. To restrict the UAVs to the search area and to allow sufficient space for turning around within the search area, a strip of width 100 m around the boundary of the search area was deemed out of place for placing the data collection locations. Locations were randomly placed at positions within the 800 m × 800 m area inside the search area. The UAVs were restricted to flying inside this smaller area. The UAVs were also initially placed at random positions inside the fly zone with an angle of direction of flight generated randomly. All UAVs had a minimum turning radius of 50 m. A UAV that flies outside the fly zone will immediately turn around and can comfortably return inside the 100-m strip without leaving the search space. All UAVs were flying at a constant speed of 10 m/s. It is assumed that all UAVs fly at the same constant speed at the same altitude during the entire mission and that each UAV can store three different types of environment data.

The quantity for each type of environment data at each location of interest was generated as a random integer value in the range 0–3. The algorithmic parameters were set to the same values as in Manathara et al.'s paper, except that the storage capacity for each type of environment data of the UAVs was randomly generated with values ranging from 0 to the (number-of-locations-of-interest/2) and stored only if, for each type of environment data, the total sum of storage capacity for that specific type of data of all UAVs was equal to or greater than the total sum of quantities of the specific environment data of all the locations. This was done to ensure that the combined storage capacity of all the UAVs was sufficient to store all the environment data to be collected at the various locations.

In the simulations reported in this paper, the mission completion time, and not the coalition path length, was used as the cost. Simulations were performed for each of 5, 10, 15 and 20 locations of interest or targets with varying numbers of UAVs ranging from 5–20. The following values were recorded: mission completion time, i.e., time taken from the start of a mission until all the environment data of the last location were collected, total coalition time, i.e., the sum of the coalition time of all locations, and simulation time, i.e., the time taken to complete a simulation from the time when the UAVs were set in motion until the time when all the environment data of the last location of interest were collected.

### *5.1. Comparing Simulation Times of Algorithms Whose Cost Is Based on Either Coalition Time or on Mission Time*

In [40], Manathara et al. used coalition time as the cost for minimization in their algorithms. Results reported in this section are based on the algorithms using mission time instead, as the cost for minimization. However, since coalition path length was key for the inclusion of a UAV in a potential coalition in both PTCFA and OCFA, no differences were observed in the results when either coalition time or mission time was considered as the cost in Table 1. However, a difference was observed between the simulation times of PTCFA and OCFA for each of the two types of cost approaches. Of the 16 experiments conducted in the case of OCFA, only two cases with mission time as the cost produced smaller simulation times. PTCFA with mission time as the cost achieved smaller simulation times than PTCFA with coalition time as the cost in 11 of the 16 cases. PSO with mission time as the cost reflected smaller simulation times than PSO with coalition time as the cost in all cases, bearing in mind that PSO takes the longest to execute of all three algorithms.

**Table 1.** Simulation times in milliseconds of the optimal coalition formation algorithm (OCFA), the polynomial time coalition formation algorithm (PTCFA) and particle swarm optimization (PSO).

| | | OCFA | | PTCFA | | PSO | |
|---|---|---|---|---|---|---|---|
| **No of Targets** | **No of UAVs** | **Based on Coalition Time** | **Based on Mission Time** | **Based on Coalition Time** | **Based on Mission Time** | **Based on Coalition Time** | **Based on Mission Time** |
| 5 | 5 | 19.17 | 19.95 | 12.62 | 8.61 | 22,414.8 | 21,570.1 |
| 5 | 10 | 12.98 | 15.67 | 8.30 | 4.89 | 34,006.9 | 32,734.6 |
| 5 | 15 | 9.67 | 10.76 | 7.39 | 5.14 | 43,106.9 | 405,858.5 |
| 5 | 20 | 8.87 | 9.53 | 7.21 | 19.25 | 52,920.2 | 49,756.5 |
| 10 | 5 | 16.36 | 17.46 | 13.34 | 19.00 | 24,591.3 | 23,793.2 |
| 10 | 10 | 16.55 | 16.33 | 12.45 | 16.81 | 40,502.0 | 37,648.8 |
| 10 | 15 | 14.43 | 15.06 | 11.38 | 10.37 | 49,174.7 | 47,941.2 |
| 10 | 20 | 13.97 | 15.34 | 11.41 | 9.54 | 59,669.6 | 57,586.5 |
| 15 | 5 | 24.89 | 28.07 | 24.60 | 20.63 | 28,950.8 | 24,245.7 |
| 15 | 10 | 19.66 | 21.77 | 21.66 | 16.95 | 59,202.1 | 45,361.7 |
| 15 | 15 | 22.98 | 25.71 | 19.68 | 21.32 | 71,315.7 | 55,908.2 |
| 15 | 20 | 22.32 | 29.67 | 16.88 | 22.53 | 88,065.8 | 66,963.6 |
| 20 | 5 | 30.23 | 33.67 | 27.69 | 27.07 | 31,280.2 | 20,218.5 |
| 20 | 10 | 27.10 | 31.03 | 26.23 | 26.19 | 50,740.8 | 50,217.7 |
| 20 | 15 | 25.47 | 28.18 | 24.31 | 22.56 | 62,134.9 | 60,292.0 |
| 20 | 20 | 24.43 | 24.29 | 19.63 | 18.81 | 72,133.8 | 71,825.6 |

### *5.2. Effect of Increased Sensor Range*

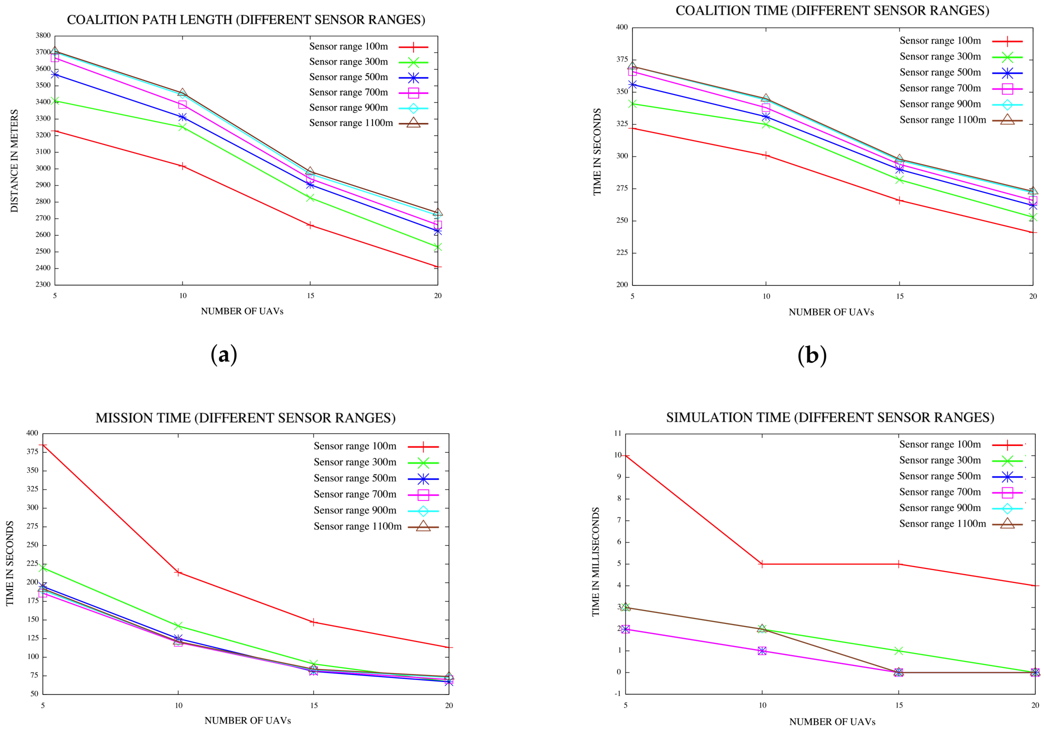
In [40], Manathara et al. compared the performances of PTCFA and OCFA with PSO. In their paper, the positions of the targets were not known to the UAVs for both PTCFA and OCFA, while the positions of the targets were known to the UAVs in PSO. In such a comparison, PSO has an unfair advantage over both PTCFA and OCFA, since the latter two algorithms must first search and find a target before they can form a coalition. Searching for a target consumes additional time, while PSO does not spend any time in searching for the targets and can immediately proceed with coalition formation, if the target's resource requirements are met. A large sensor range that covers the entire search area enables a UAV to detect the positions of all the targets or locations of interest. In the second set of experiments, the sensor range of the UAVs was gradually increased from 100 m–1100 m in increments of 200 m; noting that locations are placed inside a square with each side 800 m long. A sensor range of 1100 m effectively covers a total distance of 2 × radius (= 2200 m), which is the diameter of the circle of the area covered by a sensor with a sensor range of 1100 m. A distance of 1100 m should cover almost the full distance of 1131 m between any two corners positioned diagonally opposite each other in the fly zone. A sensor range of 1100 m should enable the sensor of a UAV to detect all the locations of interest. A hundred simulations were performed with each sensor range value. Results are reported in Figure 11a–d.

Plots in Figure 11a indicate that coalition path length increases with an increase in sensor range. It should be noted that coalition path length refers only to the path length of the largest Dubins longest path to reach a target. The coalition path length does not include the distance covered by the UAV while searching for a target. With a small sensor range, coalition formation can only be initiated when a target is detected within this smaller range; hence, coalition formation is only triggered when the distance to a target is within the distance covered by the smaller sensor range. In the case of a larger sensor range, coalition formation is initiated similarly only when the distance between a UAV and a target falls within the larger sensor range. With a larger sensor range, coalition formation can be initiated much sooner than the case with a smaller sensor range. In the case of the larger sensor range, the coalition leader that detects the target might be the closest to the target with all other UAVs that join the coalition to be further from the target. Hence, generally, a larger sensor range will lead to larger coalition path lengths.

Since all UAVs fly at a constant speed and coalition time is calculated as coalition path length divided by the speed of the UAV, one would expect the coalition time to also increase with an increase in sensor range, as reflected in Figure 11b.

Mission time consists of time spent on searching for the locations of interest or targets and time spent on executing a coalition. During the execution of a coalition, the UAVs traverse the Dubins longest paths to reach the target, with the radii of the participating UAVs increased to ensure that all UAVs reach the target at the same time. With a smaller sensor range, targets are only detected once they are within this smaller sensor range of a UAV. Often, UAVs could find themselves flying away from a target that is just outside the sensor range and move effectively further away form a target while searching for a target. This is time wasted by the UAV that could have better been utilized by executing a coalition. With a larger sensor range, less time is spent and wasted on searching for targets, and coalitions are formed earlier, which leads to reduced mission times. The plots in Figure 11c confirm that mission time decreases with an increase in sensor range and that mission time decreases with an increase in the number of UAVs.

Simulation time decreases with an increase in sensor range, as reflected in Figure 11d. A sensor range of 500 m covers a circle with a corresponding diameter of 1000 m, which covers a large part of the square fly zone with each side length being 800 m. The plots in Figure 11d indicate that sensor ranges 500 m and 700 m produce identical simulation times, which on their face, implies that PTCFA did not benefit at all from the larger sensor range. However, Figure 11c puts this misleading conclusion to rest, since a sensor range of 700 m produced smaller mission times than a sensor range of 500 m. One can conclude that the benefits and losses with respect to simulation time in the case of 500 m and 700 m sensor ranges offset one another, resulting in similar simulation times. Similar simulation times are reported also for sensor ranges of 500 m, 700 m and 900 m in the case of 15 and 20 UAVs.



(a)

(b)

(c)

(d)

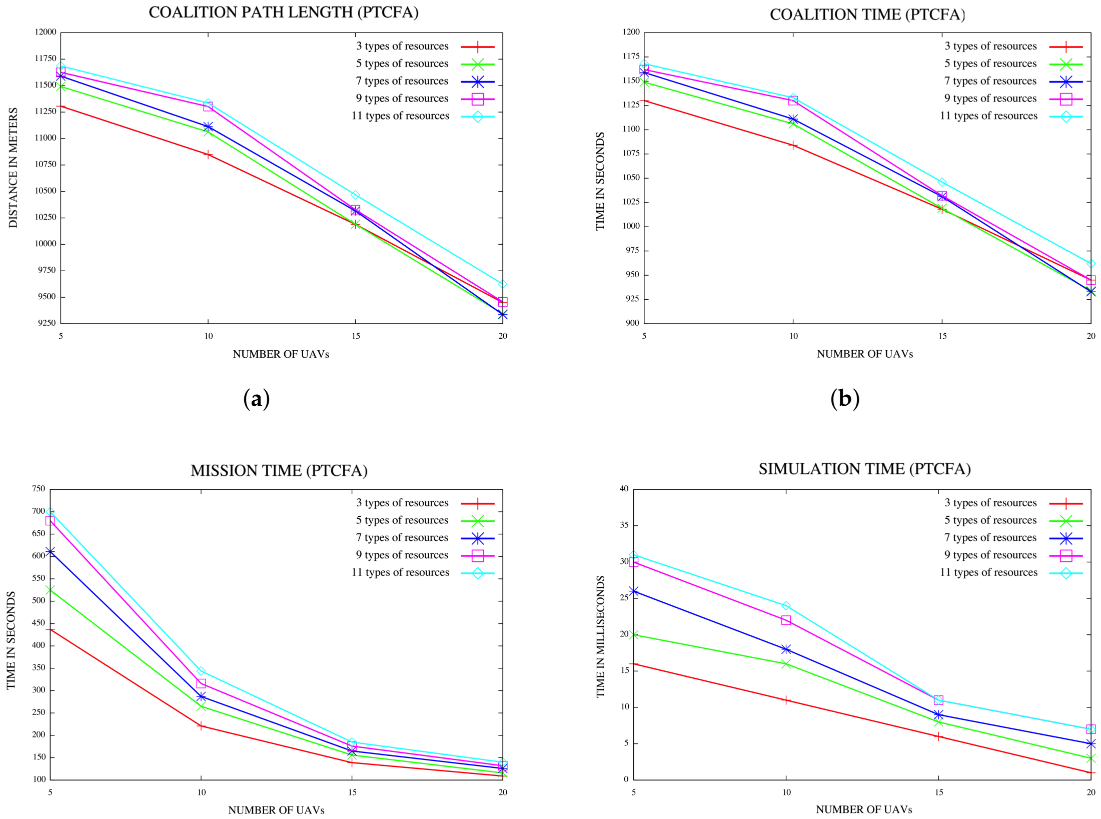
**Figure 11.** (**a**) Coalition path lengths for various sensor ranges; (**b**) coalition times for various sensor ranges; (**c**) mission time for various sensor ranges; (**d**) simulation time for various sensor ranges.

### *5.3. Effect of Increased Number of Types of Resources or Environment Data*

Another set of experiments was conducted to investigate the effect of an increased number of types of resources or environment data on the mission, w.r.t. coalition time, mission completion time and simulation time. The number of types of resources was increased from 3–11 in increments of two. Configurations comprised of 15 locations of interest or targets and UAVs varying from 5–20 in increments of five were considered. For each configuration, a hundred simulations were performed, and the average for each of the characteristics referred to above were recorded. Results are reported in Figure 12a–d. The plots in Figure 12a indicate that the coalition path length increases with an increase in the number of types of resources or environment data. A similar trend is noticed with the coalition time as reflected in Figure 12b. For each configuration of the number of environment data or resources, the coalition time decreases with an increase in the number of UAVs in both Figure 12a,b.

The plots in Figure 12c indicate that the mission time increases with an increase in the number of types of resources or environment data. For a fixed number of types of resources or environment data, the mission time decreases with an increase in the number of UAVs.

Graphs in Figure 12d indicate that simulation time also increases with an increase in the number of types of resources or environment data.



(**a**)

(**b**)

(**c**)

(**d**)

**Figure 12.** (**a**) Coalition path lengths for various numbers of resources for 15 locations; (**b**) coalition times for various numbers of types of resources for 15 locations; (**c**) mission time for various numbers of types of resources for 15 locations; (**d**) simulation time for various numbers of types of resources for 15 locations.

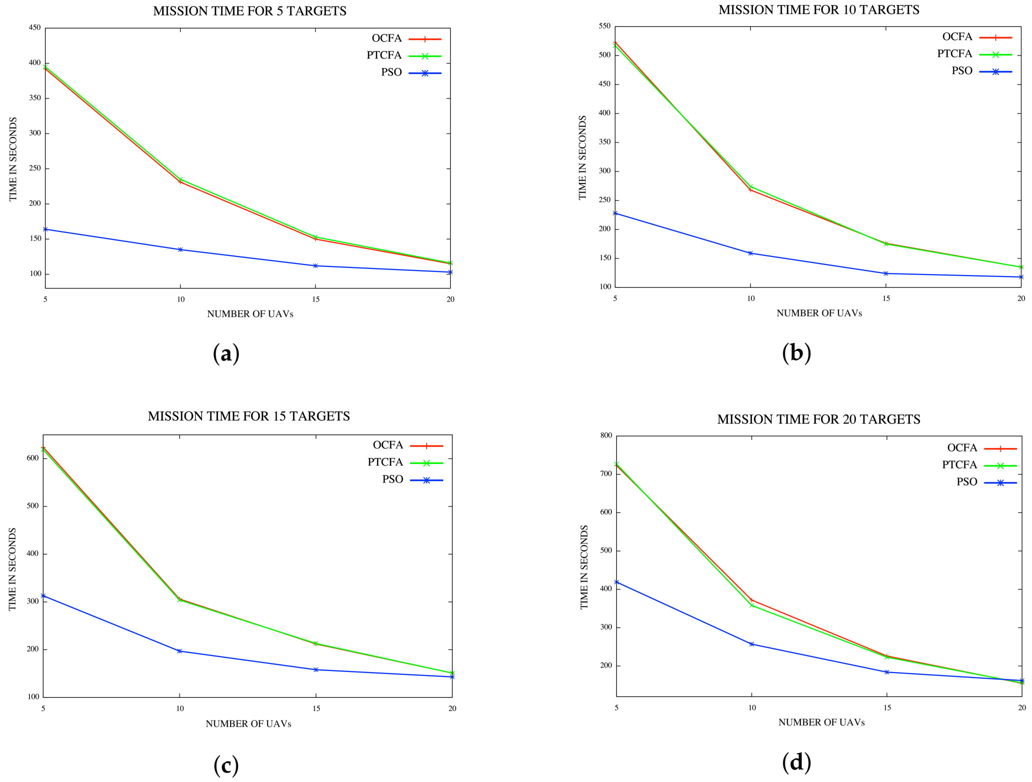
### *5.4. Comparing PSO with PTCFA and OCFA*

In the last set of experiments, the performance of PSO was compared to that of PTCFA and OCFA. For a fair comparison, sensor range was set to 1500 m to ensure that all targets or locations of interest were detectable from any position in the fly zone with the execution of PTCFA and OCFA. Average mission times and simulation times are reported for various configurations of UAVs and targets or locations of interest in Figures 13a–14d.

#### 5.4.1. Average Mission Time

The mission times of PSO, PTCFA and OCFA are reflected in Figure 13a–d. Results indicate that PSO had the best mission time of all three algorithms. PTCFA and OCFA had similar performances, with PTCFA achieving slightly better mission times than OCFA at times in Figure 13a–c. A possible explanation is that OCFA is optimized per target/location, and occasionally, the inclusion of the closest UAV in a coalition for a specific target/location may result in the distance to the next target/location being much further than had the UAV closest to the target/location been excluded from the first coalition. The OCFA algorithm optimizes a coalition w.r.t. a single target/location, and the solution proposed may not necessarily be an optimal solution across all targets.

Figure 13d reflects coalition path lengths, coalition times, mission times, coalitions and simulation times for a single run using five targets/locations and five UAVs for PTCFA and OCFA. Note, coalitions comprising different coalition members were formed for PTCFA and OCFA. PTCFA produced smaller coalition path lengths and mission completion times than OCFA. The simulation time of PTCFA was also much smaller than that of OCFA.



(**a**) (**b**)

(**c**) (**d**)

**Figure 13.** (**a**) Mission time for five targets/locations; (**b**) mission time for 10 targets/locations; (**c**) mission time for 15 targets/locations; (**d**) mission time for 20 targets/locations.

Results for a single run are recorded in Table 2. Note the various coalitions formed for PTCFA and OCFA. Results show that different coalitions were formed by the two algorithms. Although OCFA had a lower mission time than PTCFA, its coalition path lengths and simulation times were higher than those of PTCFA. PTCFA produced smaller mission times than OCFA in 10 out the 100 simulations.

**Table 2.** Comparison of a single execution between PTCFA and OCFA for 5 targets/locations and 5 UAVs.

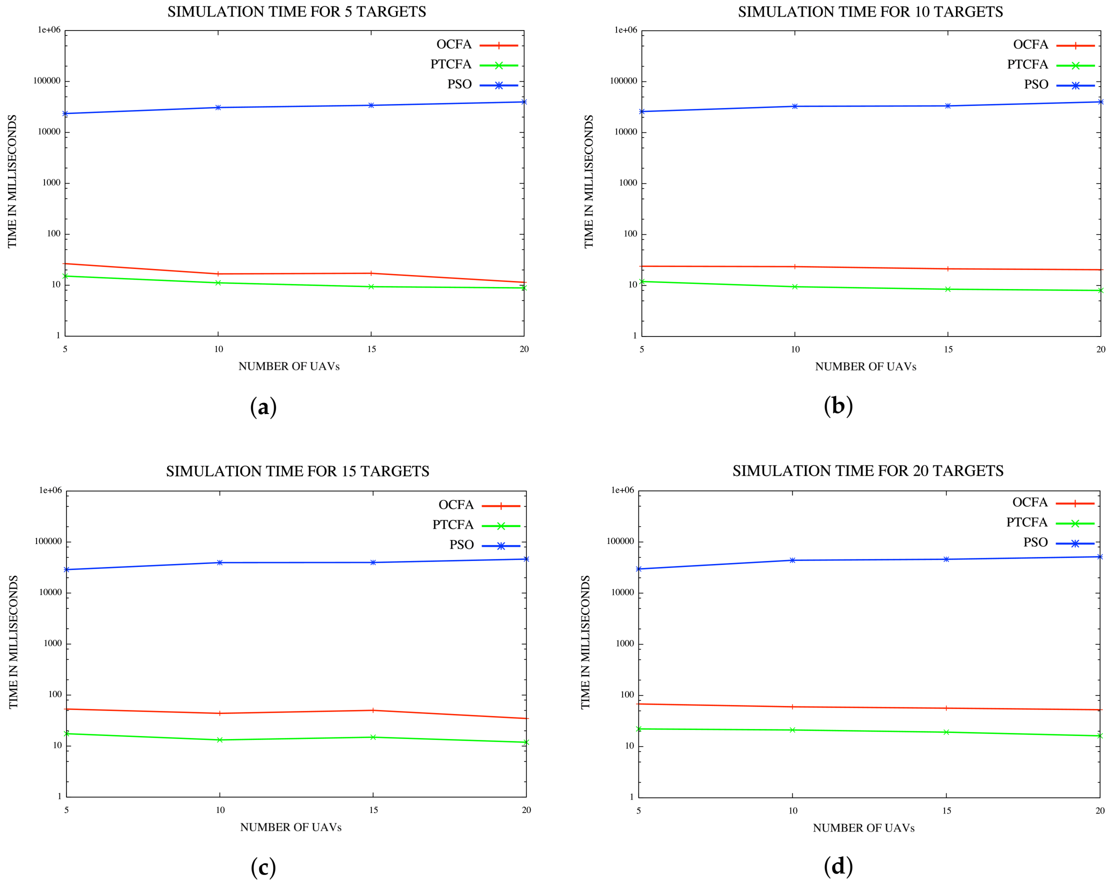
| PTCFA | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Location or Target No.** | **Agents** | **Detected at Time in Seconds** | **Coalition Path Length in Meters** | **Coalition Time in Seconds** | **Coalition Leader** | **Mission Completion Time in Seconds** | **Simulation Time in Milliseconds** |
| 4 | 0 | 0.0 | 362.44 | 36.2 | 0 | 269.0 | 12.0 |
| 3 | 0, 2 | 60.7 | 564.76 | 56.5 | 2 | | |
| 1 | 0 | 97.5 | 851.11 | 85.1 | 0 | | |
| 2 | 0, 3 | 195.0 | 446.48 | 44.6 | 3 | | |
| 0 | 2, 1 | 212.7 | 841.71 | 84.2 | 1 | | |
| | | Total | 3066.5 | 306.65 | | | |
| **OCFA** | | | | | | | |
| **Location or Target No.** | **Agents** | **Detected at Time in Seconds** | **Coalition Path Length in Meters** | **Coalition Time in Seconds** | **Coalition Leader** | **Mission Completion Time in Seconds** | **Simulation Time in Milliseconds** |
| 4 | 0 | 0.0 | 362.44 | 36.2 | 0 | 303.0 | 20.0 |
| 3 | 0, 2 | 60.7 | 564.76 | 56.5 | 2 | | |
| 1 | 1 | 97.5 | 791.49 | 79.1 | 1 | | |
| 2 | 3, 4 | 189.1 | 657.18 | 65.7 | 4 | | |
| 0 | 2, 1 | 212.7 | 903.14 | 90.3 | 1 | | |
| | | Total | 3279.01 | 327.9 | | | |

### 5.4.2. Average Simulation Time

The plots in Figure 14a–d indicate that PSO took the longest simulation time of all three algorithms. In the case of PSO, the average simulation time also increased with an increase in the number of UAVs. PTCFA had the lowest simulation time of all the algorithms. The simulation time of OCFA was slightly longer than that of PTFCA, but much smaller than that of PSO. Results indicate that PTCFA with the shortest simulation time of all three algorithms is the best candidate for coalition formation since the computational overhead must be minimal.

### 5.4.3. Energy Savings

In the experiments conducted by Manathara et al. in [40], data files were randomly populated and not initialized to ensure that the total sum of the quantities of resources of the UAVs was at least equal to the total sum of the quantities of the corresponding resource of all the targets. Their argument is that often, at the start of a search and prosecute mission, one is not aware of the resource requirements of the various targets. However, there are cases when the resource requirements are known at the start of a mission, e.g., the size of the data to be collected at various targets when performing a city's remote sensing is known in advance. In these cases, one can ensure that the resource requirements are met before the mission is initiated. Furthermore, if the environment data of all the locations of interest cannot be stored in the UAVs, then the mission will not be completed successfully. These unsuccessful missions imply that energy was wasted while performing the various coalitions. To quantify the benefit in energy savings, 1600 data files were populated randomly without ensuring that the UAVs had sufficient storage capacity to accommodate all the environment data to be collected at the various locations. Of the 1600 data files, in 457 cases, the UAVs could not store all the environment data collected. For the experiments reported in this paper, it was ensured that the UAVs had sufficient storage capacity to store all the data collected. In the case of data files that were randomly populated, 28.6% (= 4571/600) would not lead to successful prosecution of the targets. The approach adopted in the experiments resulted in an energy savings of 71.4%.



(**a**) (**b**) (**c**) (**d**)

**Figure 14.** (**a**) Simulation time for five targets/locations; (**b**) simulation time for 10 targets/locations; (**c**) simulation time for 15 targets/locations; (**d**) simulation for time 20 targets/locations.

## 6. Conclusions

The Dubins longest path can be used to describe the path traversed by a fixed-wing unmanned aerial vehicle in pursuit of a target. This paper provides an approach that calculates the Dubins longest path using elementary geometry. Experiments were conducted where the Dubins longest path was applied to algorithms that included particle swarm optimization, the polynomial time coalition formation algorithm and the optimal coalition formation algorithm. Simulation results indicated that although PSO had the lowest average mission completion time, and it had the longest simulation time of all three algorithms. The latter does not make PSO a good candidate for coalition formation for UAVs where minimal computational time is of paramount importance. PTCFA's smaller simulation time than OCFA and its fairly reasonable mission time for a large number of UAVs compared to PSO make it a viable option as an algorithm for coalition formation for fixed-wing UAVs. Results confirm that for each type of algorithm, an increase in the number of UAVs led to a decrease in mission time. The constraint that UAVs have sufficient resources that meet the resource requirement of the targets led to a considerable savings in energy. An increase in the sensor range produced an increase in coalition time with a corresponding reduction in mission times. Results indicate that an increase in the number of types of resources led to an increase in coalition time and also to an increase in mission time. An increase in the number of UAVs that all contained the same number of types of resources assigned to a mission led to a decrease in mission time. Results also indicate that an increase in the number of UAVs led to an increase in simulation time in the case of PSO, but to a decrease in the case of PTCFA and OCFA.

For future work, the proposed model will be improved by reducing the assumptions. Environmental conditions such as the wind and weather need to be considered, as well as their impact on path planning and coalition formation. When many UAVs must be used in a relatively small area, collision avoidance will be of paramount importance for successful completion of a mission. Collision avoidance will be investigated and applied to the coalition formation algorithms used in this study. Experiments will be extended by considering heterogeneous UAVs, as well as the effect of different UAV specifications on the performance of coalition formation in data muling.

**Author Contributions:** Conceptualization, A.I. and B.A.B.; Methodology, A.I.; Software, A.I.; Validation, A.I., B.A.B. and E.T.; Formal Analysis, A.I., E.T. and B.A.B.; Investigation, A.I. and E.T.; Resources, A.I. and B.A.B.; Data Curation, A.I.; Writing—Original Draft Preparation, A.I. and B.A.B.; Writing—Review & Editing, A.I., E.T. and B.A.B.; Visualization, A.I. and B.A.B.; Supervision, B.A.B.; Project Administration, B.A.B.; Funding Acquisition, B.A.B.

**Funding:** The authors are grateful to the University of the Western Cape for the financial assistance provided.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zlot, R.; Stentz, A. Market-based multirobot coordination for complex tasks. *Int. J. Robot. Res.* **2006**, *25*, 73–101. [CrossRef]
2. Ahmed, A.; Patel, A.; Brown, T.; Ham, M.; Jang, M.; Agha, G. Task assignment for a physical agent team via a dynamic forward/reverse auction mechanism. In Proceedings of the International Conference of Integration of Knowledge-Intensive Multi-Agent Systems: Modelling, Evolutions and Engineering, Waltham, MA, USA, 18–21 April 2005 ; pp. 311–317.
3. Berhault, M.; Huang, H.; Keskinocak, P.; Koenig, S.; Elmaghraby, W.; Grin, P.; Kleywegt, A. Robot exploration with combinatorial auctions. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 27–31 October 2003; pp. 1957–1962.
4. Henriques, R.; Bacao, F.; Lobo, V. UAV path planning based on event density detection. In Proceedings of the International Conference on Advanced Geographic Information Systems and Web Services, Cancun, Mexico, 1–7 February 2009; pp. 112–116.
5. Tisdale, J.; Kim, Z.; Hedrick, J.K. Autonomous UAV path planning and estimation: An online path planning framework for cooperative search and localization. *IEEE Robot. Autom. Mag.* **2009**, *16*, 35–42. [CrossRef]
6. Pongpunwattana, A.; Rysdyk, R. Real-time planning for multiple autonomous vehicles in dynamic uncertain environments. *J. Aerosp. Comput. Inf. Commun.* **2005**, *1*, 580–604. [CrossRef]
7. Gurfil, P. Evaluating UAV flock mission performance using Dudek's taxonomy. In Proceedings of the 2005 American Control Conference, Portland, OR, USA, 8–10 June 2005; pp. 4679–4684.
8. Simmons, R.; Apfelbaum, D.; Burgard, W.; Fox, D.; Moors, M.; Thrun, S.; Younes, H. Coordination for multi-robot exploration and mapping. In Proceedings of the National Conference on Artificial Intelligence, Austin, TX, USA, 30 July– 3 August 2000; pp. 852–858.
9. Zlot, R.; Stentz, A.; Dias, M.B.; Thayer, S. Multi-robot exploration controlled by a market economy. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; pp. 3016–3023.
10. Ding, X.C.; Rahmani, A.; Egerstedt, M. Optimal multi-UAV convoy protection. In Proceedings of the 2nd International Conference on Robot Communication and Coordination, Odense, Denmark, 31 March–2 April 2009; pp. 1–6.
11. Nigam, N.; Kroo, I. Persistent surveillance using multiple unmanned air vehicles. In Proceedings of the 2008 IEEE Aerospace Conference, Big Sky, Montana, 1–8 March 2008; pp. 1–14.
12. Rubio, J.C.; Vagners, J.; Rysdyk, R. Adaptive path planning for autonomous UAV oceanic search missions. In Proceedings of the AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, USA, 20–22 September 2004.
13. Bagula, A.; Tuyishimire, E.; Wadepoel, J.; Boudriga, N.; Rekhis, S. Internet-of-things in motion: A cooperative data muling model for public safety. In Proceedings of the International IEEE 2016 Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, Toulouse, France, 18–21 July 2016; pp. 17–24.

14. Tuyishimire, E.; Ismail, A.; Rekhis, S.; Bagula, B.A.; Boudriga, N. Internet of things in motion: A cooperative data muling model under revisit constraints. In Proceedings of the International IEEE 2016 Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress,Toulouse, France, 18–21 July 2016; pp. 1123–1130.
15. Cunningham, C.T.; Roberts, R.S. An adaptive path planning algorithm for cooperating unmanned air vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea, 21–26 May 2001; pp. 3981–3986.
16. Tuyishimire, E.; Ismail, A.Optimal clustering for efficient data muling in the internet-of-things in motion. In Proceedings of the International Symposium on Ubiquitous Networking, Hammamet, Tunisia, 2–5 May 2018; Springer: New York, NY, USA , 2018; pp. 35–41.
17. Techy, L.; Woolsey, C.A.; Schmale, D.G. Path planning for efficient UAV coordination in aerobiological sampling missions. In Proceedings of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 2814–2819.
18. Han, Z.; Swindlehurst, A.L.; Liu, K.R. Smart deployment/movementof unmanned air vehicle to improve connectivity in MANET. In Proceedings of the 2006 IEEE Wireless Communications and Networking Conference (WCNC), Las Vegas, NV, USA, 3–6 April 2006; pp. 252–257.
19. Bagula, A.; Castelli, L.; Zennaro, M. On the design of smart parking networks in the smart cities: An optimal sensor placement model. *Sensors* **2015**, *15*, 15443–15467. [CrossRef] [PubMed]
20. Karbab, E.M.; Djenouri, D.; Boulkaboul, S.; Bagula, A. Car park management with networked wireless sensors and active RFID. In Proceedings of the IEEE International Conference on Electro/Information Technology, Dekalb, IL, USA, 21–23 May 2015; pp. 373–378.
21. Masinde, M.; Bagula, A. A framework for predicting droughts in developing countries using sensor networks and mobile phones. In Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, Bela Bela, South Africa, 11–13 October 2010; pp. 390–393.
22. Van't Hof, G. Ambulancedrone gaat levens redden. *Dutch Newspaper: 'Algemeen Dagblad'*, 28 October 2014.
23. Jimenez, L.A.V. Market-Based Distributed Task Allocation Methodologies Applied to Multi-Robot Exploration. Ph.D. Thesis, University of Seville, Sevilla, Spain, 2009.
24. Dubins, L.E. On curves of minimal length with a constraint on average curvature. *Am. J. Math.* **1957**, *79*, 497–516. [CrossRef]
25. Reeds, J.A.; Shepp, L.A. Optimal paths for a car that goes both forwards and backwards. *Pac. J. Math.* **1990**, *145*, 367—393. [CrossRef]
26. Kavraki, L.E.; Svestka, P.; Latombe, J.-C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
27. Subchan, S.; White, B.A.; Tsourdos, A.; Shanmugavel, M.; Żbikowski, R. Dubins path planning of multiple UAVs for tracking contaminant cloud. In Proceedings of the 17th International Federation of Automatic Control (lFAC) World Congress, Seoul, Korea, 6–11 July 2008; Volume 41, pp. 5718–5723.
28. Lugo-Cárdenas, I.; Flores, G.; Salazar, S.; Lozano, R. Dubins path generation for a fixed-wing UAV. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems, Orlando, FL, USA, 27–30 May 2014.
29. Costanzi, R.; Fanelli, F.; Meli, E.; Ridolfi, A.; Allotta, B.Generic path planning algorithm for mobile robots based on Bézier curves. In Proceedings of the 9th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2016, Leipzig, Germany, 29 June–1 July 2016; Volume 49, pp. 145–150.
30. Tuyishimire, E.; Bagula, A.; Rekhis, S.; Boudriga, N. Cooperative data muling from ground sensors to base stations using UAVs. In Proceedings of the 2017 IEEE Symposium on Computers and Communications, Heraklion, Greece, 3–6 July 2017; pp. 35–41.
31. Lin, L.; Sun, Q.; Wang, S.; Yang, F. Research on PSO based multiple UAVs real-time task assignment. In Proceedings of the Control and Decision Conference, Guiyang, China, 25–27 May 2013.
32. Fargeas, J.L.; Kabamba, P.; Girard, A. Cooperative surveillance and pursuit using unmanned aerial vehicles and unattended ground sensors. *Sensors* **2015**, *15*, 1365–1388. [CrossRef] [PubMed]
33. Kwok, N.M.; Ha, Q.P.; Fang, G. Motion coordination for Construction vehicles using swarm intelligence. *Int. J. Adv. Robot. Syst.* **2007**, *4*, 44 . [CrossRef]

34. Bellingham, J.; Tillerson, M.; Richards, A.; How, J.P. Multi-task allocation and path planning for cooperating UAVs. In *Cooperative Control: Models, Applications and Algorithms*; Butenko, S., Murphey, R., Pardalos, P.M., Eds.; Cooperative Systems; Springer: Boston, MA, USA, 2003; Volume 1.
35. Tong, H.; Chao, W.W.; Qiang, H.C.; Bo, X.Y. Path planning of UAV based on Voronoi diagram and DPSO. In Proceedings of the International Workshop on Electronics and Engineering, Xi'an, China, 23–25 August 2012; Volume 29, pp. 4198–4203.
36. Howlett, J.K.; McLain, T.; Goodrich, M. Learning real-time A* path planner for unmanned air vehicle target sensing. *AIAA J. Aerosp. Comput. Inf. Commun.* **2006**, *3*, 108–122. [CrossRef]
37. Oh, H.; Shin, H.S.; Tsourdos, A.; White, B.A.; Silson, P. Coordinated road network search for multiple UAVs using Dubins path. In *Advances in Aerospace Guidance, Navigation and Control*; Springer: New York, NY, USA, 2011; pp. 55–65.
38. Schwarzrock, J.; Zacarias, I.; Bazzan, A.L.C.; Fernandes, R.Q.D.; Moreira, L.H.; de Freitas, E.P. Solving task allocation problem in multi unmanned aerial vehicles systems using swarm intelligence. *Eng. Appl. AI* **2018**, *72*, 10–20. [CrossRef]
39. Liu, W.; Zheng, Z.; Cai, K. Adaptive path planning for unmanned aerial vehicles based on bi-level programming and variable planning time interval. *Chin. J. Aeronaut.* **2013**, *26*, 646–660. [CrossRef]
40. Manathara, J.G.; Surit, P.B.; Beard, R.W. Multiple UAV coalitions for a search and prosecute mission. *J. Intell. Robot. Syst.* **2011**, *62*, 125–158. [CrossRef]
41. Ismail, A.; Tuyishimire, E.; Bagula, A. Generating Dubins path for fixed-wing UAVs in search missions. In Proceedings of the International Symposium on Ubiquitous Networking, Hammamet, Tunisia, 2–5 May 2018; Springer: New York, NY, USA, 2018.
42. Kennedy, J.F.; Kennedy, J.; Eberhart, R.C. *Swarm Intelligence*; Morgan Kaufmann Publishers: Burlington, MA, USA, 2001.
43. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.