Aerospace Science and Technology 77 (2018) 168–179

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Grey wolf optimization based sense and avoid algorithm in a Bayesian framework for multiple UAV path planning in an uncertain environment ☆

Mohammadreza Radmanesh [a,*], Manish Kumar [b], Mohammad Sarim [b]

[a] Department of Mechanical and Materials Engineering/Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA

[b] Department of Mechanical and Materials Engineering, University of Cincinnati, Cincinnati, OH 45221, USA

ARTICLE INFO

Article history:
Received 16 May 2017
Received in revised form 22 January 2018
Accepted 20 February 2018
Available online 8 March 2018

Keywords:
Grey wolf optimization
Trajectory planning
Cooperative flight
Bayesian framework

ABSTRACT

Unmanned Air Vehicles (UAVs), which have been popular in the military context, have recently attracted attention of many researchers because of their potential civilian applications. However, before UAVs can fly in civilian airspace, they need to be able to navigate safely to their goal while maintaining separation with other manned and unmanned aircraft during the transit. Algorithms for autonomous navigation of UAVs require access to accurate information about the state of the environment in order to perform well. However, this information is often uncertain and dynamically changing. In this paper, a Grey Wolf Optimization (GWO) based algorithm is proposed to find the optimal UAV trajectory in presence of moving obstacles, referred to as Intruder Aircraft (IAs), with unknown trajectories. The solution uses an efficient Bayesian formalism with a notion of cell weighting based on Distance Based Value Function (DBVF). The assumption is that the UAV is equipped with the Automatic Dependent Surveillance-Broadcast (ADS-B) and is provided with the position of IAs either via the ADS-B or ground-based radar. However, future trajectories of the IAs are unknown to the UAV. The proposed method is verified using simulations performed on multiple scenarios. The results demonstrate the effectiveness of the proposed method in solving the trajectory planning problem of the UAVs.

1. Introduction

Unmanned Air Vehicles (UAVs) have traditionally been used in military operations for a number of years. Recently, UAVs have generated a lot of interest due to their potential application in civilian domains such as emergency management, law enforcement, precision agriculture, package delivery, and imaging/surveillance [1]. However, before the usage of UAVs becomes a reality in civilian domains, a number of technological challenges need to be overcome. Particularly, the challenges emanating from integration of UAVs in the National Airspace System (NAS) are extremely critical to be solved before they can start flying in civilian airspace. An important challenge among these is the ability for the UAVs to not only plan their path for fulfilling a mission but also to re-plan or adjust the trajectory (called Sense and Avoid capability) in order to avoid collision with other aircraft. Furthermore, the increase in the number of aircraft has been dramatic over the last 50 years. This increase in manned aircraft along with incorporation of unmanned fleet in future will pose severe challenges to the current Air Traffic Control (ATC). Hence, the Radio Technical Commission for Aviation (RTCA) and Federal Aviation Administration (FAA) have been charged with a responsibility to implement a seamless change from ATC to Air Traffic Management (ATM) by 2020 [2,3] and [4]. For the manned aircraft, the notion of pilot preferred trajectories (PPT) has been implemented to allow pilots and airlines to plan and manage the flight trajectories to their unique operational requirements. This system has been shown to be unreliable and will become less useful in a futuristic scenario that will include UAVs in the airspace. As a solution to this problem, the automatic dependent surveillance broadcast (ADS-B) system for transfer of in-flight data is proposed to be used during the flight by the year 2020. One of the predicted advantages of implementing ADS-B is that by enhancing the autonomy of flights in NAS, an

☆ A preliminary version of this paper was presented in International Conference on Unmanned Aircraft Systems 2016 (ICUAS 2016).

* Corresponding author.
E-mail addresses: Radmanma@mail.uc.edu (M. Radmanesh), Manish.Kumar@uc.edu (M. Kumar).

https://doi.org/10.1016/j.ast.2018.02.031
1270-9638/

Nomenclature

ρ	Flight path	m	Index of the UAV
d	Rows in tessellated area	$R_{m,t}$	Set of all cells in ϕ which the UAV m can occupy in the next time step $t+1$
$DBVF$	Distance Based Value Function	$R_{m,t}^{i\in\phi}$	Event of choosing the ith cell from the $R_{m,t}$ which can be occupied by the UAV m in next step $t+1$
$E_{m,t}^{i\in\phi}$	Event of receiving the accurate data in next step $t+1$	t	Time step
$F_{m,t}$	Set of cells in ϕ that provides the danger of collision by IA in time step $t+1$	$T_{m,t}$	Set of all cells around UAV m which can be intruded by IA at time $t+1$.
$F_{m,t}^{i\in\phi}$	Event of all cell $i\in\phi$ in the neighborhood of UAV m which has the possibility of collision in the next step $t+1$.	$T_{m,t}^{i\in\phi}$	Event of intruding the ith cell in collision area around UAV m being intruded by the IA in the next step $t+1$
$H_{m,t}$	Set of all decisions that IA could make for the next time step $t+1$.	v	Columns in tessellated area
$H_{m,t}^{i\in\phi}$	Event of choosing cell $i\in H_{m,t}$ by the IA as the next position at time step $t+1$.	$x_{0,m}$	Initial location of UAV m
J	Cost function	$x_{f,m}$	Final location of UAV m

aircraft could navigate with minimum pilot interference and ultimately fly fully independent of pilots [5–7].

There are various methods for calculating escape trajectories that have been proposed in literature for collision avoidance including classical control [8], Fuzzy Logic [9], E-Field maneuver planning [10,11], game theory [12], Mixed Integer Linear Programming (MILP) [13,14], and its application in NC Machines path planning [15,16], automotive trajectory planning [17], and air traffic management [18]. Path planning for UAVs have been studied as a part of task assignment problem [19–23] and [24].

Applications of multiple UAVs for different applications have attracted many researchers. Apart from the fact that multiple UAVs provide the ability to perform complex and heterogeneous tasks, one of the advantages of cooperative flight performances is also fuel saving [25,26]. Path planning of such systems offers many challenging problems from both theoretical and practical points of view [27]. Flight formation refers to a particular problem of management of a group of UAVs flying in tight cooperation within a defined volume [28], and often with a pre-defined shape. Although studies on active path planning of a UAV have been considered many times (e.g., see [29–31]), cooperative path planning approaches for UAVs have only recently begun to appear. The problem of formation flight is widely studied in literature. Considering only the flight control, classical leader-wingman configuration is investigated via proportional-integral control [32] or non-linear control [33]. A reactive behavior-based controller is discussed in [34]. Proposed solution for trajectory optimization of large formations using centralized or distributed algorithms is discussed respectively in [35,36], taking into account some constraints on the shape of the formation. Reconfiguration in the formations is introduced in [37] by proposing a scheme where trajectories are computed off-line for switching between a limited number of formation configurations. In [38,20,39,40], by implementation of mixed-integer linear programming (MILP), tightly-coupled task assignment problems with timing constraints are solved for a group of UAVs.

The problem of collision avoidance becomes more complicated in real-world scenarios which present several challenges, the most significant among them being uncertainty. This is relevant in NAS since IAs could be added at any time, their flight plans may not be shared (non-cooperating IAs) or erroneous due to sensing errors or communication delays. Keeping these issues in view, this paper focuses on scenarios where the information about IAs has uncertainties associated with it. There has been a mass of work for UAV trajectory planning under uncertainties [41–44] and [45]. Furthermore, methods based on Bayesian mathematics have been vastly used to overcome different challenges during the path planning of the UAVs and proved to be very helpful in this area [46–48] and [49].

This paper extends the previous works via the use of Distance Based Value Function (DBVF) and utilization of Bayesian update method for building the risk map based on uncertain information provided by ADS-B and radar. This then allows the information to be incorporated into an optimization scheme based on Grey Wolf Optimization (GWO) that plans the paths for the vehicles during the flight mission. GWO is a novel intelligent algorithm developed in [50]. Implementation of this method on a number different applications has shown its stability and promising convergence speed [51,52] and [53]. This method is developed based on social hierarchy of predatory grey wolves. The approach exploits the information that is used to model the intruder aircraft with uncertain motion and appropriately takes that into account while planning the paths in a dynamic fashion. In [54], a GWO algorithm was employed to find an optimal path for UAVs two-dimensional path planning problem in difficult combating environments. The cooperative target tracking by multi-UAVs in urban environment was studied in [55]. In [55], by formulating the problem as an optimization problem, and solving that by integrating the real-time performance of Model Predictive Control (MPC) and the strong searching ability of GWO, trajectory planning is achieved in an optimal manner. Finally, multiple flight scenarios relevant to NAS are utilized to demonstrate the effectiveness of the proposed method.

The paper is organized as follows: in the next section, a cost function is formulated for this problem. Then, in section 3, the DBVF method is introduced. In section 4, a general formulation of the problem is presented. Then in section 5, we describe our efficient Bayesian method. Subsequently, the GWO algorithm is presented in section 6. Dynamics of the quadcopter considered for this problem is defined in section 7. Finally, several flight scenarios and flight simulation results and analysis are provided in sections 8 and 9 respectively.

2. Trajectory planning model and formulation

We assume that the UAV can fly only within a defined speed range and has limited maneuverability. The UAV m travels in the region R. In this paper, ρ is the flight path of the UAV denoted as the set of all unit areas or cells in R from the initial location $x_{0,m}$ to the goal position of $x_{f,m}$. The problem under consideration can be formulated as weighted anisotropic shortest path problem. The objective is to look for optimal path ρ^* such that:

$$J[\rho^*] = min(J[\rho]_{\rho\in R}) \tag{1}$$

where $J[\rho]$ denotes the cost function defined by:

$$J[\rho] = \sum_{i(t)\in\rho} W(i(t)) \tag{2}$$

$W(i(t))$ in the above equation represents the weight function of ith cell at time t for the path ρ that includes the distance and risk associated with a path. A derivation for this weight function is presented in Section 5 (see Equation (24)). Risk function is defined by the sum of cell weights of the entire path. Cell weight associated with a particular cell represents a measure of risk of collision with other aircraft and stationary obstacles. This is calculated in a dynamic sense based on the data obtained from radar or ADS-B. It may be noted that the total weight of the path is the combination of the distance from the goal position and the risk of collision with the intruder aircraft.

3. Distance based value function (DBVF)

In the current formulation, as mentioned above, the entire flight region is divided into a set of tessellated areas. The initial weight function is based on DBVF and is obtained in a way that the weight of a cell increases as its distance from the goal position increases.

Consider that the area has been divided into v columns and d rows. Equation (3) represents the value of the cell based on DBVF.

$$\begin{aligned} &\forall i \in v \times d \\ &DBVF(i) = \min_{j\in N_e} \{DBVF(j)\} + 1 \\ &DBVF(Goal) = 0 \end{aligned} \tag{3}$$

where N_e represents all the cells in the immediate neighborhood of cell i. Equation (3) represents a value or cost associated with a cell that encodes the distance of that cell from the goal cell. A larger distance corresponds to a larger DBVF. Minimizing this value is responsible for navigating the UAV from the current location and to the goal location.

4. Risk model

This section presents our proposed method of determining risk of collision. The risk essentially quantifies the danger of collision of the $m^{th} \in [0, ..., M]$ UAV in ith cell due to the presence of Intruder Aircraft (IAs) in set of cells ϕ in the neighborhood of UAV m. The risk at time t is modeled as:

$$\sum_{i\in\rho} \mathbb{C}_i(R_m^\phi, F_m^\phi, T_m^\phi, E_m^\phi, H_m^\phi) \tag{4}$$

in which R, F, T, E and H in ϕ over path ρ for a specified UAV m represent:

- $R_{m,t}$: Set of all cells in ϕ which the UAV m can occupy in the next time step $t+1$.
- $R_{m,t}^{i\in\phi}$: Event of choosing the ith cell from the $R_{m,t}$ which can be occupied by the UAV m in next step $t+1$.
- $F_{m,t}$: Set of cells in ϕ that provides the danger of collision by IA in time step $t+1$.
- $F_{m,t}^{i\in\phi}$: Event of all cell $i \in \phi$ in the neighborhood of UAV m which has the possibility of collision in the next step $t+1$.
- $T_{m,t} = \{\phi \cap Collision \quad Area\}$: Set of all cells around UAV m which can be intruded by IA at time $t+1$. Collision area is defined as an area that represents a risk of collision with the UAV.
- $T_{m,t}^{i\in\phi}$: Event of intruding the ith cell in collision area around UAV m being intruded by the IA in the next step $t+1$.

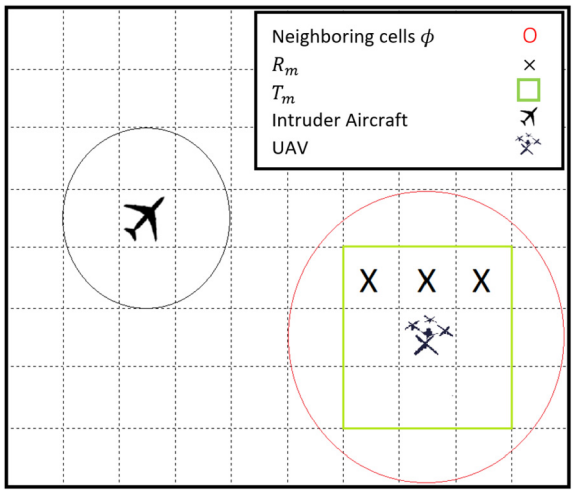

Fig. 1. Schematic diagram showing the UAV, intruder aircraft, and neighboring cells of the UAV ϕ. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

- $E_{m,t}^{i\in\phi}$: Event of receiving the accurate data in next step $t+1$.
- $H_{m,t}$: Set of all decisions that IA could make for the next time step $t+1$.
- $H_{m,t}^{i\in\phi}$: Event of choosing cell $i \in H_{m,t}$ by the IA as the next position at time step $t+1$.

For the sake of simplicity in writing, the subscript t has been omitted in the equations and $i \in \phi$ is replaced by only ϕ.

4.1. Derivation of the risk function

First of all, we divide the entire area into discrete, equal-area cells. Each cell represents a 2-D area and corresponding to each cell the information about the environment or more specifically any potential threat within that physical area presented by the cell is attached. The weight of a cell or a collection of cells is defined using the event definitions. Let an over bar on these events represent the complement of the event. For example, $\bar{F}_m$ is the set of cells in ϕ which are not going to be intruded by an intruder aircraft. There are some relationships between these sets that are utilized in derivation of probability. For example, the set of cells that UAV can occupy in the next time step is a subset of the collision area, i.e., $R_m \subseteq T_m$. Likewise, the region ϕ needs to be intruded to have a possibility of collision, so $T_m \subset F_m$. Additionally, event H_m^ϕ is an independent event and only depends on the location of intruder aircraft. Fig. 1 represents the schematic showing the different variables. Local goal is represented by the lowest weighted cell placed in ϕ.

For a specific UAV m, the event of receiving the data accurately, E_m^ϕ, is based on the performance of the ADS-B and/or radar. The maneuvering of intruder aircraft would affect the parameter H_m^ϕ which is independent of the events such as E_m^ϕ, T_m^ϕ, R_m^ϕ and F_m^ϕ. The observation factor for $m \in [0, ..., M]$ is defined as $P(F_m^\phi, H_m^\phi, E_m^\phi)$ and is given by:

$$P(F_m^\phi, H_m^\phi, E_m^\phi) = P(F_m^\phi, E_m^\phi).P(H_m^\phi) \tag{5}$$

Using the conditional probability relationship, Equation (5) can be written as:

$$P(F_m^\phi, E_m^\phi).P(H_m^\phi) = P(F_m^\phi \mid E_m^\phi).P(E_m^\phi).P(H_m^\phi) \tag{6}$$

Let σ be a function that takes as an argument a collection of cells (ϕ) and returns the observation factor for the collection of

cells. The total observation factor σ for ϕ based on all intruder aircraft is obtained from the formula:

$$\sigma = \sum_{m=1}^{M} P(F_m^{\phi}, H_m^{\phi}, E_m^{\phi}) = \sum_{m=1}^{M} P(F_m^{\phi} \mid E_m^{\phi}).P(E_m^{\phi}).P(H_m^{\phi}) \tag{7}$$

The probability value $P(H_m^{\phi})$ of cells is assumed to be uniformly distributed over all the possible cells that can be occupied by the IA in the next time step and is represented by constant C_m^x for cell $x \in H_m$.

Here, N^i and N are the cardinality of H_m and ϕ.

$$N^i = |H_m|\,, N = |\phi|$$

Therefore the observation factor can be written as:

$$\sigma(\phi) = \sum_{m=1}^{M} \sum_{x=1}^{N} P(F_m^{\phi} \mid E_m^{\phi}).P(E_m^{\phi}).C_m^x \tag{8}$$

The observation factor, σ, quantitatively represents the risk or probability of collision in the area ϕ.

5. Heuristic update method of the cell weights

Now, H_m^{ϕ} and $\bar{H_m^{\phi}}$ are mutually exclusive and collectively exhaustive, i.e., $P(H_m^{\phi}) + P(\bar{H_m^{\phi}}) = 1$. Here, the probability (or belief) that the IA has not entered the region ϕ is defined as $\bar{H_m^{\phi}}$. In this case, the value of σ depends on the chance of intruder aircraft entering ϕ.

Similarly, it is clear that events F_m^{ϕ} and $\bar{F_m^{\phi}}$ are mutually exclusive and collectively exhaustive, hence $P(F_m^{\phi}) = 1 - P(\bar{F_m^{\phi}})$. Let $\bar{T_m^{\phi}}$ be the complement of event T_m^{ϕ}. Here $\bar{T_m^{\phi}}$ represents the event that an intruder is in ϕ but not in cells with the possibility of collision. Hence, $P(\bar{T_m^{\phi}}) = 1 - P(T_m^{\phi})$.

It is to be noted that since an intruder aircraft must first enter ϕ in order to enter the collision area, event $T_m^{\phi} \subset F_m^{\phi}$. Similarly, if the intruder aircraft is not placed in ϕ then it cannot be placed in the collision area. Mathematically, it means $\bar{F_m^{\phi}} \subset \bar{T_m^{\phi}}$.

The heuristic method of cell weight assignment based on the probability of collision is carried out in three steps. The first step calculates the probability of intruder aircraft entering region ϕ. The second step calculates the probability of intruder aircraft entering the collision area T_m^{ϕ}. The third step calculates the probability of intruder aircraft entering region R_m^{ϕ}, which is the set of cells that the UAV may occupy in the next time step.

Step 1: Calculation of posterior probability of ϕ being intruded after IA makes its decision

The posterior probability of intruder aircraft entering the ϕ can be given by:

$$P(F_m^{\phi} \mid H_m^{\phi}, E^{\phi}) = \frac{P(H_m^{\phi} \mid F_m^{\phi}, E^{\phi}) P(F_m^{\phi} \mid E^{\phi})}{P(H_m^{\phi} \mid E^{\phi})} \tag{9}$$

Let us consider each component of the Equation (9). First,

$$\begin{aligned} &P(H_m^{\phi} \mid F_m^{\phi}, E^{\phi}) \\ &= P(H_m^{\phi}, T_m^{\phi} \mid F_m^{\phi}, E^{\phi}) + P(H_m^{\phi}, \bar{T_m^{\phi}} \mid F_m^{\phi}, E^{\phi}) \\ &= P(H_m^{\phi} \mid T_m^{\phi}, F_m^{\phi}, E^{\phi}) P(T_m^{\phi} \mid F_m^{\phi}, E^{\phi}) \\ &\quad + P(H_m^{\phi} \mid \bar{T_m^{\phi}}, F_m^{\phi}, E^{\phi}) P(\bar{T_m^{\phi}} \mid F_m^{\phi}, E^{\phi}) \end{aligned} \tag{10}$$

It may also be noted that:

$$P(\bar{T_m^{\phi}} \mid F_m^{\phi}, E^{\phi}) = 1 - P(T_m^{\phi} \mid F_m^{\phi}, E^{\phi}) \tag{11}$$

Furthermore,

$$\begin{aligned} &P(H_m^{\phi} \mid T_m^{\phi}, F_m^{\phi}, E^{\phi}) \\ &= P(H_m^{\phi}, R_m^{\phi} \mid T_m^{\phi}, F_m^{\phi}, E^{\phi}) + P(H_m^{\phi}, \bar{R_m^{\phi}} \mid T_m^{\phi}, F_m^{\phi}, E^{\phi}) \\ &= P(H_m^{\phi} \mid R_m^{\phi}, T_m^{\phi}, F_m^{\phi}, E^{\phi}) P(R_m^{\phi} \mid T_m^{\phi}, F_m^{\phi}, E^{\phi}) \\ &\quad + P(H_m^{\phi} \mid \bar{R_m^{\phi}}, T_m^{\phi}, F_m^{\phi}, E^{\phi}) P(\bar{R_m^{\phi}} \mid T_m^{\phi}, F_m^{\phi}, E^{\phi}) \\ &= P(H_m^{\phi} \mid R_m^{\phi}, E^{\phi}) P(R_m^{\phi} \mid T_m^{\phi}, E^{\phi}) \\ &\quad + P(H_m^{\phi} \mid \bar{R_m^{\phi}}, T_m^{\phi}, E^{\phi}) P(\bar{R_m^{\phi}} \mid T_m^{\phi}, E^{\phi}) \end{aligned} \tag{12}$$

It may be noted that Equation (12) is simplified using the following relation:

$$R_m^{\phi} \subseteq T_m^{\phi} \subset F_m^{\phi} \tag{13}$$

Hence the probability $P(H_m^{\phi} \mid R_m^{\phi}, T_m^{\phi}, F_m^{\phi}, E^{\phi})$ can be simplified to $P(H_m^{\phi} \mid R_m^{\phi}, E^{\phi})$. Similarly:

$$\begin{aligned} &P(R_m^{\phi} \mid F_m^{\phi}, T_m^{\phi}, E^{\phi}) \\ &= P(R_m^{\phi} \mid T_m^{\phi}, E^{\phi}) \\ &= 1 - P(\bar{R_m^{\phi}} \mid F_m^{\phi}, T_m^{\phi}, E^{\phi}) = 1 - P(\bar{R_m^{\phi}} \mid T_m^{\phi}, E^{\phi}) \\ &= 1 - \frac{P(T_m^{\phi} \mid \bar{R_m^{\phi}}, E^{\phi}) P(\bar{R_m^{\phi}} \mid E^{\phi})}{P(T_m^{\phi} \mid E^{\phi})} = 1 - \frac{\zeta_m^{\phi}(1 - \xi_m^{\phi})}{\Lambda_m^{\phi}} \end{aligned} \tag{14}$$

where:

$$\zeta_m^{\phi} = P(T_m^{\phi} \mid \bar{R_m^{\phi}}, E^{\phi})$$
$$\xi_m^{\phi} = P(R_m^{\phi} \mid E^{\phi})$$
$$\Lambda_m^{\phi} = P(T_m^{\phi} \mid E^{\phi})$$

The probability value needed to be found is $P(F_m^{\phi} \mid E^{\phi})$. In this case:

$$\begin{aligned} P(F_m^{\phi} \mid E^{\phi}) &= P(F_m^{\phi}, T_m^{\phi} \mid E^{\phi}) \cup P(F_m^{\phi}, \bar{T_m^{\phi}} \mid E^{\phi}) \\ &= P(F_m^{\phi}, T_m^{\phi} \mid E^{\phi}) + P(F_m^{\phi}, \bar{T_m^{\phi}} \mid E^{\phi}) \\ &= P(F_m^{\phi}, T_m^{\phi}, R_m^{\phi} \mid E^{\phi}) + P(F_m^{\phi}, T_m^{\phi}, \bar{R_m^{\phi}} \mid E^{\phi}) \\ &\quad + P(F_m^{\phi}, \bar{T_m^{\phi}}, R_m^{\phi} \mid E^{\phi}) + P(F_m^{\phi}, \bar{T_m^{\phi}}, \bar{R_m^{\phi}} \mid E^{\phi}) \end{aligned} \tag{15}$$

in which:

$$P(F_m^{\phi}, T_m^{\phi}, R_m^{\phi} \mid E^{\phi}) = P(R_m^{\phi} \mid E^{\phi}) = \zeta_m^{\phi},$$
$$P(F_m^{\phi}, \bar{T_m^{\phi}}, R_m^{\phi} \mid E^{\phi}) = 0,$$
$$P(F_m^{\phi}, T_m^{\phi}, \bar{R_m^{\phi}} \mid E^{\phi}) =$$
$$P(T_m^{\phi}, \bar{R_m^{\phi}} \mid E^{\phi}) = P(T_m^{\phi} \mid \bar{R_m^{\phi}}, E^{\phi}) P(\bar{R_m^{\phi}} \mid E^{\phi}) = \zeta_m^{\phi}(1 - \xi_m^{\phi}),$$
$$P(F_m^{\phi}, \bar{T_m^{\phi}}, \bar{R_m^{\phi}} \mid E^{\phi}) =$$
$$P(F_m^{\phi}, \bar{T_m^{\phi}} \mid E^{\phi}) = P(F_m^{\phi} \mid \bar{T_m^{\phi}}, E^{\phi}) P(\bar{T_m^{\phi}} \mid E^{\phi}) = \varrho_m^{\phi}(1 - \Lambda_m^{\phi})$$

where:

$$\varrho_m^{\phi} = P(F_m^{\phi} \mid \bar{T_m^{\phi}}, E^{\phi})$$

After all these calculations, the value for $P(F_m^{\phi} \mid E^{\phi})$ can be written as:

$$P(F_m^{\phi} \mid E^{\phi}) = \zeta_m^{\phi}(2 - \xi_m^{\phi}) + \varrho_m^{\phi}(1 - \Lambda_m^{\phi}) \tag{16}$$

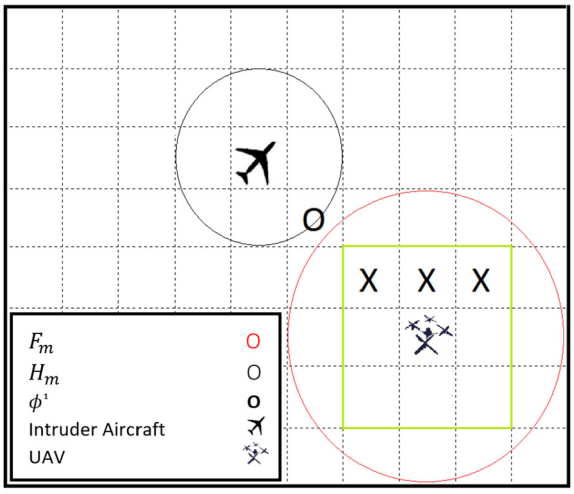

Fig. 2. Schematic diagram showing ϕ^1.

Finally, the total value for the $P(F_m^\phi \mid H_m^\phi, E^\phi)$ is calculated by Equation (17).

$$P(F_m^\phi \mid H_m^\phi, E^\phi) \tag{17}$$
$$= \frac{1}{\nu_m^\phi}[((\chi_m^\phi . \frac{\zeta_m^\phi(1-\xi_m^\phi)}{\Lambda_m^\phi}) + \varpi_m^\phi . (1 - \frac{\zeta_m^\phi(1-\xi_m^\phi)}{\Lambda_m^\phi}))\Gamma_m^\phi$$
$$+ \epsilon_m^\phi . (1 - \Gamma_m^\phi))](\zeta_m^\phi(2-\xi_m^\phi) + \varrho_m^\phi(1-\Lambda_m^\phi))$$

where:

$$\chi_m^\phi = P(H_m^\phi \mid R_m^\phi, E^\phi)$$
$$\varpi_m^\phi = P(H_m^\phi \mid \bar{R_m^\phi}, T_m^\phi, E^\phi)$$
$$\epsilon_m^\phi = P(H_m^\phi \mid \bar{T_m^\phi}, F_m^\phi, E^\phi)$$
$$\nu_m^\phi = P(H_m^\phi \mid E^\phi)$$
$$\Gamma_m^\phi = P(T_m^\phi \mid F_m^\phi, E^\phi)$$

The total weight of the cell can be found by summing the probability calculated by Equation (17) for all cells that can be occupied by the intruder aircraft and belong to region ϕ.

$$P(F_m^\phi \mid H_m^\phi, E^\phi) = W_m^{\phi^1} \tag{18}$$

Consider N^1 is the cardinality of the set $\phi^1 = (H_m \cap F_m)$. N^1 represents the number of cells in ϕ affected by the intrusion of the IA around UAV m. ϕ^1 is the set of cells with greater than zero probability of occupancy by an intruder aircraft in the next step. In Fig. 2, a schematic diagram showing ϕ^1 and N^1 has been presented.

Step 2: Calculation of probability of T_m being intruded by an IA

If the intruder aircraft enters region ϕ with collision possibility, the coefficient κ needs to be multiplied to the probability that causes the UAV to repel immediately from the intruder aircraft location. We can define a set ϕ^2 that represents subset of T_m which is in danger of intrusion by the IA. The cell weight update can be carried out as:

$$\forall \phi^2 = [T_m \cap \phi^1]$$
$$W_m^{\phi^2} = \kappa P(T_m^\phi \mid H_m^\phi, E^\phi) = \kappa \frac{P(H_m^\phi \mid T_m^\phi, E^\phi) P(T_m^\phi \mid E^\phi)}{P(H_m^\phi \mid E^\phi)} \tag{19}$$

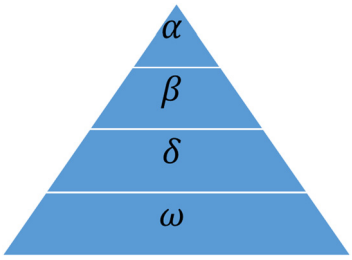

Fig. 3. Different grades of grey wolves.

To compute the above probability, at first $P(H_m^\phi \mid T_m^\phi, E^\phi)$ is considered.

$$\begin{aligned} P(H_m^\phi \mid T_m^\phi, E^\phi) &= P(H_m^\phi, R_m^\phi \mid T_m^\phi, E^\phi) \\ &+ P(H_m^\phi, \bar{R_m^\phi} \mid T_m^\phi, E^\phi) \\ &= P(H_m^\phi \mid R_m^\phi, T_m^\phi, E^\phi) P(R_m^\phi \mid T_m^\phi, E^\phi) \\ &+ P(H_m^\phi \mid \bar{R_m^\phi}, T_m^\phi, E^\phi) P(\bar{R_m^\phi} \mid T_m^\phi, E^\phi) \\ &= P(H_m^\phi \mid R_m^\phi, E^\phi) P(R_m^\phi \mid T_m^\phi, E^\phi) \\ &+ P(H_m^\phi \mid \bar{R_m^\phi}, T_m^\phi, E^\phi) P(\bar{R_m^\phi} \mid T_m^\phi, E^\phi) \\ &= \chi_m^\phi [1 - \frac{\zeta_m^\phi(1-\xi_m^\phi)}{\Lambda_m^\phi}] + \varpi_m^\phi \frac{\zeta_m^\phi(1-\xi_m^\phi)}{\Lambda_m^\phi} \end{aligned} \tag{20}$$

Hence Equation (19) can be written as:

$$W_m^{\phi^2} = \kappa P(T_m^\phi \mid H_m^\phi, E^\phi) \tag{21}$$
$$= \kappa \frac{[\chi_m^\phi [1 - \frac{\zeta_m^\phi(1-\xi_m^\phi)}{\Lambda_m^\phi}] + \varpi_m^\phi \frac{\zeta_m^\phi(1-\xi_m^\phi)}{\Lambda_m^\phi}][\zeta_m^\phi(1-\xi_m^\phi) + \xi_m^\phi]}{\nu_m^\phi}$$

Step 3: Calculation of probability of R_m being intruded by an IA

Let us define ϕ^3 as the subset of R_m that is intruded by an IA. The cell weight in the decision area of the UAV is computed as:

$$\forall \phi^3 \in [R_m \cap \phi^1]$$
$$W_m^{\phi^3} = \beta P(R_m^\phi \mid H_m^\phi, E^\phi) = \beta \frac{P(H_m^\phi \mid R_m^\phi, E^\phi) P(R_m^\phi \mid E^\phi)}{P(H_m^\phi \mid E^\phi)} \tag{22}$$

Equation (22) can be simplified to:

$$W_m^{\phi^3} = \beta \frac{\chi_m^\phi \xi_m^\phi}{\nu_m^\phi} \tag{23}$$

where β is a constant greater than one and more than κ to repel the UAV the area ϕ^3 more quickly as compared to ϕ^2.

The probability values calculated using the three steps are then added to the rank of the cells found in section 3 obtained via DBVF to obtain the overall weight of the cell. The equation for deriving the cell weight for a cell $i \in \phi$ for UAV m can be written as equation (24).

$$W_m^{i \in \phi} = W_m^{\phi^1} + W_m^{\phi^2} + W_m^{\phi^3} + DBVF \tag{24}$$

6. Grey Wolf Optimization (GWO) algorithm

GWO is inspired from the social collaborative behavior observed in the grey wolves during hunting. During the hunting, the population of wolves divides into sub-groups, referred to as grades. The hierarchy of the wolves can be seen in Fig. 3.

As it can be seen in Fig. 3, every grade of the grey wolves, α, β, δ, and ω represents a role in the optimization process. α is the

best solution, and is the outcome of the optimization process. Similarly, the second, third, and fourth fittest solutions are represented by β, δ and ω respectively.

The algorithm begins by considering a population of S wolves. The position of each member of the population is considered to be:

$$\mathbb{S} = [\mathbb{S}_1, ..., \mathbb{S}_S];$$

and the position of the sth wolf in N dimensional solution space can be considered as:

$$\mathbb{S}_s = [x_s^1, ..., x_s^n, ..., x_s^N]^T.$$

The predatory process for this behavior can be defined with a recursive relation given by:

$$x_s^n(k+1) = x_p^n(k) - A_s^n(k)|\mathbb{C}_s^n.x_p^n(k) - x_s^n(k)|. \tag{25}$$

In Equation (25), k represents current iteration and x_p^n is the position of the prey in nth dimension. The last term in the Equation (25) represents the simulation of encirclement of the prey by the wolves. A_s^n and $\mathbb{C}_s^n$ are represented by:

$$A_s^n = 2e.r_1 - e; \tag{26}$$

$$\mathbb{C}_s^n = 2.r_2; \tag{27}$$

in which r_1 and r_2 are two random numbers between zero and one. Variable e is decreasing linearly from 2 to 0 as a function of the iteration steps and is calculated by:

$$e = 2 - \frac{k}{k_{max}}. \tag{28}$$

k_{max} represents the maximum iteration. For every iteration, it is considered that the desired position of goal can be updated using the Equations (29)–(32).

$$x_{s,\alpha}^n(k+1) = x_\alpha^n(k) - A_{s,\alpha}^n(k)|\mathbb{C}_{s,\alpha}^n.x_\alpha^n(k) - x_s^n(k)|, \tag{29}$$

$$x_{s,\beta}^n(k+1) = x_\beta^n(k) - A_{s,\beta}^n(t)|\mathbb{C}_{s,\beta}^n.x_\beta^n(k) - x_s^n(k)|, \tag{30}$$

$$x_{s,\delta}^n(k+1) = x_\delta^n(k) - A_{s,\delta}^n(k)|\mathbb{C}_{s,\delta}^n.x_\delta^n(k) - x_s^n(k)|, \tag{31}$$

$$x_s^n(t+1) = \frac{1}{3}\sum_{j\in\{\alpha,\beta,\delta\}} x_{s,j}^n(t+1) \tag{32}$$

It may be noted that for the optimization problem considered in this paper, three top grades, α, β, and δ, are sufficient to provide an acceptable solution. Hence, grade ω has not been used in the optimization algorithm. For further discussion about the derivation of the equations, refer to [50]. The algorithm used for solving the path planning problem is presented as Algorithm 1 below. It is considered that the position of sth wolf is the solution of the optimization problem which becomes the control input of the UAV.

Algorithm 1: Pseudo code for path planning of single UAV.

```
while Next decision is not goal do
    Start by grey wolves pack as 𝕊 = [𝕊_1, ..., 𝕊_S];
    t = 1;
    search for a random viable path;
    for UAV m ∈ [1, ..., M] do
        while k ≤ k_max do
            while s < S do
                update the position of the wth wolf using equation (32);
                Update A and C;
                Update the α, β and γ
            end
            k = k + 1;
            evaluate J;
            Sort the case from best to worst J function
        end
        Output the optimal solution;
        Go to the next position; Feed the next position to quadrotor
        dynamics
    end
end
Post process and visualization
```

7. Quadrotor dynamics

7.1. Dynamic model

We present a dynamic model of a quadrotor UAV in this section. The coordinate systems and the free body diagram of the UAV are as shown in Fig. 4. Here, W denotes the fixed reference frame and B is the body frame which is attached to the quadrotor. F_i's are vertical forces from the rotors, and in addition to that, each rotor also produces a moment perpendicular to the plane of its rotation.

The Z–X–Y Euler angle convention is used to model the rotation of the quadrotor in the W frame. The transformation from W to

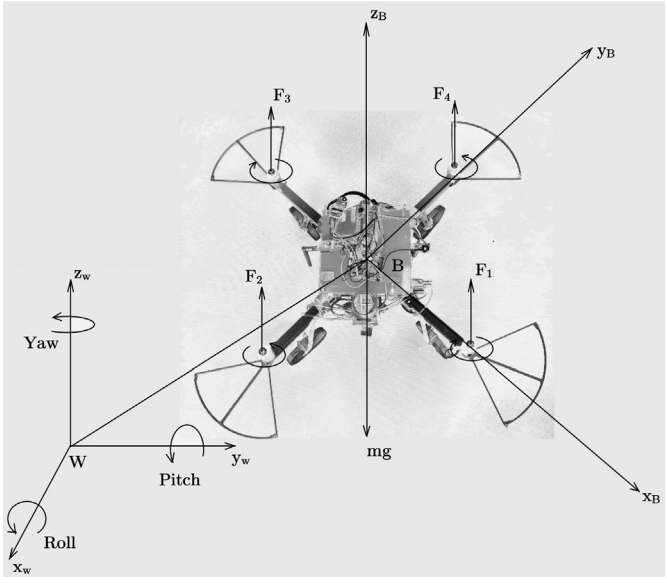

Fig. 4. The coordinate systems and the free body diagram.

B frame can be achieved by first rotating through z_W by yaw angle, ψ, then through the intermediate x-axis by the roll angle, ϕ, and then about the y_B by the pitch angle, θ. Hence, the rotation matrix R from W to B frame can be written as:

$$R = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\phi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \tag{33}$$

where $c(\bullet)$ and $s(\bullet)$ denote $cos(\bullet)$ and $sin(\bullet)$ respectively for each of the angles.

The equations of motion governing the acceleration of the center of mass can be written as follows:

$$m\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R\begin{bmatrix} 0 \\ 0 \\ \sum F_i \end{bmatrix} \tag{34}$$

where m is the mass of the quadrotor and g is the acceleration due to gravity.

7.2. Motor control

The rotor force F_i due to the rotation are given by:

$$F_i = k_F \omega_i^2 \tag{35}$$

where ω_i is the rotational speed of rotor i, and k_F is a constant whose value is taken from [56], [57] as $k_F = 6.11 \times 10^{-8}$ N/rpm^2. In addition, Euler equations are written in order to obtain angular accelerations of the vehicle given by:

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_1 - F_3) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{36}$$

where L is distance of each rotor from the vehicle's center of gravity. I is the matrix for moment of inertia along x_B, y_B, and z_B directions respectively. M_i $(i = 1, 2, 3, 4)$ are rotor moments produced by angular velocity of rotors and are given by:

$$M_i = k_M \omega_i^2 \tag{37}$$

where ω_i is the angular velocity of ith rotor and k_M is the constant, taken from [56] as $k_M = 1.5 \times 10^{-9}$ Nm/rpm^2.

For hover state, the thrusts from the propellers must balance the weight of the vehicle, hence

$$F_{i,0} = \frac{mg}{4} \tag{38}$$

and the motor speeds are given by

$$\omega_{i,0} = \omega_H = \sqrt{\frac{mg}{4k_F}} \tag{39}$$

7.3. Attitude control

The controller inputs are the speeds of the four propellers. The dynamic model of the quadcopter, described in (34) and (36), is used to design a PD controller.

Substituting (35) and (37) in (36) and assuming that the component of angular velocity in z_B direction is small compared to other terms, we get the vector of desired rotor speeds as linear combination of four terms:

$$\begin{bmatrix} \omega_1^{des} \\ \omega_2^{des} \\ \omega_3^{des} \\ \omega_4^{des} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \omega_H + \Delta\omega_F \\ \Delta\omega_\phi \\ \Delta\omega_\theta \\ \Delta\omega_\psi \end{bmatrix} \tag{40}$$

where ω_i^{des}, $(i = 1, 2, 3, 4)$ are the desired angular velocities of the respective rotors, and ω_H is hovering speed. The proportional-derivative control laws are used to control $\Delta\omega_\phi$, $\Delta\omega_\theta$, $\Delta\omega_\psi$, and $\Delta\omega_F$, which are deviations that result into forces/moments causing roll, pitch, yaw, and a net force along the z_B axis, respectively, and are calculated as:

$$\begin{aligned} \Delta\omega_\phi &= k_{p,\phi}(\phi^{des} - \phi) + k_{d,\phi}(p^{des} - p) \\ \Delta\omega_\theta &= k_{p,\theta}(\theta^{des} - \theta) + k_{d,\theta}(q^{des} - q) \\ \Delta\omega_\psi &= k_{p,\psi}(\psi^{des} - \psi) + k_{d,\psi}(r^{des} - r) \\ \Delta\omega_F &= \frac{m}{8k_F\omega_H}\ddot{Z}^{des} \end{aligned} \tag{41}$$

where p, q, and r are the components of angular velocities of the vehicle in the body frame. The relationship between these components and the pitch, roll, and yaw are provided in [58].

Table 1
Quadrotor specifications.

Parameter	Value	Units
I_{xx}	2.32×10^{-3}	kg/m^2
I_{yy}	2.32×10^{-3}	kg/m^2
I_{zz}	4.00×10^{-3}	kg/m^2
m	0.5	kg
L	0.175	m
k_F	6.11×10^{-8}	N/rpm^2
k_M	1.5×10^{-9}	N/rpm^2

7.4. Position control

In order to have the quadrotor track a desired trajectory $r_{i,T}$, the command acceleration, $\ddot{r}_i^{des}$ is calculated from proportional-derivative controller based on position error, as [59]:

$$(\ddot{r}_{i,T} - \ddot{r}_i^{des}) + k_{d,i}(\dot{r}_{i,T} - \dot{r}_i) + k_{p,i}(r_{i,T} - r_i) = 0 \tag{42}$$

where r_i and $r_{i,T}$ $(i = 1, 2, 3)$ are the 3-dimensional position of the quadrotor and desired trajectory respectively. It may be noted that $\dot{r}_{i,T} = \ddot{r}_{i,T} = 0$ for hover.

During the flight, the orientation of the vehicle needs to be set close to zero. This can be achieved by linearizing the equation of motion that correspond to the nominal hover states. The nominal hover state is ($r = r_0, \phi = \theta = 0, \psi = \psi_T, \dot{r} = 0$ and $\dot{\theta} = \dot{\psi} = \dot{\phi} = 0$). The change of the pitch and roll angles are supposed to be small during flight. By linearizing equation (34) about these nominal hovering states, desired pitch and roll angles to cause the motion can be derived as given by the following equations:

$$\begin{aligned} \phi^{des} &= \frac{1}{g}\left(\ddot{X}^{des} sin\psi_T - \ddot{Y}^{des} cos\psi_T\right) \\ \theta^{des} &= \frac{1}{g}\left(\ddot{X}^{des} cos\psi_T + \ddot{Y}^{des} sin\psi_T\right) \end{aligned} \tag{43}$$

where $\ddot{X}^{des}$ and $\ddot{Y}^{des}$ are the desired accelerations in X and Y directions respectively. ψ_T is the yaw angle to be tracked which is the same as the desired yaw angle ψ^{des}.

Table 1 represents the specifications of the Hummingbird quadrotor platform used in this paper [56].

8. Flight simulation and results

In this section, we consider multiple quadrotors, $m \in [1, ..., M]$, flying to reach the goal position G.

It is assumed that the intruder aircraft are either broadcasting their position to the nearby UAV (via ADS-B) or they are recognized by ground-based radar continuously without interruptions. Furthermore, without loss of generality, it is assumed that the UAV could occupy any one of the neighboring cells in the next time step t. By this assumption:

$$T_m^\phi = R_m^\phi$$

Several flight scenarios are designed and simulated to demonstrate the performance of the proposed method. In all of the scenarios, the UAV tries to reach the goal position using the shortest and safest path obtained via the GWO. The area is tessellated using rectangular cells of unit size 1×1, and the total number cells in the entire simulation area is 2500. F_ϕ covers the range of six cells around the UAV. Here, we provide results from two scenarios. Also, for avoiding corners of obstacles, one-cell safety margin around each obstacle has been considered [corner cutting]. During the simulations, there are some obstacles which behave as pop-up threat. Pop-up threats are those that the UAV is not aware of

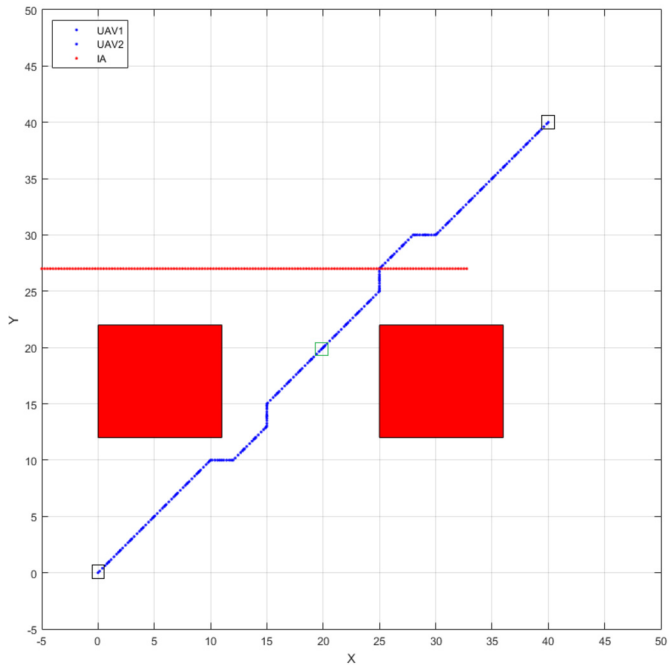

Fig. 5. Result of the simulation for trajectory planning of the UAVs in presence of an IA and pop-up obstacles.

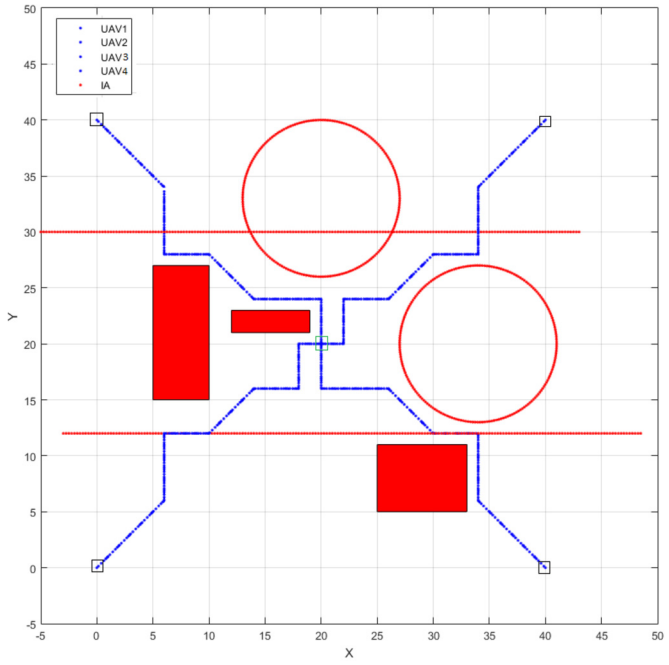

Fig. 6. Result of the simulation for trajectory planning of the UAVs in presence of multiple IA and pop-up obstacles.

a-priori but gets informed after getting close to them. Furthermore, the UAV is constrained to go within the tessellated area only.

During each of the scenarios, two set of obstacles have been introduced. Pop-up obstacles refer to the obstacles which the UAV is not aware of at first and are recognized only using on-board sensors. This is simulated by letting the UAV recognize pop-up obstacles when it reaches within certain distance from the obstacle. *A-Priori* obstacles, on the other hand, refer to those obstacles that the UAV is aware of before the beginning of its mission such as structures and no-fly zones.

8.1. Cooperative flight in presence of pop-up obstacles and one IA

In this Scenario, the two UAVs start from cell $(0, 0)$ and $(40, 40)$ and try to reach at the goal at $(20, 20)$. There are two obstacles in this area *not* defined *a-priori* (called pop-up threats). There is another aerial vehicle flying *non-cooperatively* (called intruder aircraft or IA) in the environment. The positions of the UAVs at different instants of time and the trajectory of the UAVs are shown in Fig. 5. A video showing the motion of the UAVs can be seen at[1].

8.2. Cooperative flight of four UAVs in presence of four IAs and multiple fixed obstacles

The purpose of this scenario is to increase the complexity of the problem and test the robustness of the proposed algorithm. There are four UAVs defined for this problem starting from $(0, 0)$, $(0, 40)$, $(40, 0)$, and $(40, 40)$. The goal is placed at $(20, 20)$. The obstacles are not defined *a-priori* to the UAVs. The scheme of the solution is shown in Fig. 6 and the video of the solution can be found in[2].

9. Scalability of the solution

In Section 8, the performance of GWO to solve the trajectory planning problems is studied from two criteria: ability to obtain the globally optimal solution and time required for obtaining the solution. While this method presents promising results, it is important to show the ability of the proposed solution to deal with large and complex problems.

Fig. 7 represents the solution obtained using the proposed GWO for the trajectory planning of 10 UAVs. The green area in the figure represents the goal position. The goal position needs to be considered bigger to support a large number of the UAVs, entering that area. The result shows that the solution methodology is well capable of finding the solution for larger-scale problems. The motion of the UAVs can be seen in the video located at[3].

Fig. 7 demonstrates another significant aspect about the behavior of cooperation between the UAVs. The UAVs sacrifice their respective cost function to reach to the goal position at the same time. Further details can be illustrated in Fig. 8.

As is clear from Fig. 8, some of the UAVs end up increasing their individual cost function during a short period of time to reach the goal position at the same time as others. But, as is clear from Fig. 9, the total cost function of the entire fleet is monotonically decreasing over the time.

The solution of this problem is compared with the exact solution found using Mixed Integer Linear Programming (MILP). The cost function and formulations for solving this problem in MILP frame work can be seen using Equations (44) and (45) to (47), respectively.

$$J = \sum_{m=1}^{M} \sum_{j=1}^{Q} \sum_{i=1}^{Q} \mathring{\mathbb{J}}_{ij,e} w_{ij,e} \tag{44}$$

$$\forall m \in \{1, \ldots, M\}$$

$$\sum_{i=1}^{Q} \mathring{\mathbb{J}}_{ij_{final},e} = 1 \tag{45}$$

$$1 \leq i, j \leq Q; \tag{46}$$

$$W_{ii,m} = M_{big}; \tag{47}$$

[1] https://youtu.be/c2PhyAf3ua8.
[2] https://youtu.be/9H-n5-ia3cI.
[3] https://youtu.be/kAz26KVmth8.

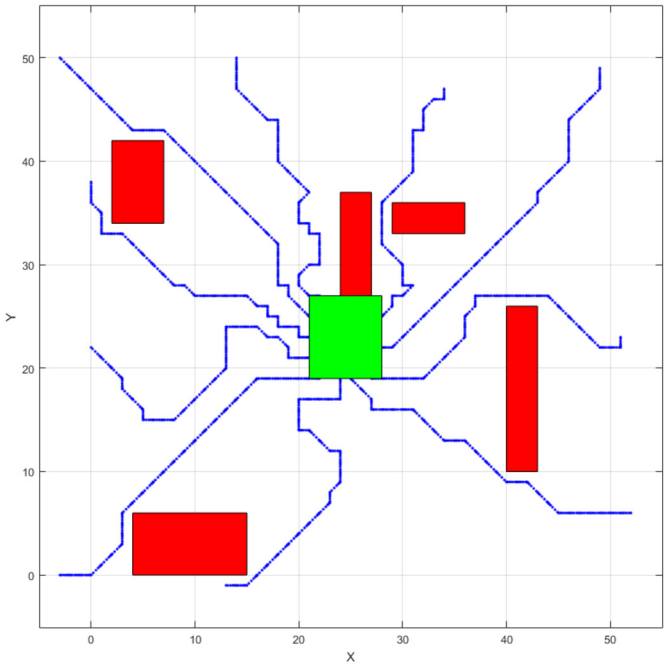

Fig. 7. Result of the simulation for trajectory planning of 10 UAVs in presence of pop-up obstacles.

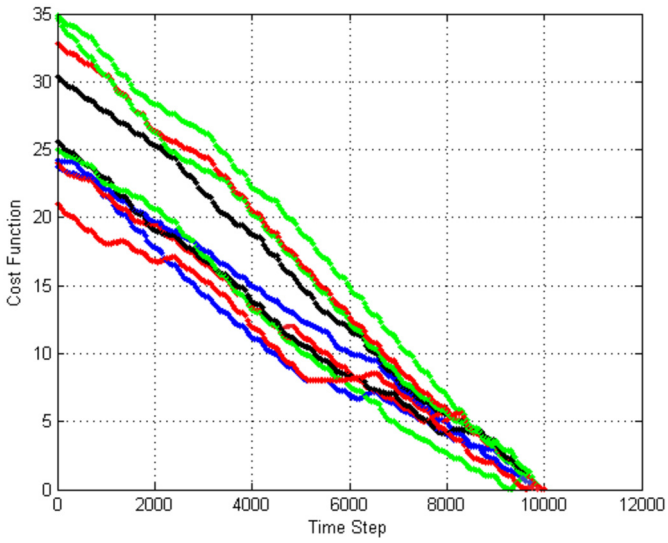

Fig. 8. Cost function of 10 UAVs shown in Fig. 7.

where $w_{ij,m}$ is the cost function of traveling from cell i to cell j for UAV m. And $\mathbb{j}_{ij,m}$ is a binary variable equal to one if UAV e travels from cell i to j along its tour and zero otherwise.

Another bio-mimetic algorithm that can solve this problem is Genetic Algorithm (GA) [6]. The algorithm based on this approach for solving this problem is written in Algorithm 3.

In Algorithm 3, all the cells with the possibility where the UAVs could travel through are considered to have binary values. During the GA based optimization process, if the UAV plans to travel to a certain cell, its value considered as one, and if the UAV does not plan to travel there, its value is considered as zero. Best fitness could be found by minimizing J function and upon creating a logical path.

In the GWO algorithm, it can be seen that the complexity of the solution depends on these three factors:

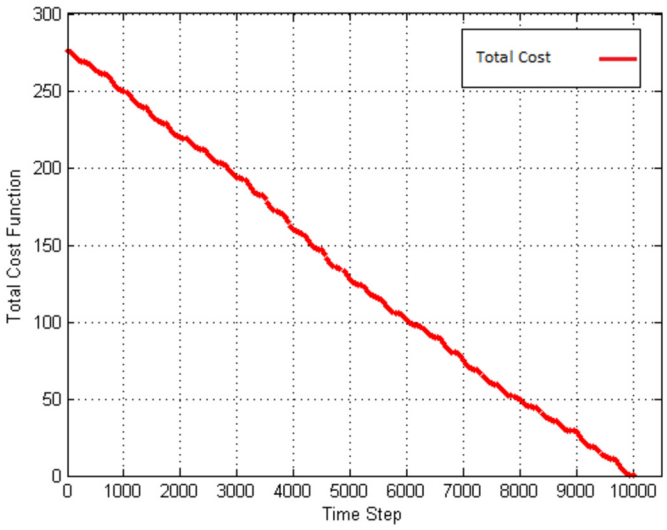

Fig. 9. Total cost function of the problem for 10 UAVs shown in Fig. 7.

Algorithm 2: Overview of GA used to solve the optimization problem.

```
t ← 0;
Evaluate population(gen);
while Next decision is not goal do
    gen ← 0;
    generate a random path;
    while Terminal conditions are not satisfied do
        Select population(gen) from population(gen−1);
        Crossover population(gen);
        Mutate population(gen);
        Evaluate the j;
        Find the best fit;
    end
end
```

- Number of UAVs flying in the area,
- Number of cells in the flight area,
- Number of IAs flying in the area

Hence in Sections 9.1 and 9.2, all these aspects of the complexity are tested to find the difference between the GA and GWO algorithm.

9.1. Optimality

The results of the GWO solution for the scenario mentioned in case of Fig. 7 are compared with the MILP solution (which provides the exact/optimal solution). Since GWO has built-in randomness, the results are obtained for 160 times of run. As can be seen in Fig. 10, for more than 95% of the cases, GWO represents the optimal or less than 2% error in the cost function value of the obtained solution. For the same problem, as it can be seen in Fig. 11, GA results in only 90% accuracy in the optimality of the solution.

In another case, for the same scenario mentioned in Fig. 7, the size of the cells are decreased from 1×1 to 0.5×0.5 and 0.25×0.25. Hence the number of cells increase from 2500 to 10000 and 40000 respectively. It can be seen that the computation time for the GWO as well as the GA increases as the numbers of cells increase (signifying larger number of decision variables).

Both GA and GWO are run 20 times for each of three cases of numbers of cells and the average results are reported in Figs. 12 and 13. Here, it can be seen in Fig. 12 that the GWO shows slower increase in computation time as compared to the GA when number of cells increase. Fig. 13 represents the comparison between the cost function of the GA and GWO.

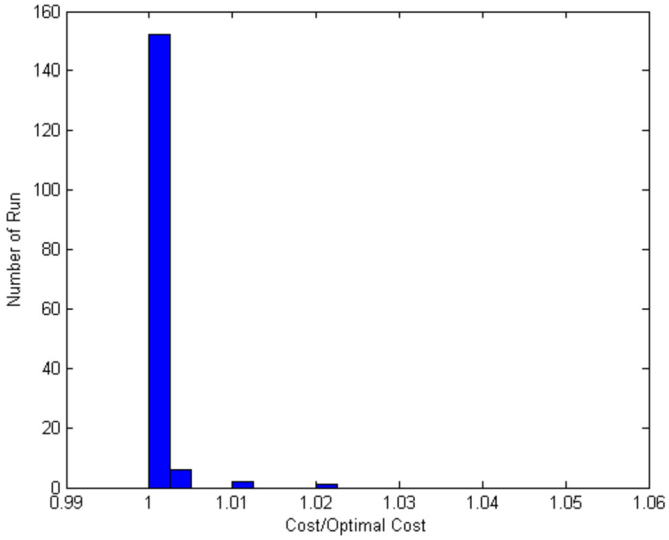

Fig. 10. Performance of GWO for 160 run for the scenario mentioned in Fig. 7.

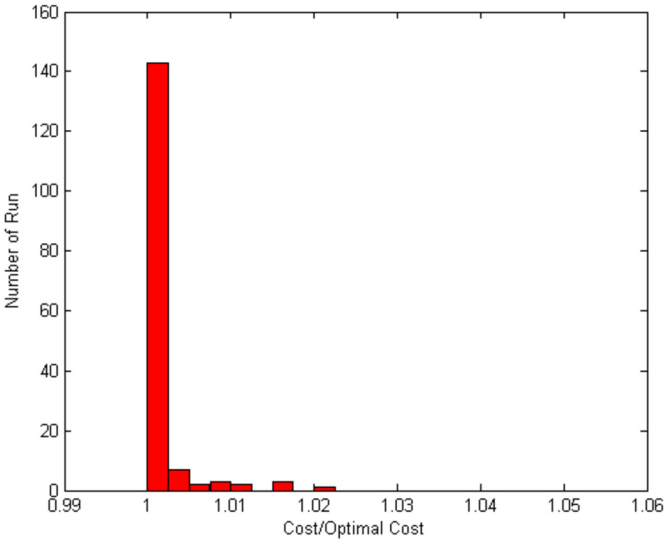

Fig. 11. Performance of GA for 160 run for the scenario mentioned in Fig. 7.

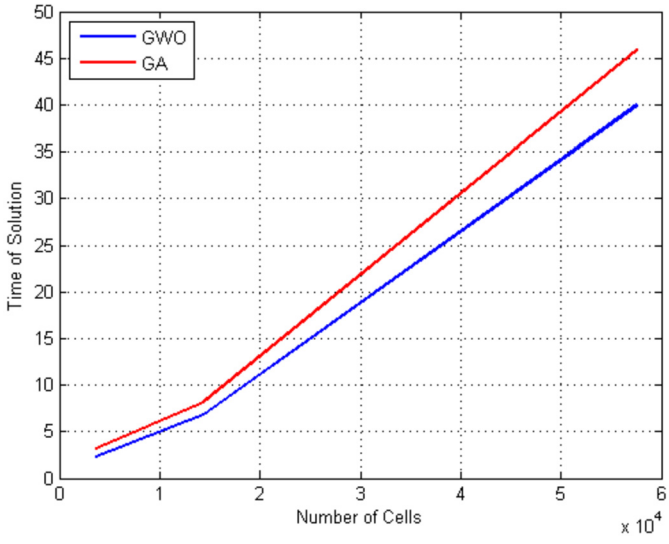

Fig. 12. Effect of number of cells on the computation time for 20 runs of GA and GWO for the scenario mentioned in Fig. 7.

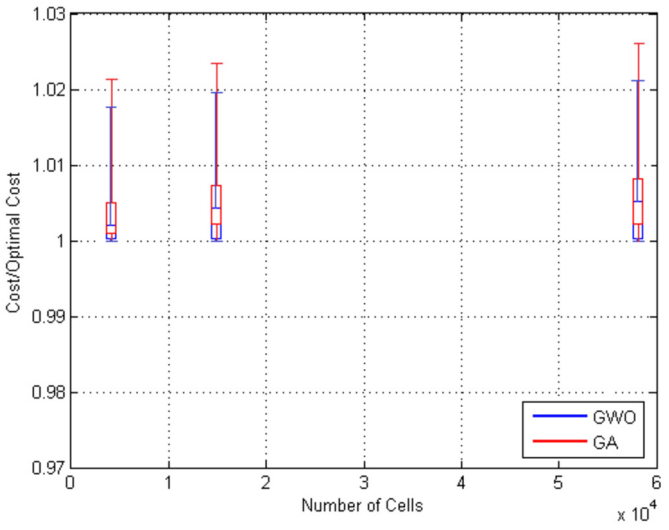

Fig. 13. Effect of number of cells on the cost function for 20 runs of GA and GWO for the scenario mentioned in Fig. 7.

9.2. Computational time requirement

The aim of this section is to compare the computation time for multiple scenarios to understand the performance of the GWO as compared to GA and MILP. In real-time flight scenarios, it is important to implement the less computationally intensive algorithms. Considering that for larger problems, it is impossible to enumerate all the possible solutions, so an approximate solution has to be used. Here, 10 different scenarios with different number of UAVs in order of increasing complexity of the problem have been generated. The results shown in Table 2 denote the average of 25 runs of each algorithm. For each scenario, the following parameter is calculated:

– Normalized cost = The cost calculated by GWO or GA divided by the exact cost found by MILP.

Table 2 shows the difference between the behavior of different algorithms in achieving the most optimized solution for the problem. The notation used in Table 2 for enumerating the cases, the first three characters represent the number of obstacles, the next three characters represent the number of IAs and the last four characters represent number of UAVs in the area. For instance "O02I03U004" means there are two fixed obstacles, three IAs and four UAVs. The goal is located at the middle of 400×400 grid area. Starting position of the UAVs are places on the perimeter of the outside rectangle of the tessellated area. Third and fourth columns of Table 2 represents the average normalized cost, obtained from multiple runs for the same scenario using the GWO and GA respectively. Optimized cost calculated from the MILP solution is represented in the fifth column of this table. Sixth and seventh columns represent the time consumption for solving the problem using GWO and GA algorithm in seconds, respectively. In this paper the simulations are carried out on a computer with the following specifications:

1. Operating System: Windows 10 Enterprise 64-bit
2. Processor: Intel(R) Core(TM)i7 -2500 CPU @ 3.30 GHz
3. (2 CPUs)
4. Memory: 8.00 GB RAM
5. Programming language: MATLAB R2014a

Table 2
Overview of solution characteristics for each method for different scenarios.

Mission identity		Normalized average cost		Best cost	Time consumption (s)$\times 10^{-1}$	
Case	UAVs	GWO	GA	MILP	GWO	GA
O02I01U002	2	1.002	1.006	1467.32	87.2452	98.2634
O03I03U003	3	1.002	1.006	2167.41	91.9874	103.3647
O05I05U003	3	1.003	1.007	2191.42	92.3561	105.3268
O02I06U007	7	1.003	1.008	43235.79	103.3561	109.3256
O04I10U007	7	1.004	1.008	4376.14	106.7465	110.2547
O04I10U025	25	1.004	1.008	13514.25	145.2536	167.3256
O02I05U050	50	1.006	1.011	27861.94	198.2351	203.9851
O05I06U100	100	1.009	1.013	53257.21	286.9851	310.2531
O04I10U150	150	1.011	1.016	81422.32	423.2531	471.4039
O05I10U200	200	1.014	1.019	12985.70	642.2102	715.7809

10. Conclusions

This paper proposes a novel approach that utilizes the formulation of dynamic Bayesian, DBVF, and GWO to solve the problem of path planning and collision avoidance for UAVs in presence of fixed and moving obstacles in an uncertain environment. A probabilistic framework for solving the problem is utilized to represent the uncertainties arising out of unavailability of trajectories of the intruder aircraft. Upon intrusion by an intruder aircraft in the region close to the UAV, cells around the UAV would change their weights based on the probability values representing the risks of collision. The path planning problem is then formulated as an optimization problem with obstacles (both stationary and dynamic ones) incorporated as constraints. The cost function used for the optimization comprises of two components: total distance of flight and a safety metric modeled as weights of cells representing the risk of collision. The numerical results obtained from several scenarios show the effectiveness of the proposed approach to carry out the path planning for different flight missions as compared to two other algorithms (GA and MILP) used in literature.

Conflict of interest statement

The authors declare that there is no conflict of interest regarding the publication of this paper.

Appendix A

A.1. Implementation of GA

A GA path-planning solution represents a feasible shortest collision-free path, i.e., no cells on the path belong to the obstacle space, or none of the line segments of the path intersects an obstacle. The length of a chromosome fluctuates between 2 and a maximum length N_{max}. After generating the path, the GA evaluates the path for feasibility.

Algorithm 3: GA overview.

```
gen ← 0;
Evaluate population(gen);
while termination conditions are not satisfied do
    gen ← gen+1;
    Select population(gen) from population(gen−1);
    Crossover population(gen);
    Mutate population(gen);
    Evaluate population(gen);
end
```

For implementing GA, this paper considers a binary array with the length of $1 \times$ **Number of Cells** as a chromosome and each chromosome represents a complete path. The cells in the tessellated area have been already labeled and addressed in the array. If a cell is a part of a particular path, then the binary variable corresponding to that cell in the chromosome is assigned a value of 1. Otherwise, it is assigned a value of 0. A schematic diagram showing a particular path and the corresponding chromosome is shown in Fig. 14. The rest of the implementation details are provided in Algorithm 3.

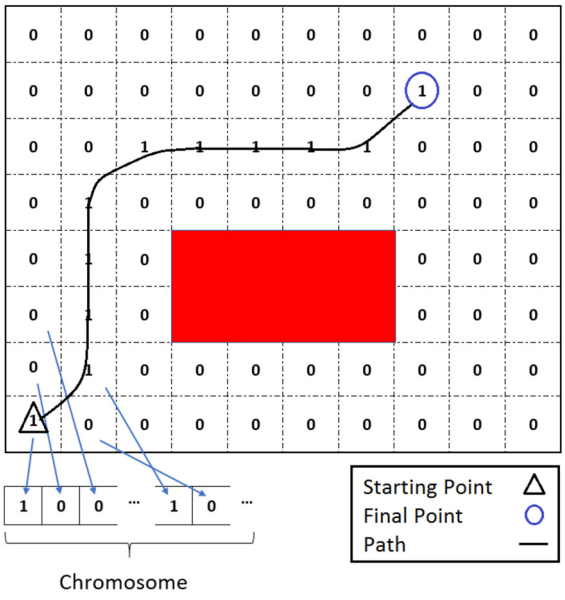

Fig. 14. Chromosome scheme for the GA.

References

[1] M. Radmanesh, O. Nematollahi, M. Nili-Ahmadabadi, M. Hassanalian, A novel strategy for designing and manufacturing a fixed wing MAV for the purpose of increasing maneuverability and stability in longitudinal axis, J. Appl. Fluid Mech. 7 (3) (2014) 435–446.
[2] R. Holdsworth, Autonomous In-Flight Path Planning to Replace Pure Collision Avoidance for Free Flight Aircraft Using Automatic Dependent Surveillance Broadcast, PhD thesis, Swinburne University, 2003.
[3] I. Ray, ADS integration into the flight management computer, in: FAA, the First Annual International Satellite Surveillance and Communication Symposium, 1991, pp. 277–280 (SEE N 91-32093 24-04).
[4] T.S. Perry, In search of the future of air traffic control, IEEE Spectr. 34 (8) (1997) 18–35.
[5] M. Radmanesh, M. Kumar, Grey wolf optimization based sense and avoid algorithm for UAV path planning in uncertain environment using a bayesian framework, in: 2016 International Conference on Unmanned Aircraft Systems, ICUAS, IEEE, 2016, pp. 68–76.
[6] M. Radmanesh, UAV Traffic Management for National Airspace Integration, M.S thesis, University of Cincinnati, 2016.
[7] A.E. Smith, C. Evers, Method and apparatus for improving utility of automatic dependent surveillance, Oct. 14 2003. US Patent 6,633,259.
[8] R. Asep, A. Achaibou, F. Mora-Camino, Automatic collision avoidance based on supervised predictive controllers, Control Eng. Pract. 4 (8) (1996) 1169–1175.

[9] K. Mostov, A. Soloviev, Fuzzy adaptive stabilization of higher order Kalman filters in application to precision kinematic GPS, in: Proceedings of the 9th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 1996), 1996, pp. 1451–1456.
[10] A. Miura, H. Morikawa, M. Mizumachi, Aircraft collision avoidance with potential gradient—ground-based avoidance for horizontal maneuvers, Electron. Commun. Jpn., Part 3 Fundam. Electron. Sci. 78 (10) (1995) 104–114.
[11] D. Reichardt, J. Shick, Collision avoidance in dynamic environments applied to autonomous vehicle guidance on the motorway, in: Proceedings of the Intelligent Vehicles' 94 Symposium, IEEE, 1994, pp. 74–78.
[12] C.J. Tomlin, J. Lygeros, S.S. Sastry, A game theoretic approach to controller design for hybrid systems, Proc. IEEE 88 (7) (2000) 949–970.
[13] M. Radmanesh, M. Kumar, A. Nemati, M. Sarim, Dynamic optimal UAV trajectory planning in the national airspace system via mixed integer linear programming, Proc. Inst. Mech. Eng. G J. Aerosp. Eng. 230 (9) (2016) 1668–1682.
[14] M. Radmanesh, M. Kumar, Flight formation of UAVs in presence of moving obstacles using fast-dynamic mixed integer linear programming, Aerosp. Sci. Technol. (2015).
[15] L.E. Chiang, 3-D CNC trajectory interpolation using Bresenham's algorithm, in: Proceedings of the IEEE International Symposium on Industrial Electronics, ISIE'94, IEEE, 1994, pp. 264–268.
[16] S.A. Velastin, C. Xu, Line and circle finding by the weighted Mahalanobis distance transform and extended Kalman filtering, in: Proceedings of the IEEE International Symposium on Industrial Electronics, ISIE'94, IEEE, 1994, pp. 258–263.
[17] P. An, C. Harris, R. Tribe, N. Clarke, Aspects of neural networks in intelligent collision avoidance systems for Prometheus, 1993.
[18] J.C. Clements, The optimal control of collision avoidance trajectories in air traffic management, Transp. Res., Part B, Methodol. 33 (4) (1999) 265–280.
[19] Y. Kim, D.-W. Gu, I. Postlethwaite, Real-time optimal mission scheduling and flight path selection, IEEE Trans. Autom. Control 52 (6) (2007) 1119–1123.
[20] M. Alighanbari, Y. Kuwata, J.P. How, Coordination and control of multiple UAVs with timing constraints and loitering, in: Proceedings of the 2003 American Control Conference, vol. 6, IEEE, 2003, pp. 5311–5316.
[21] D.B. Kingston, C.J. Schumacher, Time-dependent cooperative assignment, in: Proceedings of the 2005 American Control Conference, IEEE, 2005, pp. 4084–4089.
[22] D.P. Garg, M. Kumar, Optimization techniques applied to multiple manipulators for path planning and torque minimization, Eng. Appl. Artif. Intell. 15 (3) (2002) 241–252.
[23] S. Tennety, S. Sarkar, E.L. Hall, M. Kumar, Support vector machines based mobile robot path planning in an unknown environment, in: ASME 2009 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, 2009, pp. 395–401.
[24] S. Sarkar, E.L. Hall, M. Kumar, Mobile robot path planning using support vector machines, in: ASME 2008 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, 2008, pp. 709–715.
[25] W. Blake, D. Multhopp, Design, performance and modeling considerations for close formation flight, CLJ 150 (1998) 2.
[26] D.F. Chichka, J.L. Speyer, Solar-Powered, Formation-Enhanced Aerial Vehicle Systems for Sustained Endurance, Proceedings of the 1998 American Control Conference, vol. 2, IEEE, 1998, pp. 684–688.
[27] M. Radmanesh, M. Kumar, P.H. Guentert, M. Sarim, Overview of path planning and obstacle avoidance algorithms for UAVs: a comparative study, in: Unmanned Systems, 2018.
[28] G. Hattenberger, R. Alami, S. Lacroix, Planning and control for Unmanned Air Vehicle formation flight, in: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2006, pp. 5931–5936.
[29] N. Faiz, S.K. Agrawal, R.M. Murray, Trajectory planning of differentially flat systems with dynamics and inequalities, J. Guid. Control Dyn. 24 (2) (2001) 219–227.
[30] O.A. Yakimenko, Direct method for rapid prototyping of near-optimal aircraft trajectories, J. Guid. Control Dyn. 23 (5) (2000) 865–875.
[31] G. Yang, V. Kapila, Optimal path planning for Unmanned Air Vehicles with kinematic and tactical constraints, in: Proceedings of the 41st IEEE Conference on Decision and Control, 2002, vol. 2, IEEE, 2002, pp. 1301–1306.
[32] L.E. Buzogany, M. Pachter, J. d'Azzo, Automated control of aircraft in formation flight, in: Proc. AIAA Guidance, Navigation, and Control Conference, 1993, pp. 1349–1370.
[33] C. Schumacher, S.N. Singh, Nonlinear control of multiple UAVs in close-coupled formation flight, AIAA paper 4373 (2000) 2000.
[34] T. Balch, R.C. Arkin, Behavior-based formation control for multirobot teams, IEEE Trans. Robot. Autom. 14 (6) (1998) 926–939.
[35] F.-L. Lian, R. Murray, Real-time trajectory generation for the cooperative path planning of multi-vehicle systems, in: Proceedings of the 41st IEEE Conference on Decision and Control, 2002, vol. 4, IEEE, 2002, pp. 3766–3769.
[36] R.L. Raffard, C.J. Tomlin, S.P. Boyd, Distributed optimization for cooperative agents: application to formation flight, in: 43rd IEEE Conference on Decision and Control CDC, vol. 3, IEEE, 2004, pp. 2453–2459.
[37] S. Zelinski, T.J. Koo, S. Sastry, Hybrid system design for formations of autonomous vehicles, in: Proceedings of the 42nd IEEE Conference on Decision and Control, vol. 1, IEEE, 2003, pp. 1–6.
[38] A. Richards, J. Bellingham, M. Tillerson, J. How, Coordination and control of multiple UAVs, in: AIAA Guidance, Navigation, and Control Conference, Monterey, CA, 2002.
[39] E.J. Forsmo, E. Grøtli, T. Fossen, T.A. Johansen, et al., Optimal search mission with Unmanned Aerial Vehicles using mixed integer linear programming, in: 2013 International Conference on Unmanned Aircraft Systems, ICUAS, IEEE, 2013, pp. 253–259.
[40] J.T. Feddema, C. Lewis, D. Schoenwald, et al., Decentralized control of cooperative robotic vehicles: theory and application, IEEE Trans. Robot. Autom. 18 (5) (2002) 852–864.
[41] B. Bouilly, T. Simeon, R. Alami, A numerical technique for planning motion strategies of a mobile robot in presence of uncertainty, in: Proceedings of the 1995 IEEE International Conference on Robotics and Automation, vol. 2, IEEE, 1995, pp. 1327–1332.
[42] L.A. Page, A.C. Sanderson, A path-space search algorithm for motion planning with uncertainties, in: Proceedings of the IEEE International Symposium on Assembly and Task Planning 1995, IEEE, 1995, pp. 334–340.
[43] A. Lambert, D. Gruyer, Safe path planning in an uncertain-configuration space, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2003, ICRA'03, vol. 3, IEEE, 2003, pp. 4185–4190.
[44] R. Alterovitz, T. Siméon, K.Y. Goldberg, The stochastic motion roadmap: a sampling framework for planning with Markov motion uncertainty, in: Robotics: Science and Systems, vol. 3, 2007, pp. 233–241, Citeseer.
[45] J.P. Gonzalez, A. Stentz, Planning with uncertainty in position an optimal and efficient planner, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2005, IEEE, 2005, pp. 2435–2442.
[46] M. Flint, E. Fernández, M. Polycarpou, Efficient bayesian methods for updating and storing uncertain search information for UAVs, in: 43rd IEEE Conference on Decision and Control, 2004 CDC, vol. 1, IEEE, 2004, pp. 1093–1098.
[47] T.A. Wilkin, A.E. Nicholson, Efficient inference in dynamic belief networks with variable temporal resolution, in: PRICAI 2000 Topics in Artificial Intelligence, Springer, 2000, pp. 264–274.
[48] B.J. Capozzi, Evolution-Based Path Planning and Management for Autonomous Vehicles, PhD thesis, University of Washington, 2001.
[49] A. Censi, D. Calisi, A. De Luca, G. Oriolo, A bayesian framework for optimal motion planning with uncertainty, in: IEEE International Conference on Robotics and Automation, ICRA 2008, IEEE, 2008, pp. 1798–1805.
[50] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[51] S. Mirjalili, How effective is the grey wolf optimizer in training multi-layer perceptrons, Appl. Intell. 43 (1) (2015) 150–161.
[52] Y.T.K. Priyanto, L. Hendarwin, Multi objective optimal power flow to minimize losses and carbon emission using wolf algorithm, in: 2015 International Seminar on Intelligent Technology and Its Applications, ISITIA, IEEE, 2015, pp. 153–158.
[53] N. Jayakumar, S. Subramanian, S. Ganesan, E. Elanchezhian, Grey wolf optimization for combined heat and power dispatch with cogeneration systems, Int. J. Electr. Power Energy Syst. 74 (2016) 252–264.
[54] S. Zhang, Y. Zhou, Z. Li, W. Pan, Grey wolf optimizer for unmanned combat aerial vehicle path planning, Adv. Eng. Softw. 99 (2016) 121–136.
[55] P. Yao, H. Wang, H. Ji, Multi-UAVS tracking target in urban environment by model predictive control and improved grey wolf optimizer, Aerosp. Sci. Technol. 55 (2016) 131–143.
[56] C. Powers, D. Mellinger, V. Kumar, Quadrotor kinematics and dynamics, in: Handbook of Unmanned Aerial Vehicles, Springer, 2015, pp. 307–328.
[57] M. Sarim, A. Nemati, M. Kumar, K. Cohen, Extended Kalman filter based quadrotor state estimation based on asynchronous multisensor data, in: ASME 2015 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, 2015, V001T06A008–V001T06A008.
[58] A. Nemati, M. Kumar, Modeling and control of a single axis tilting quadcopter, in: American Control Conference, ACC 2014, IEEE, 2014, pp. 3077–3082.
[59] R. Tan, M. Kumar, Proportional navigation (pn) based tracking of ground targets by quadrotor UAVs, in: ASME 2013 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, 2013, V001T01A004–V001T01A004.