IETE Technical Review

Taylor & Francis
Taylor & Francis Group

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/titr20

Path Planning for Multiple Targets Interception by the Swarm of UAVs based on Swarm Intelligence Algorithms: A Review

Abhishek Sharma, Shraga Shoval, Abhinav Sharma & Jitendra Kumar Pandey

IETE TECHNICAL REVIEW
https://doi.org/10.1080/02564602.2021.1894250

REVIEW ARTICLE

Path Planning for Multiple Targets Interception by the Swarm of UAVs based on Swarm Intelligence Algorithms: A Review

Abhishek Sharma[1], Shraga Shoval[2], Abhinav Sharma[3] and Jitendra Kumar Pandey[4]

[1]Research and Development Department, School of Engineering, University of Petroleum & Energy Studies, Bidholi, Dehradun, 248007, India; [2]Department of Industrial Engineering & Management, Ariel University, Ariel, 40700, Kiryat hamada, Israel; [3]Department of Electrical and Electronics Engineering, School of Engineering, University of Petroleum and Energy Studies, Bidholi, Dehradun, 248007, India; [4]School of Basic and Applied Sciences, Adamas University, Kolkata, 700126, West Bengal, India

ABSTRACT
The dramatic increase in the capabilities and availability of autonomous ground and aerial tools introduces safety and security challenges, particularly in protecting strategic infrastructures. In this context, the interception of multiple mobile threats, aiming to invade restricted spaces of such infrastructures is an important topic. This paper focuses on the problem of path planning for intercepting multiple aerial targets by a swarm of UAVs. 3D path planning for interception of moving targets is a challenging task, in particular when the interception is performed by a swarm of UAVs, as there are multiple kinematic and dynamic constraints. The aim is first to allocate targets to the individual UAVs (task assignment) and to construct a 3D path for each one. Many algorithms have been recognized as noble schemes for solving this kind of problems based on Swarm Intelligence (SI), many of them are based on biological systems such as particle swarm optimization (PSO), ant colony optimization (ACO), artificial bee colony optimization (ABC), bat-inspired algorithm (BA), etc. The paper presents a comprehensive review of SI algorithms centered on the problems related to 3D path planning for target interception by a swarm of UAVs. It also focuses on the improvement of existing SI algorithms for better path optimization. A comprehensive investigation for each algorithm is presented by analyzing its merits and demerits in the context of target interception. This broad review is an outline for scholars and professionals in the field of the swarm of UAVs.

KEYWORDS
ABC; ACO; BA; 3D path planning; PSO; SI; UAV

1. INTRODUCTION

The dramatic increase in the availability of ground and aerial autonomous and semi-autonomous vehicles extends the opportunities of using them in a variety of services and tasks. Autonomous vehicles are already being used in search and rescue operations in disaster-prone areas [1], surveillance [2,3], 3D mapping [4,5], remote sensing [6], and more. Amazon has recently started using drones for delivering packages [7], and 40% of the rice crops in Japan are sprayed by Unmanned Aerial Vehicles (UAVs) [8]. The extended use of autonomous vehicles in both civil and military contexts presents many potential benefits, as well as risks. The risks vary from minor unintentional accidents (thumping into an object) to threats to national security. Mitigating the risks introduced by the extended use of autonomous mobile vehicles, in particular drones, include the implementations of rules and regulations [9], detection and identification, taking control, and possibly destroying them.

An important task in mitigating the risk of hostile UAVs that threatens to invade a restricted space is the ability to intercept them before they cause damages. By interception, we refer to engaging a defending agent with the threat to monitor its action from proximity or to prevent it from completing its intended mission. Many applications that require interception of multiple dynamic ground or aerial threats can be executed by a swarm of UAVs as it can perform simultaneous tasks while adapting to the dynamic changes in the environment. A swarm of UAVs is also more resistive to distinct types of faults under certain situations, in particular, a failure of a single or a few UAVs within the swarm that may not necessarily affect the entire mission [10]. A key feature in the operation of a swarm of UAVs is the ability to assign the targets and to construct a 3D trajectory for each member of the swarm. Various approaches have been developed and implemented to solve the general problems related to 3D path planning for a single UAV, like the Probabilistic Road Maps [11], A* algorithms [12], Artificial Potential Field [13], Probabilistic Navigation Function [14], and many others. Many of these algorithms use sampling-based and graph-based search methods, which are well suited for high-dimensional configuration space, and are relatively easy for implementation. They are also known to be probabilistically completed such that the probability for finding a solution increases, given sufficient time. However, some of these methods suffer from shortcomings such as possible trap in local minima, and limitation due to constraints related to the characteristics of the grid. These algorithms are often computationally complex and lack in the balance between exploration and exploitation. Some of these algorithms do no't have robust nature and thereby fail to work in automated real-time applications as well as in environments with multiple dynamic obstacles [15,16]. In [17], the authors abridge extensive work in this area, focusing on 2D path planning, but there is no direct reference to swarm intelligence in interception missions. Beard *et al.* [18] present a new method for path planning and target interception for multiple UAVs named Eppstein's k-best. This method generates a trajectory that is not constrained to the predefined waypoints, and thus gives only suboptimal solution.

1.1 Motivation

UAVs are extensively used for risk evaluation and operations under threat without causing any risk to human life. UAVs need some interaction with humans, particularly when continuous monitoring is required. However, a human's ability to monitor a large swarm of UAVs is limited. In such cases, the UAVs should have the ability to make self-decisions for task allocation and safe path navigation which, in turn, require efficient path planning algorithms or techniques.

Planning a safe path for UAV, from initial to target position, is not an easy task and depends on various factors such as path length, obstacle avoidance, power consumption, etc. By implementing integrated task allocation and efficient path planning algorithms (based on swarm intelligence), UAVs can follow optimized paths for target interception combined with collision avoidance. In this review paper, we discuss the problems related to the combination of path planning and target interception by a swarm of UAVs. A swarm of UAVs is typically characterized by highly redundant and relatively simple UAVs, having relatively simple onboard sensors to interact with other UAVs, as well as with the environment. Due to these properties, a swarm of UAVs is highly scalable, flexible, and robust. Although there are various review papers on path planning of UAVs swarms [19–25] which considers the path planning problem, they do not consider the problem of interception of a moving target. Path planning for interception of moving target is fully associated with the obstacles present in the environment, as well as with the dynamics of the targets. In this review paper, we focus on existing Swarm Intelligence (SI) based algorithms and their modifications for solving the issues related to combined path planning and target interception for a swarm of UAVs in a 3D environment.

1.2 Contribution and Key Findings of the Paper

The main contribution and key findings of the paper are given as follows:

(1) We provide an extensive review of path planning algorithms based on swarm intelligence techniques for target interception missions by a swarm of UAVs.
(2) A critical evaluation of path planning techniques is carried out and depicted in tabular form. The parameters considered for the evaluation are computational complexity, the memory of previous states, search techniques, convergence speed, and scalability.
(3) We analyze and compare the distinct methods for environment modeling, target allocation, and formulation of fitness function in terms of the effective path planning of swarm of UAVs.

The paper is organized as follows: Section 2 presents the issues related to the modeling of 3D environments and the different problem formulations for evaluation (cost) functions. Introduction to the general problem of target interception is described in Section 3. This section also discusses applications of SI for target interception. Section 4 presents a brief overview of metaheuristic algorithms and Section 5 elucidate and discuss SI algorithms for path planning and target interception in the case of a swarm of UAVs.

A comparison of the algorithms grounded on multiple parameters is given in Section 6, and Section 7 provides the concluding remarks.

2. PLANNING A 3D PATH FOR INTERCEPTION MISSIONS BY A SWARM OF UAVS

Path planning for interception of targets in a 3D space is one of the most challenging tasks in motion planning, especially in the case of a swarm of UAVs. Path planning is usually the process of assigning targets to the swarm's members and constructing an optimal path from source to target while avoiding collisions with obstacles and other swarm members while optimizing the outcome with an evaluation (cost) function by considering kinematic and dynamic constraints. Figure 1 shows the general process scheme for target interception. The process can be abridged in the following three steps.

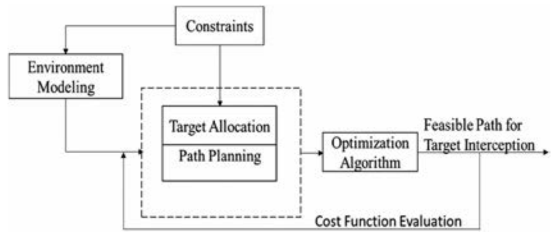

Figure 1: Path planning process for target interception

Step 1. Perception and modeling of the 3D environment.

In this step all relevant features of the environment, in particular the status of the swarm, the targets, and the obstacles, are determined. Generally, it is based on dynamic occupancy grid methods [26], or on intercept probability estimation methods combined with predictive control algorithm [27] to model the positions of the obstacles and the targets or threats.

Step 2. Target Allocation and path planning.

In this step, all the targets present in the environment are allocated to individual UAVs constituting the swarm. The allocation is based on the mission's constraints (*e.g.* the number of UAVs allocated to each target), the type of the targets and their order of perception, time constraints, etc. Based on the target allocation, the path of each UAV in the swarm is constructed.

Step 3. Formulation of a fitness function.

A fitness function is formulated by considering all kinematic and dynamic constraints of the 3D environment and finding the optimal path for each UAV.

2.1 Environment Modeling

Environment modeling is a significant factor in target allocation and motion planning, in particular for governing the motion of a swarm of UAVs, because of the critical interaction between the UAVs and the obstacles in the environment. The methodologies and limitations of several popular algorithms used for environment modeling in interception missions are depicted in Table 1.

In a study on 3D path planning algorithms [19], the environment is defined as a workspace in which obstacles are denoted as a set of points and a set of functions models of the obstacles and the targets geometries within the workspace. Authors in [18] implement the Voronoi diagram method for environment modeling in interception missions by UAVs. In this method, the region of interest,

Table 1: Limitations of techniques used for environment modeling

References	Methodology for environment modeling	Limitations
[18]	Voronoi diagram	Use fixed boundaries
[19]	Set of points and functions	Not good for dynamic environment
[24]	Bernoulli distribution	Information is shared between adjacent cells only
[25]	Euclidian space	Requires preliminary sensor data
[26]	Kalman filter	Assumes linear equation for representing the environment
[27]	Octomap	Constructs online 3D model
[28]	Trigonometric function	Generates only discontinuous model of environment
[29]	Time- environment dynamic map	Not suitable for 3D modeling
[30]	Marcov decision process	Not easy to implement

having n number of obstacles, is divided into n convex cells, constructed by equidistance lines from the obstacles, forming the Voronoi polygon edges. Each edge of the Voronoi polygon is associated with a cost function and is used for finding the global solution for a swarm of UAVs in 3D space. In [28], a numerical framework, based on Bernoulli distribution, is proposed for modeling the environment for target search in an open space scenario. The entire environment is divided into a cell grid. Each cell is observed by an individual UAV and has the probability of target existence, modeled based on Bernoulli distribution as 0 (target not found) or 1 (target is found) detection result. Only the UAVs that are present in the adjacent cells can share the information related to the presence of the target with each other in that cell.

A global method for modeling the environment in target interception mission is introduced in [29], where the entire environment is considered as a 3D Euclidean space. A curve algorithm known as L+Dumo is implemented for efficient target interception. The algorithm is iterative and uses preliminary sensor data for generating the 3D model of the environment. In another study [30], an onboard pinhole camera is employed for modeling the 3D world. Every pixel of the image taken by the camera gives the information of the measured distance (by taking the cumulative sum of the range of each pixel) from the starting position to the target position and the position of the obstacle. An extended Kalman filter-based navigation technique, combined with sensor data fusion, is used for efficiently avoiding the obstacles.

Sanfourche *et al.* [31] introduce an online 3D environment modeling by using the octomap. It is open source software, which processes the measured data obtained from any kind of distance sensor for constructing a 3D model of the unknown environment. In [32], the author implements a trigonometric function and city surface based terrain modeling. In city surface-based modeling, a 3D terrain is simulated by using the basic information extracted from the map of the city, such as the heights and the locations of the centers of the buildings. The modeling of the building is done based on the three matrices, as given below.

$$Bu_i^c|^{i=1,2,\ldots\ldots\ldots N^B} = (x_{1,i}, x_{2,i}) \tag{1}$$

$$[x_{1,i}x_{2,i}]^T = \begin{matrix} rand[{x_1}^L, {x_1}^U] \\ rand[{x_2}^L, {x_2}^U] \end{matrix} \tag{2}$$

$$Bu_i^w = rand[0, H^L] \tag{3}$$

where the position of each individual building is denoted by Bu_i^c, where N^B denotes the overall count of buildings, x^U and x^L present the maximum and minimum limit of horizontal distance of the search space, respectively, H^L is the maximum height limit of buildings, and Bu_i^w stands for each building's width. These three matrices present discrete and discontinuous model of the terrain and adopt Bezier curve for generating a smooth trajectory for UAVs from start to goal point.

Zhao *et al.* [33] proposed a new method for the modeling of the environment by implementing the time-environment dynamic map (TEDM), Where, the modeling of a dynamic obstacle is carried out by introducing a time axis, and each target is bifurcated into two parts: local and global target. Simulation results clearly show the efficiency of TEDM in the modeling of irregular obstacles. A Markov decision process (MDP) based method is presented by the authors in [34] for online modeling of environment and overcome the problems of uncertainty existing in A* based algorithms.

2.2 Target Allocation

The main objective of target allocation in an interception mission by a UAV swarm is to minimize the survival probability of the targets and to minimize the total time of the mission. Target allocation can be divided into two types: static and dynamic. In static target allocation, all the targets are allocated to the individual UAVs in a single stage, while in dynamic target allocation, the targets can be reallocated to the individual UAVs during the mission due to the alteration in the environment and the target status. Table 2 provides a list of common methodologies used for environment modeling and their limitations.

Generally, UAVs are equipped with a small payload of distinct types and capabilities. Also, some targets may have different values from others. Therefore, the characteristics of the UAVs and the targets must be considered in target allocation. Authors in [35] adopted model

Table 2: Limitations of techniques used for target allocation

References	Methodology for target allocation	Limitations
[31]	Model predictive control	Lack of flexibility
[32]	PSO	Suffer with local optimum
[33]	Negotiation mechanism	Not effective for homogenous group of UAVs
[34]	Decentralized auction mechanism	Not effective for moving target
[35]	Asynchronous mechanism	Time sequence constraint
[36]	MTWPS algorithm	Not suitable for offline mode
[3]	FLGA	Not effective for heterogenous group of UAVs

predictive control technology to maximize the objective function that considers the value of the targets and the probability for the interception, while avoiding collisions and minimizing the time. One of the major drawbacks of the model predictive control technique is its complexity. To deal with this complexity, Cruz *et al.* [36] have implemented the SI-based algorithm known as particle swarm optimization (PSO), where the best possible target allocation is done on the basis of Equation (4).

$$\max J = \sum_t \sum_u \sum_d \sum_j C_{tudj} x_{tudj} \tag{4}$$

C_{tudj} is the total cost of the target allocation, x_{tudj} denotes the decision variable, u represents the indices for the UAVs, d stands for the decoys and j represents the jammer. The value of function is 1 if one of the combinations of (u, d, j) is allocated to the target (t), and all the targets are allocated otherwise 0.

Sujit *et al.* [37] proposed a decentralized target allocation method known as negotiation mechanism, where UAVs communicate with each other and share the targets' information such as position and value, for the negotiation of target allocation. They have also considered the effect of the range of onboard sensors in target allocation. Experimental results demonstrate that when the range of the sensors is low, the negotiation mechanism with sharing information performs better than without sharing information. However, the proposed mechanism lacks in terms of convergence speed because it uses a random approach to find the solution. For this, Liu *et al.* [38] proposed a ubiquitous method by combining the ant colony system and decentralized auction mechanism. The decentralized auction mechanism is used for allocating the targets to the agents (UAVs) in minimum time and the ant colony system is used for path planning to intercept the target.

Authors in [39] implemented asynchronous task allocation mechanism for allocating the targets. In this mechanism, a swarm of UAV is divided into subgroups and allocated to the targets based on a priority basis. Simulation results indicate that this mechanism reduces the communication time and the complexity of the system.

Chen *et al.* [40] proposed a modified two-part wolf pack search (MTWPS) algorithm where the time-sensitive and parameter uncertainties are considered for task assignment in the case of heterogeneous multi-UAVs. To solve the problem of parameter uncertainties, authors have used the interior method and an online hierarchical planning algorithm is implemented to remove the time-sensitive uncertainties. Experimental results clearly show that MTWPS outperforms the other algorithms in terms of complexity and time.

In another study [3], authors have utilized a fuzzy logic-based genetic algorithm (FLGA) for the task allocation of multi-UAVs. Fuzzy logic determines the solution based on trial and error method and thus takes time to solve the problem of task allocation. To overcome this problem, the hybridization of a genetic algorithm with fuzzy logic is implemented by the authors. Simulation results depict that FLGA outperforms the k-means and c-means algorithms in terms of scalability and computational cost.

2.3 Fitness Function

The purpose of a fitness (cost) function is to establish a link between an optimization algorithm for motion planning and the real world. When implementing SI-based algorithms, the establishment of an effective fitness function is required for determining the efficiency of a UAVs swarm. Once the effective fitness function is established, it becomes the root criteria for determining the value of a trajectory for each UAV for intercepting a target, as well as the efficiency of the entire swarm. Several popular fitness functions and their shortcomings in terms of determining the efficiency of UAVs swarm are listed in Table 3.

In some previous research [41–43], the cost functions were determined only by considering the path length from source to target, and the obstacles were presented

Table 3: Shortcomings of objective function used for determining the efficiency of UAVs swarm

References	Methodology for formulating objective function	Limitations
[41]	Weighted sum	Not considered smoothing of path
[42]	Adaptive weighted sum	Not suitable for dynamic environment
[43]	Weighted sum	Not considered fuel cost
[44]	Normalized weighted sum	Suitable for static environment only
[45]	Weighted sum	Not considered fuel cost

in a static environment. However, this type of cost function is not suitable for a dynamic environment, where the obstacles and the targets change their position. Constraints such as flying height, turning angle, and heading angle, related to the UAVs capabilities, should also be included while establishing the cost function for efficient path planning of target interception [46]. Besada-Portas *et al.* [44] formulated a cost function by including the cost of fuel consumption during real-time mission in the battlefield. Zhu, W. *et al.* [45] proposed a cost function that considers only the threat (such as obstacle) zone cost (J_T) and the fuel cost (J_F) and are defined in Equations (5) and (6), respectively.

$$J_T = \int_0^L W_T dl \tag{5}$$

$$J_F = \int_0^L W_F dl \tag{6}$$

where W_T and W_F are the variables related to the total length of the flyable path and respectively present the threat (obstacles, radar detection, etc.) and fuel cost of each line segment on the generated path-L. Since both threat and fuel cost functions are in different dimensions, both are normalized between zero and one before optimization.

A cost function for intercepting a moving target using the cooperation strategy of a swarm of UAVs is designed by including the threat exposure to UAV and the average Euclidean distance from start to the goal position [47]. The cost function (J_t) is minimized by implementing the probability density function (PDF), and is defined as follows:

$$J_t = K_{tc} \sum_{j=1}^{N} f_{min,j} + \alpha K_{dc} \sum_{j=1}^{N} d_{min,j} \tag{7}$$

where K_{tc} and K_{dc} denote the weight coefficients, $d_{min,j}$ defines the average Euclidean distance from the initial position of the jth UAV to the target, α is defined as the normalization factor and $f_{min,j}$ denotes the minimum distance of a threat from the jth UAV. In another study [48], authors suggested to include an additional restriction function in the total cost function as given in [49]. The restriction function is implemented to generate the paths such that a swarm of UAVs can avoid the obstacles while maintaining a safe turning radius from nearby obstacles. PSO is used for minimizing the total cost function (J_{cost}), as given in Equation (8).

$$J_{cost} = \frac{W_4 J_4 + W_5 J_5}{W_4 + W_5} \tag{8}$$

where w_4 and w_5 present the weights chosen based on different flight tasks. The total length of the generated path, presented by $J_4.J_5$, is basically the cumulative sum of the angular displacement of the generated obstacle-free path. An effective fitness function is developed in [50] by including two new functions that affect the performance of a UAV during its operation. One function is the distance to destination, and the other is the constrained searching space (the specific area in which a swarm of UAVs should perform its task). The distance to destination function is used for determining the right search direction for the moving target in the 3D space, and the constrained searching space function is used for defining the maximum and minimum distance from the threat areas.

3. TARGET INTERCEPTION

Various approaches have been proposed for solving the issues related to target interception by a swarm of the mobile robot [51,52]. This section presents the different parameters and assumptions related to this problem as shown in Table 4. The parameters presented in the table are:

- Prior knowledge of the environment – the level of information available before and during the interception process. Information can be given a-priori or can be acquired during the process using real-time data flow.
- Type of communication – agents can share information (cooperative) or act independently without sharing information with other agents (non-cooperative).
- Obstacle avoidance – some methods focus on the actual interception process only, while others also consider collision avoidance with other agents and obstacles.
- Location of targets – like the prior knowledge of the environment, some methods require prior knowledge about the locations of the targets (as well as their trajectories) while others use real-time data acquisition tools for determining targets' locations.
- Number of targets – the number of targets has a significant impact on the process complexity.

A distributed control strategy is implemented by the authors in [59] for cooperative target interception by a swarm of mobile robots. Each robot is assigned to a specific moving target according to a minimum weighted distance, without using a centralized controller, and intercept the moving target by adopting parallel navigation. In parallel navigation, the mobile robots maintain their orientation parallel to each neighboring robot for intercepting the moving target, instead of following the path taken by the target. Experimental results show targets can be intercepted in minimum time. Sheridan *et al.* [60] present a method for an online interception of a group of moving targets by a swarm of robots using the concept of rolling-horizon. In this method, the swarm is divided into sub swarms based on the number of targets, for reducing the computational time. They show that this concept minimizes the time for mission completion. Unfortunately, these methods are suitable only for target interception in a 2D plane. In [61], the authors employ a reliable least square estimator technique for processing the generated guided waypoints to intercept moving targets in 3D space. Tracking errors (the difference between the desired waypoint and the current waypoint) are calculated and recorded using on-board sensors and a companion computer at the generation of each guided waypoint (determined by a GPS). They are then synchronized with the previously guided waypoint to keep the right track of the UAV. Each guided waypoint is represented by a set of variables $[x, y, \theta, T]$, where $[x, y]$ represents the coordinates of the waypoints, θ the heading of the subsequent path segment and T is the time taken to generate the new way point. In a recent study [62], limit cycle based algorithm is implemented to intercept a moving target with an unknown speed. Kalman filter is used for calculating the position and velocity of the target. Each target is represented by the state vector $X_k = [X_t, Y_t, \theta_t, X_t, Y_t, \theta_t]$, where $[X_t, Y_t, \theta_t]$ denotes the initial position and heading of the target, and $[X_t, Y_t, \theta_t]$ represents the final position and

Table 4: Problem variants in different target interception studies

Method	Prior knowledge of environment	Type of communication between UAVs	Obstacle avoidance	Location of Target	Number of targets
Xin *et al.*[53]	Known	Cooperative	Not considered	Unknown	25
Zhong *et al.* [54]	Known	Cooperative	Considered	Known	3
Jianjun and Simon [55]	Unknown	Cooperative	Considered	Unknown	Multiple
Fu *et al.* [56]	Unknown	Cooperative	Not considered	Known	Multiple
Khosravi and Aghdam [17]	Known	Cooperative	Considered	Unknown	Multiple
Khosravi and Aghdam [57]	Unknown	Non-cooperative	Not considered	Unknown	Multiple
Pierson [58]	Known	Cooperative	Not considered	Unknown	Multiple

heading of the target, assuming that the target is moving in a straight line with a constant velocity. Kothari *et al.* [63] design a new coordinated controller based on sliding mode and consensus control technique, for the interception of a target by forming a circular pattern at a fixed distance from the target. Sliding mode control is a non-linear control technique having beneficial properties like accuracy, robustness, easy tuning, and implementation. The consensus is a method of reaching an agreement on some information of interest in a distributed manner as described in [64]. They assume that both the swarm of UAVs and the targets are at a constant altitude. They conclude that by using sliding mode control, there is no need for complete information of the target.

In recent investigation [65], a uniform gene encoding strategy is employed for automatic assignment of targets to UAVs based on the best flight cost of the target. The triple vector $[U_i, T_j, C\ (i, j)]$ is adopted to represent the uniform gene encoding, U_i is the ith UAV, T_j is the jth target and $C\ (i, j)$ denotes the flight cost of the ith UAV to the jth target. In order to convert the discrete space between the UAVs and targets into a continuous space, an improved differential evolution algorithm is implemented that uses the cost value in gene code.

4. METAHEURISITC ALGORITHMS

Metaheuristic algorithm have gained prominence in the field of artificial intelligence and mathematical optimization over the last two decades. The term metaheuristic was first coined by Glover in 1986 which combines the Greek prefix meta (high level) and heuristic (to find/to discover). These algorithms are stochastic in nature and finds optimum solution with less computation effort as compared to conventional techniques. The unique features of these algorithms are their simplicity, flexibility, derivative free operation and local optima avoidance capability. Rationally these algorithms have proved that no algorithm can provide unique solution for all sets of problems which motivates researchers to develop new algorithms. These algorithm are problem independent and can be mainly classified into four different classes: swarm-based, physics-based, evolutionary-based and human-based algorithm. Swarm intelligence (SI) is defined as the collective intelligence of group of agents such as birds, ants, fish, wolves, etc. SI algorithm are inspired from this social behavior and finds the optimal solution by maintaining a balance between exploration and exploitation in the search space. Exploration refers to globally investigating the search space while exploitation refers to locally searching around the propitious areas found in exploration phase.

5. SWARM INTELLIGENCE-BASED ALGORITHMS FOR PATH PLANNING OF TARGET INTERCEPTION

In this section we discuss the distinct types of SI-based algorithms for path planning of target interception by a swarm of UAVs and recommend improvements for path optimization.

5.1 Particle Swarm Optimization

PSO is a SI-based algorithm that was first proposed by Kennedy and Eberhart [66,67] to provide a graphical simulation of flocking behavior of birds [68]. In PSO, each potential solution is considered as a particle, having its random velocity and position in the search space. The search space is defined as the set of all probabilistic solutions for the problem to be optimized. Each particle achieves its best position and velocity according to the best solution (fitness) in the solution space. The ith particle in the PSO algorithm updates its velocity and position at every Tth step according to the equations given below.

$$V_i^{T+1} = W * V_i^{T} + r_1 * C_1 * (P_{best} - X_i^{T}) + r_2 * C_2 * (G_{best} - X_i^{T}) \tag{9}$$

$$X_i^{T+1} = X_i^{T} + V_i^{T} \tag{10}$$

where X_i^T and V_i^T denote the position and velocity vector of the ith particle in the swarm, W represents the inertia weight to maintain the balance between local and global search ability, and C_1, C_2 denote the acceleration constant and are predefined by the user. r_1 and r_2 are random numbers generated in the range of [0 1]. P_{best} is the personal best position of the ith particle at time T, and G_{best} is the global best position of the ith particle within the swarm. The term inertia weight (W) was

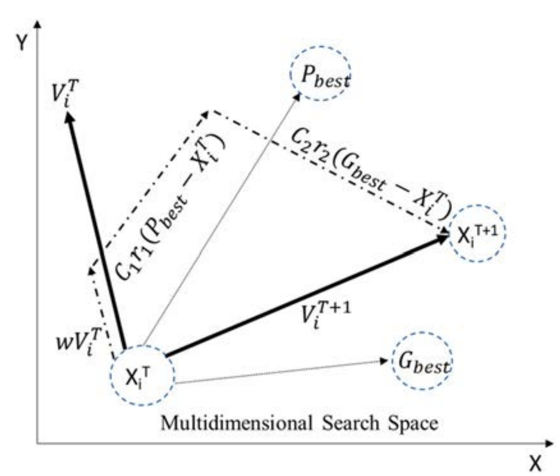

Figure 2: The representation of Particle Swarm Optimization model

not included initially in the ordinary PSO, it was included by Shi and Eberhart in 1998 [69]. The representation of PSO model is shown in Figure 2. In this figure, the bold lines depict the velocity and position of the particle after each iteration. The dotted lines depict the components of Equation (7), While X and Y represents horizontal and vertical direction of search in a solution space, respectively.

The authors in [70] proposed a modified version of PSO for the interception of moving targets by adapting the graph theory. They assumed that the time taken by each UAV to perform the task (attack) is negligible compared with the time taken by each UAV to reach the target. The number of UAVs required for intercepting each target is calculated according to the property of that target. For increasing the convergence speed of the algorithm, they suggested to decrease the inertia weight (W) linearly from 1.2 to 0.2 and to vary the values of C_1, C_2 from 2.05 to 0.4 and 3.7, respectively.

In [71], a new approach was implemented with PSO in a swarm of UAVs where each particle was defined by three vectors: position, velocity and particle's own best position. The particle's own best position (determined by the algorithm itself) keeps track of previous positions of each particle, and the best position of the whole swarm is calculated according to the set of particle's own best position. In this implementation, the entire swarm is divided into two groups: one for intercepting the target, and the other for patrolling. The function of the patrolling group is to identify an intruder and to alert the interception group. Recently, Xing *et al.* [72] proposed an improved version of PSO to solve the problem of local minima of the solution and inability of local precise search during the time of execution. The entire swarm is divided into m sub-swarms according to the fitness function of each particle, and then the maximum fitness value for the whole swarm is calculated based on the maximum fitness value of each m sub-swarm. Each particle updates its velocity and position according to the following equation.

$$V_i^{T} = W * V_i^{T-1} + r_1 * C_1 * Sub_1 + r_2 * C_2 * Sub_2 \quad (11)$$

where Sub_1 and Sub_2 are updated at every T^{th} step according to the following equations.

$$Sub_1 = P_{best} - Swarm \quad (12)$$

$$Sub_2 = G_{best} - Swarm \quad (13)$$

In [73], the authors proposed an enhanced PSO for three-dimensional path planning of UAV having a fast and more optimal solution. In this algorithm chaos-based logistic map is adopted for the initial deployment of particles in the search space. Thereafter, the mutation strategy is adopted while acceleration coefficients (C_1, C_2) and maximum velocity are made adaptive to obtain the optimal solution. In [74], a hybrid PSO and APF (artificial potential field) algorithm is proposed. The proposed algorithm is fast and finds the optimal path in obstacle-rich environment.

Improved version of PSO acquires a more desirable result in the case of complex environment to avoid the obstacles as compared with the basic version of PSO.

5.2 Ant Colony Optimization

Ant colony optimization (ACO) was originally proposed by Dorigo *et al.* [75,76] as a multidimensional optimization algorithm inspired by the foraging behavior of a specific ant species. Ants wander randomly in the search of food by laying down their trail a chemical substance known as pheromone and use it as feedback for returning to the nest. The selection of a path by other ants is also based on the pheromone trail. The density of the pheromone trail along the path increases the probability that other ants will choose the same path and thus the pheromone trails on that path become denser. The representation of the ACO model is shown in Figure 3. The ant colony then collectively determines the shortest path based on the density of the pheromone trail [77].

A multi-colony approach based on ACO was implemented by the authors in [78] to avoid collisions between UAVs and to minimize the loiter time to complete the

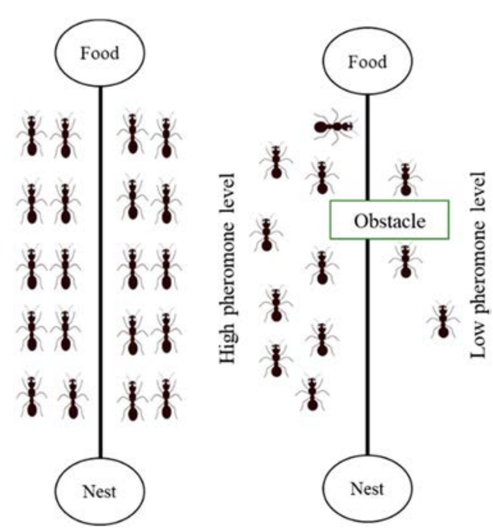

Figure 3: Searching behavior of ants in ant colony optimization

task (search and attack). In this system, the swarm of UAVs selected the shortest path from the *i*th node to the *j*th node, in a directed graph representing the environment, based on the previous N different loitering times taken by the ant colony. Experimental results clearly show that by implementing this approach, the time taken to complete the task is reduced.

As stated by Gao *et al.* [79], the main challenge in using ACO for a swarm of UAVs for target search and attack is path smoothing and pheromone update. Path smoothing is necessary for determining the shortest path between any two points, and the secretion amount of pheromone by the ants in the foraging area is very important since the ants decide their movement based on that amount. The authors have suggested a modification to the ACO by introducing an iteration threshold. Each ant has its processor to deal with the problem of local minima and to establish proper communication with other ants. For avoiding the oscillatory behavior of ants, a new rule of pheromone updating for the global search was proposed as follows.

$$T^{i}_{(x,y)}(K+1) = T^{i}_{(x,y)}(K) + f \times \Delta T_{g0} \tag{14}$$

where $T^{i}_{(x,y)}(K+1)$ is the pheromone level after each iteration, f presents the uncertainty in the environment, and ΔT_{g0} shows the global pheromone update coefficient.

In [80], improved ACO was adopted for path planning of intercepting multiple targets through coordinated unmanned combat aerial vehicles (UCAVs). The authors have considered three UCAVs (as UCAV I, UCAV II, and UCAV III) and two targets, where each UCAV is equipped with an attacking device. The targets are detected by UCAV I using its onboard sensors. After target detection, UCAV I generate the optimal trajectories for all UCAVs by considering the environmental constraints. UCAV II and UCAV III receive all the information regarding the trajectories from UCAV I and follow these trajectories. The rolling pseudospectral method (RPM) was employed with improved ACO for better path planning and for dealing with the dynamic constraints related to uncertainties in the environment. The RPM method uses the discretization of control and state variables for solving the optimization problem.

In [81], a hybrid approach of ACO and artificial potential field (APF) algorithm is proposed for cooperative mission planning of UAV swarm in an uncertain dynamic environment. The simulation results present that the proposed hybrid approach efficiently executes the mission and avoids the obstacles without collisions.

An improved ACO algorithm is proposed for UAV path planning in complex mountainous area for emergency rescue operation [82]. The proposed approach utilizes Tyson polygon structure to build the initial solution and presents faster solution with minimum path length as compared with traditional algorithm.

In general, enhanced version of ACO can produce smooth path from start position of UCAV to the target position in very efficient way and outperforms PSO and GA.

5.3 Firefly Algorithm

Firefly algorithm (FA) was defined by Xin-She Yang [83,84] in 2008. It is inspired by the flashing pattern of the fireflies in tropical region to attract other fireflies. The representation of FF algorithm is shown in Figure 4. FA is formulated by considering the following three rules.

- All fireflies are unisex.
- The attractiveness of each firefly is directly proportional to its light intensity.
- The brightness of each firefly represents a possible solution to the optimization problem and is proportional to the fitness function.

The attractiveness (β) of a firefly seen by an adjacent firefly varies according to the distance (r) and is given as in Equation (15).

$$\beta = \beta_0 e^{-\gamma r^2} \tag{15}$$

where β_0 defines the attractiveness at zero distance ($r = 0$), and γ represents the light absorption coefficient of the medium. Firefly k moves towards a brighter firefly

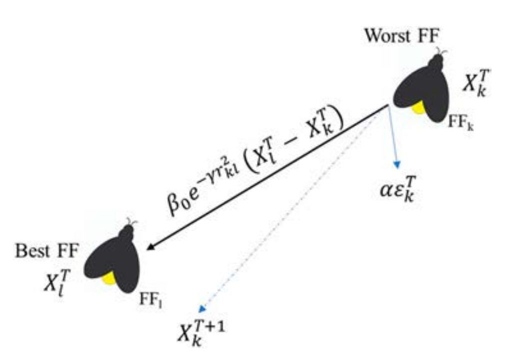

Figure 4: Attraction behavior of fireflies during search in a solution space

l according to the following equation.

$$X_k^{T+1} = X_k^T + \beta_0 e^{-\gamma r_{kl}^2}(X_l^T - X_k^T) + \alpha \varepsilon_k^T \tag{16}$$

where X_k^{T+1} represents the new position of firefly k after a single time interval (T+1). X_l^T and X_k^T define the position of firefly l and k, respectively at time T. The Euclidean distance between the two fireflies is represented by r_{kl}^2. ε_k^T is a vector of random numbers formulated by the uniform distribution between zero and one. α is the randomization parameter, used for improving the quality of the optimization. Larger value of α leads to better global search capability of the algorithm.

Liu *et al.* [85] designed the adaptive absorption (γ_i) and randomization parameter (α_i) for the efficient path planning for the target interception in static environment. For increasing the convergence speed of the FA algorithm, absorption parameter was formulated according to Equation (17).

$$\gamma_i = \gamma_0 + \frac{(\gamma_f - \gamma_0)}{n} \tag{17}$$

where, γ_0 presents the initial value of absorption coefficient, the final value of absorption coefficient is γ_f, and n shows the total number of iterations. To improve the quality of the solution, randomization parameter was formulated as given in the following equation.

$$\alpha_i = \alpha_0 + \frac{(\alpha_f - \alpha_0)}{n} \tag{18}$$

where α_0 presents the initial value of the randomization parameter, the final value of randomization parameter is α_f and n shows the total number of iterations.

In [86], two minor modifications were proposed in FA. One was adding the information (related to the target) exchange only in between the best fireflies for the fast convergence, and the other modification was the addition of levy flight with step size for decreasing the randomization. Levy flight is a specific class of random walk in which the step size is determined based on heavy-tailed probability distribution. These two parameters play a key role in the generation of best fireflies. Simulation results depict that by keeping the value of the light absorption coefficient to 1.0 and the step size of 0.25, the best results were acquired.

Firefly algorithm with Leavy flight increases the exploitation capability which is suitable for the multiple target interception with the shortest path as compared with basic firefly, PSO, and ACO.

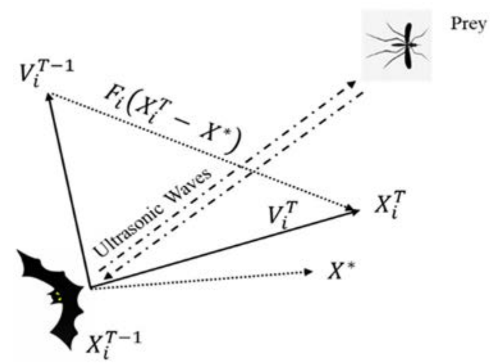

Figure 5: Searching behavior of bats in bat optimization algorithm

5.4 Bat Algorithm

Bat Algorithm (BA) is a metaheuristic method for solving the optimization problem, developed by Xin-She Yang [87] in 2010. This algorithm is inspired by the echolocation behavior of microbats. These bats emit an ultrasonic signal in the frequency range of 25-150 kHz and then listen to the echo that comes back from the prey and obstacles in the surroundings. Microbats use the short sound pulses with low frequency for the echolocation [88]. The representation of bat algorithm is shown in Figure 5. The formulation of BA is done by considering the following three rules.

- The echolocation technique is used by all bats to sense the distance of the prey and obstacles present in the surroundings.
- Each bat flies randomly having its own velocity V_i, position X_i and fixed minimum frequency F_{min}, while the wavelength (λ) and loudness (A) may vary for searching the prey.
- The loudness of each bat varies from a large positive value (A_0) to a minimum constant value (A_{min}).

The ith bat in BA updates its position, velocity and frequency for searching the prey in a D-dimensional solution space at every Tth step according to the following equations.

$$F_i = F_{min} + \beta(F_{max} - F_{min}) \tag{19}$$

$$V_i^T = V_i^{T-1} + F_i(X_i^T - X^*) \tag{20}$$

$$X_i^T = X_i^{T-1} + V_i^T \tag{21}$$

where F_i is the frequency of the ith bat, and β is a random vector drawn from the uniform distribution between one and zero. V_i^T and X_i^T denote the velocity and position of the bat at the Tth iteration respectively. Here X^* denotes the current global best position chosen out by comparing the position of all the N bats. F_{min} and F_{max} present the minimum and maximum frequency, depending upon the domain size of the problem.

BA has been adapted with MAKLINK graph theory for the task allocation of mobile robot [89]. In the MAKLINK graph theory, each obstacle in the environment is presented as a polyhedron. The simulation results show that by keeping the pulse rate (for obstacle detection) at 0.5 and loudness at 0.25, the time taken by the algorithm to generate the optimal solution was minimum. Basic BA is good at the exploitation of the solution but lacks in the exploration of the solution. For this, authors have incorporated a mutation operator from the differential evolution (DE) algorithm into the BA [90]. The function of the mutation operator was to generate a new solution for each bat and thus the exploration property of the algorithm increases. For increasing the convergence speed of the basic BA, the values of the frequency and loudness were kept constant. Experimental results clearly show that by keeping the values of the emission pulse rate at 0.6 and loudness at 0.95, the best result was found.

In simple BA, the optimal solution is generated by using random walk, which slows down the convergence speed of the algorithm. In order to remove this problem, a mutation weighting factor was included [75]. A new optimal solution was calculated according to Equation (22), for increasing the search efficiency and convergence speed of the algorithm:

$$X_{NEW} = X^T_{R1} + (X^T_{R2} - X^T_{R3})F \tag{22}$$

where X_{NEW} represents the new optimal solution. $X^T_{R1}, X^T_{R2}, X^T_{R3}$ are uniformly distributed numbers, and F denotes the mutation weighting factor. In [91], the authors proposed an improved BA for tracking an invading target in the oilfield. Initially, a prediction method is employed to estimate the trajectory of the invading target and thereafter improved BA is used for optimizing the tracking trajectory of the UAVs. The improved algorithm utilizes the food searching capability of the fruit fly optimization algorithm (FOA) to avoid local optima and improve stability. Bat algorithm has a good exploration capability as compared with GA, ACO, and DE, and thus verifies its effectiveness for generating the shortest path from start to goal position in a short period.

5.5 Artificial Bee Colony Optimization

Artificial bee colony algorithm (ABC) was formulated by Karaboga [92], based on the foraging behavior of honey

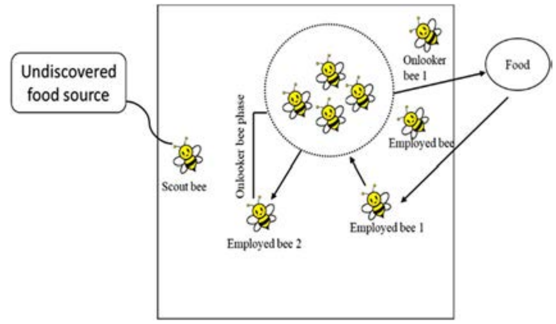

Figure 6: Foraging behavior of bees in artificial bee colony algorithm

bees, for solving a multidimensional optimization problem. In ABC algorithm, a possible solution of any problem is represented by the food source, and honeybees have the responsibility for searching the new food source. In ABC, the colony of artificial bees are divided into three groups: (1) employed bees, where the number of bees in the group is associated with the food sources near the hive, (2) onlooker bees that watch the waggle dance performed by the employed bees for sharing the information of food source with each other, and (3) scout bees that are responsible for searching new food sources randomly. After exhaustion of the food source, the employed bees become the scout bees to search for another food source. The representation of the bat algorithm is shown in Figure 6. This process runs in continuation till all the food sources are exhausted by the employed bees [93]. Authors in [94], implemented ABC for path planning and target interception through UCAV by adapting the probability density model. The probability density model was used for representing the threats. In this proposed method, the solution is generated by including avoidance of the threats, minimizing energy consumption, and shortening the distance from the source to the target. Each solution is represented as a "food source".

The UCAVs adjust their position and velocity in the solution space according to the source of the target, updated by the employed UCAV. The probability for selecting the food source is calculated by using Equation (23).

$$P_i = \frac{Obj(F_i)}{\sum_{i=1}^{n} Obj(F_i)} \tag{23}$$

where $Obj(F_i)$ is the best fitness value of the ith food source and n denotes the total number of food sources.

A memory saving algorithm known as compact artificial bee colony (CABC) algorithm was proposed by the authors in [95], where the mathematical model of the path length and the threats were included in the objective function for effective path planning and target interception. In the CABC algorithm, a PDF was used to generate the optimized solution.

In another study [96], ABC algorithm was modified by adapting the balance-evolution strategy to form a ubiquitous path planning and target interception method for UAVs. In a balance-evolution strategy, a balance between the exploitation and exploration for the search of the target by UAVs is sustained because of the internal convergence status during the iteration process. The authors depict, through experimental results, that the proposed strategy is efficient in avoiding the threats and intercepting the target in a dynamic environment. The efficient exploitation capability of the ABC algorithm makes it suitable for path planning in a static environment with multiple targets and it outperforms the BA, DE, and SA.

5.6 Bacterial Foraging Optimization

Bacterial foraging optimization algorithm (BFOA) was introduced by Kevin Passino [97] in 2002. This algorithm is inspired by the chemotactic behavior of *Escherichia coli* (*E. coli*) bacterium in a nutritious environment. *E. coli* bacterium achieves its locomotion with the help of flagella while searching for the nutrients (targets) present in the environment. During the foraging of the nutrients, this bacterium biologically possesses two types of movements: one is "swim" and the other is "tumble". The swim is the process in which bacterium moves in a particular direction and tumble is the process of changing the direction of movement. The fundamental concept of BFO algorithm is that the bacteria which are having good foraging strategy will survive, while the bacteria with poor foraging strategy will be eliminated. The representation of the BFO model is shown in Figure 7.

The BFOA consists of four steps: chemotaxis, swarming, reproduction, and elimination-dispersal. Chemotaxis is the event where the bacteria move in small steps towards the nutrient by avoiding the noxious environment. The process in which the bacteria arrange themselves in a ring pattern and then move together for the foraging of nutrients is known as swarming. In the reproduction stage, the least healthy bacteria (lower fitness value) are discarded and healthy bacteria (high fitness value) split into two bacteria for maintaining the size of the swarm. Elimination-dispersal is the final stage in which the bacteria disperse in the new environment according to the described dispersal and elimination probability [98]. At every chemotactic step, the movement of the single bacterium is defined as follows.

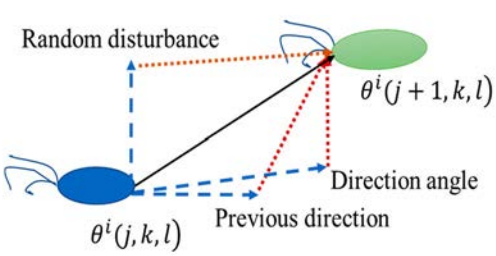

Figure 7: Searching behavior of BFO model

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\varphi(j) \tag{24}$$

where $\theta^i(j,k,l)$ represents the position of the ith bacterium at the jth chemotactic steps inside the kth reproductive stage of the lth elimination-dispersal event. $C(i)$ is used to define the step size during the running and tumbling of the bacterium. $\varphi(j)$ represents the generated unit length random direction, uniformly distributed in the range of $[0, 2\varphi]$.

Yang *et al.* [99] proposed a ubiquitous approach for target allocation of multi UCAV in a dynamic environment by using a bacterial foraging algorithm with a mixed re-allocation strategy. In this approach, each UCAV is represented as bacteria and searches for the nutrient (target) in a dynamic environment, where the position of the obstacles is not fixed. A mixed re-allocation strategy was used to increase the convergence speed of the original BFOA. The simulation results clearly show that by keeping a step size of 0.2, each UCAV is assigned to the target in a short duration of time.

A decentralize control algorithm, based on BFOA was employed by the authors in [100] for the interception of target by a swarm of mobile robots. In this algorithm, each "bacterial" robot finds the nutrient (target) based on the concentration, by adopting minimum path length and the minimum number of turns. The movement of each "bacterial" robot is calculated by the chemotactic step as follows.

$$\frac{dP_i(t)}{dt} = B\delta + Ad_i \tag{25}$$

where $P_i(t)$ denotes the position of the ith robot, δ represents the direction of movement of the robot, A and B are the coefficients of d_i and δ, respectively, d_i is the cumulative sum of interaction of the ith robot with other robots for avoiding collisions, and can be calculated by using Equation (26).

$$d_i = \sum_{n=1}^{C_i} d_i^n \tag{26}$$

where C_i represents the number of neighbors for the ith robot and d_i^n denotes the interaction of ith robot with the nth robot and can be calculated as follows.

$$d_i^n = \frac{(N_i - N_m)}{||N_i - N_m||} \tag{27}$$

where N_i and N_m represents the number of ith and mth robot, respectively.

In [101], the authors proposed an improved BFO algorithm for path planning of unmanned surface vehicles (USV) under the grid environment. In this approach, the A* algorithm is incorporated in BFO and is tested for the global path planning of USV in different working environments. Due to a smaller number of tuning parameters, the BFO algorithm is most appropriate for the case where there is a condition of target allocation for multiple UAVs in a dynamic environment and produces reliable results as compared with BA, ACO, and PSO.

5.7 Pigeon-Inspired Optimization

Pigeon-inspired optimization algorithm (PIOA) is a stochastic SI algorithm and was introduced by Haibin Duan and Peixin Qiao [102] in 2014, based on the homing behavior of pigeons. Pigeons use the magnetic field of the earth, the position of the sun, and specific landmarks for finding their way home. The representation of the PIO model is shown in Figure 8. In PIOA, the movement of each pigeon is defined by two mathematical operators, one is a map and compass operator and the other is a landmark operator.

In the map and compass operator, the position and velocity of the ith pigeon within the D-dimensional solution space in each iteration is updated as follows.

$$V^i(k) = V^i(k-1) * e^{-Pk} + rand * (X^g - X^i(k-1)) \tag{26}$$

$$X^i(k) = X^i(k-1) + V^i(k) \tag{28}$$

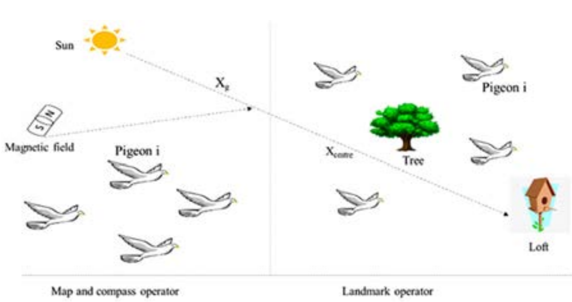

Figure 8: Optimization process of PIO

where $V^i(k)$ and $X^i(k)$ are respectively the velocity and position of the ith pigeon at the kth iteration. P denotes the map and compass operators, for finding the correct path and direction. X^g represents the current best global position of the ith pigeon and *rand* is any random number between zero and one.

In the landmark operator, the number of pigeons (N_p) in the kth iteration is reduced to half and the remaining pigeons will fly straight to the target. The pigeons that are still far away from the target location and are not familiar with the landmarks follow the pigeons which are familiar with the target location. In this operator, the ith pigeon updates its position at the kth iteration according to the following equations.

$$N_p(k) = \frac{N_p(k-1)}{2} \tag{29}$$

$$X^c(k) = \frac{\sum X^i(k).fitness(X^i(k))}{N_p \sum fitness(X^i(k))} \tag{30}$$

$$X^i(k) = X^i(k-1) + rand * (X^c(k) - X^i(k-1)) \tag{31}$$

where $X^c(k)$ is the center position of the ith pigeon at the kth iteration, and N_p is the number of pigeon. The quality of the i^{th} pigeon is represented by the fitness function $(X^i(k)) = \frac{1}{f_{min}(X^i(k))+\varepsilon}$. For solving the maximum optimization problem, the fitness function can be taken as:

$$fitness(X^i(k)) = f_{max}(X^i(k))$$

Zhang *et al.* [103] combined the predator-prey and PIOA. In this algorithm, the predator-prey concept was used for removing the tendency of basic PIOA to converge at a local solution. The modeling of predators is based on the worst solutions and can be presented mathematically as follows.

$$K^{predator} = K^{worst} + \delta\left(1 - \frac{T}{T^{max}}\right) \tag{32}$$

where $K^{predator}$ denotes a possible solution, K^{worst} presents the worst solution in the solution space, hunting rate is denoted by δ, T and T^{max} denotes the current iteration and the maximum number of iteration, respectively. The authors have also found that the predator-prey pigeon-inspired optimization algorithm (PPIOA) is superior to the different evolution (DE) technique in terms of robustness and convergence speed.

In a recent study [104], a hybrid model of edge potential function (EPF) and PIOA has been proposed for the target detection and interception by a group of UAVs at a low height. EPF is an approach used for edge-based detection in digital images [105]. Each UAV uses the EPF function for locating the target position. Hao *et al.*[106] presented a modified PIOA for solving the multiple target assignment and interception problem in a dynamic environment. A nonlinear increasing mutation approach was implemented to provide a balance between the velocity and global search capability of the UAVs in the map and compass operator (P). For increasing the convergence speed of the algorithm, referring to the genetic algorithm, a mutation crossover was also implemented. Simulation results of different scenarios show that the proposed modified PIOA is also capable of finding the optimal solution in a real dynamic world.

In a recent study, the authors proposed a robust Cauchy-mutated PIOA for path planning of multiple UAVs [107]. Mathematically, Cauchy distribution is the continuous probability distribution without variance and expectation. The author examined the robustness of the proposed algorithm by exploring it in a different environment such as plateau topography and plateau wind-driven environment.

In [108], an adaptive quantum-based PIOA is proposed for path planning of UAV in complex environments. The algorithm utilizes logical mapping for the initialization of the population in the search space and maintains a balance between local search and global search by adaptively balancing the factor parameters. The simulation results show that the proposed approach presents better results than PSO in terms of the rate of convergence and accuracy.

A good balance between exploitation and exploration search ability makes PIO as a well-established algorithm for solving the problem of multiple target allocations in a real-world dynamic environment.

5.8 Glowworm Swarm Optimization

Glowworm swarm optimization algorithm (GSOA) was proposed by Krishnan and Ghose [109] in 2005. It is a SI-based metaheuristic algorithm based on the bioluminescence behavior of the glowworm and can generate multiple local optimum solutions for a multimodal function that consists of many solutions. The unique quality of GSO is its ability to update the local decision domain range of each glowworm which leads to the generation of subgroups of the swarm and enables them to converge at multiple optimal solutions at the same time. In GSOA, each agent (glowworm) carries a luminescence quantity known as "luciferin", which also serves as the communication medium among the agents. The initial state of the GSOA consists of a random distribution of each agent in the solution space, and all the agents are having an equal amount of luciferin. Based on the transition rule [110], all the agents go through two phases: luciferin-update phase and movement phase. In the luciferin-update phase, agents update their luciferin value at each iteration as follows.

$$L_j(T+1) = (1-\rho)L_j(T) + \gamma J_j(T+1) \tag{33}$$

where $L_j(T)$ signifies the luciferin quantity of agent j at time T, γ represents the luciferin enhancement constant, ρ is the luciferin decay constant ($0 < \rho < 1$), and $J_j(T)$ denotes the value of the objective function at the location of agent j at time T. In the movement phase, each glowworm adapts a probabilistic mechanism to move towards the neighboring glowworm that has a higher luciferin value than its own. At each iteration, the mathematical model of the movement of glowworm i towards the brighter glowworm j is given by Equation (34).

$$X_i(T+1) = X_i(T) + S\frac{X_j(T) - X_i(T)}{||X_j(T) - X_i(T)||} \tag{34}$$

where S denotes the step size and $X_i(T)$ signifies the position of glowworm i at time T. Basic GSOA has slow convergence speed due to the individual location update model, which consists of only local information. The representation of GSOA model is shown in Figure 9.

In order to overcome this problem, Tang et al. [111] has combined the GSOA and PSO, where the luciferin value is denoted by waypoints. The new optimal position of UCAV in D-dimensional search space is calculated as follows.

$$\begin{aligned} X_i(T+1) = {} & X_i(T) + C_1 * rand * (X_j(T) - X_i(T)) \\ & + C_2 * rand * (X_{gb}(T) - X_i(T)) \end{aligned} \tag{35}$$

where $X_{gb}(T)$ denotes the global position of the individual UCAV, $X_i(T)$ represents the position of the ith

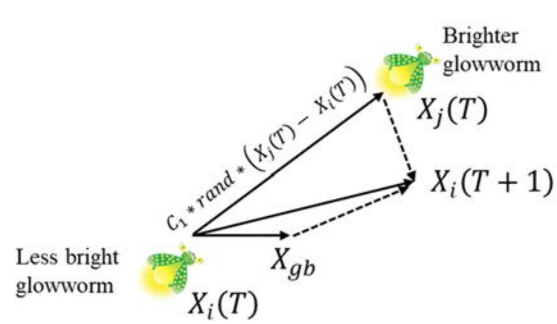

Figure 9: Movement of glowworm in glowworm swarm optimization algorithm

glowworm at time T, C_1 and C_2 denote the acceleration factor, *rand* is a random number between 0 and 1, and $X_j(T)$ signifies the position of the *j*th glowworm.

In real-world path planning of UAVs, the GSO algorithm is more competent than ACO where UAVs are having sensors with limited range. Furthermore, GSO also outperforms the PSO for the case of a dynamic environment due to its capability of avoiding local minimum and fast convergence speed.

6. COMPARATIVE ANALYSIS

This section presents a comparative analysis of the SI-based algorithms as discussed in Section 4 based on crucial parameters for target interception by a swarm of UAVs. These parameters consist of the computational complexity, required communication, optimization function, memory complexity, scalability, and search techniques. The comparison between the algorithms is presented in Table 5.

The *computational complexity* is defined as the requirement of the system for computation resources as a function of the number of robots, and the number of targets. The resources are defined by the expected computation time and expected computation storage required for the solution. The total number of robots (population) in a swarm is denoted by N_T, and the number of targets by T_T. The computation complexity is therefore determined by these two parameters. If an algorithm is endowed with partition features, then the number of robots in a subgroup of a swarm is denoted by N_S, wherein $N_S \leq N_T$, and similarly the number of targets in a subgroup is denoted by T_S. The algorithms that are endowed with partitioning technique, naturally have less computational complexity [112].

The *memory of previous states* affects the number of storage resources required by the algorithm. For a small-scale problem (low numbers of robots and/or targets), this parameter might be insignificant, but as the problems extend in terms of the number of robots and targets, it may become a crucial factor in the performance of the algorithm.

The *search technique* refers to the specific way in which the algorithm finds the solution of the problem. Most of the SI-based algorithms use three possible types of search techniques: mutation, selection and crossover [113]. The mutation is the mechanism for global exploration, while selection performs a dual role: one is to adopt the best existing solution in the search space, and the other is

Table 5: Path planning and target interception algorithm comparison

	Particle swarm Optimization (PSO) [112–115]	Ant colony Optimization (ACO) [79, 116–118]	Firefly Algorithm (FA) [86, 119]	Bat Algorithm (BA) [88, 89, 120]	Artificial Bee Colony (ABC) [93, 121, 122]	Bacterial Foraging algorithm (BFA) [99, 100, 123]	Pigeon-Inspired Optimization (PIO) [102, 124]	Glowworm swarm Optimization (GSO) [109, 125]
Computational complexity	$O(N_T)$	$O(N_S)$	$O(2N_S)$	$O(N_T)$	$O(N_S)$	$O(2N_S)$	$O(N_S)$	$O(N_S)$
Memory of previous states	No	Yes	No	No	No	Yes	Yes	No
Search techniques	Mutation and selection	Mutation and selection	Mutation	Mutation and selection	Mutation and selection	Mutation	Mutation and crossover	Mutation
Convergence speed	Fast	Slow	Fast	Fast	Slow	Slow	Fast	Slow
Scalability	Poor	Good	Good	Poor	Poor	Good	Poor	Good
Multi-Dimension	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Merits	High scalability, good diversity	Easily solve mix variable problems, having feedback system	Can easily escape from local minima, good at exploration	Good convergence speed for high-dimensional problem	Suitable for solving high-dimensional constrained problem	Suitable for solving distributed optimization problem	Good diversity and adaptability	Suitable for solving multidimensional optimization problem with equality and inequality constraints
Limitations	Easily get trapped in local minima for high-dimensional problem	Uncertain convergence time	Not good at exploitation, having low diversity	Enormous number of control parameters	Not able to provide complete optimal solution	Poor convergence for high-dimensional problem	Premature convergence	Poor accuracy, easily trapped in local minima

to maintain a driving force for convergence. Finally, a crossover provides an extended diversity in the search space.

The *convergence speed* refers to the rate at which the algorithm can reach the optimal solution. A good algorithm should have high convergence speed and must not suffer from premature convergence. Premature convergence is defined as the convergence of SI-based algorithm before reaching to a global optimal solution, and it usually occurs due to the lack of diversity [112,113].

The *Scalability* of SI-based algorithm refers to the ability of the algorithm to handle larger number of robots and targets without affecting the overall performance [114]. The main cause behind the scalability is the use of local communication and sensing. Recently in [126], authors have clearly demonstrated the scalability of the SI-based algorithms and proved the sustainability mathematically by considering the case where the targets are stationary.

One of the major features in a swarm of UAVs is the simplicity of each individual UAV, and the ability to accomplish complex tasks by working in coordination in large numbers. Therefore, any SI algorithm should have less computational complexity of the individual UAV so that task can be performed easily with limited hardware resources. Generally, most SI-based algorithms have low computational complexity and they do not depend on the problem size [127]. The computational complexity of PSO based algorithms is relatively low as the computational steps require only individual comparison and vector addition [66,128]. In Bat optimization algorithm, each individual robot calculates its own position with respect to the targets, which results in low computational complexity [89]. Similarly, the ABC-based algorithm in [94] and the PIO-based algorithm in [102], also have low computational complexity. The pheromone update and state update calculations in ACO-based algorithm [79] is computationally more complex compared with PSO, BO, PIO, and ABC. The BFO-based algorithm in [99] has higher computational complexity as it involves the computation of the best position of the individual UAV, computation of high fitness value for maintaining the size of the swarm, and computation of the best neighbor to follow the target by making good formation control. To achieve the optimal solution of the interception problem, the convergence time should be small. THE Basic PSO algorithm has a fast convergence speed, but it can easily get trapped in local minima. To overcome this problem, the authors in [72] proposed an improved version of PSO where the entire swarm of UAVs is divided into subgroups to achieve fast convergence without getting trapped in local optima. An improved version of PIO is proposed by Zhang and Haibin in [103] with mathematical proof which shows the excellent performance of the algorithms in terms of stability and convergence as compared with basic PSO.

In a collaborative task assignment by a swarm of UAVs, there is a risk of collision between the UAVs as well as with obstacles in the operating arena. Therefore, the adoption of an efficient obstacle avoidance method is required to achieve satisfactory performance. The easiest mechanism to avoid obstacles and collisions with other UAVs is to use the on-board sensor as shown in [110,129] that used the Braitenberg obstacle avoidance mechanism. The artificial repulsion is another prominent obstacle avoidance method [130], where the UAV is modeled as a particle moving in a potential field developed by the target and the obstacles present in the environment. The BFO-based approach in [131] adopted the artificial repulsion method to achieve effective obstacle avoidance. ACO-based algorithm in [80] incorporates RPM in order to avoid obstacles in each path segment. In PIO-based approach [104], the UAVs compute the safe path for the interception of a target at each iteration to avoid the obstacles. When a potential collision is detected in between the UAVs, the algorithm computes another path in consecutive iterations until a collision-free path is detected. The problem of path planning and target interception requires multiple optimal solutions in parallel, and also a balance between exploitation and exploration [132]. Exploitation is the phenomenon in which an algorithm generates a novel solution that is far better than the present solution. The main advantage of the exploitation is that it increases the convergence rate of the algorithm. Exploration is the phenomenon in which an algorithm explores the problem search space in an effective way to generate solutions with a diversity [113]. The ACO-based approach [77] and ABC-based approach [94] are both using only two operator mutations and selection for generating the optimum solution which leads to a good global search. However, they have slow convergence speed due to the lack of the crossover operator. Levy

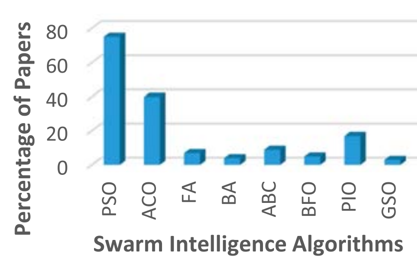

Figure 10: Percentage of papers published for UAV path planning using SI algorithm

flight in FA-based approach [85] introduces a mutation mechanism which ensures the diversity in the population and prevents the algorithm from converging at premature optima.

Figure 10 shows the percentage of papers published for UAV path planning using swarm intelligence algorithms. The graph shows that researchers have widely explored PSO and ACO algorithm for UAV path planning. Since FA, BA, ABC, BFO, PIO, and GSO algorithms have been developed in last few years therefore researchers are still exploring these algorithms for UAV path planning.

7. CONCLUSION

This paper presents the analysis and comparison of swarm intelligence-based algorithms for 3D path planning in interception missions. Various problem setups are studied in the paper, and it is found that the use of a swarm of UAVs for target interception in the environment is a very challenging task. The observations based on the proposed review are defined as follows:

- PSO algorithm has been widely used in path planning for target interception due to its low computational complexity. However, it easily gets trapped in local minima.
- ACO has good scalability but it requires an enormous number of parameters for tuning and this leads to high computational complexity.
- Firefly algorithm uses only one operator for searching the solution and thus has low computational complexity. However, it lacks in terms of exploitation.
- Bat algorithm has high convergence speed and reaches a satisfactory solution in a very fast manner, but it requires tuning of various parameters.
- ABC is very promising for getting the solution of high-dimensional problems, but it suffers from slow convergence speed.
- The number of tuning parameters is significantly less in bacterial foraging algorithm and thus it is very efficient in solving the path planning problem for multiple UAVs, but with the increment in the number of targets, convergence speed slows down.

Therefore, we conclude that a new, improved or hybrid optimization algorithm can be explored in this area which can overcome the existing limitations of the discussed algorithms.

ACKNOWLEDGEMENTS

Authors are thankful to anonymous reviewers and editor for their suggestions.

ORCID

Shraga Shoval http://orcid.org/0000-0002-0582-4821

Abhinav Sharma http://orcid.org/0000-0003-3014-9079

REFERENCES

1. G. Şerban, I. Rus, D. Vele, P. Breţcan, M. Alexe, and D. Petrea, "Flood-prone area delimitation using UAV technology, in the areas hard-to-reach for classic aircrafts: Case study in the north-east of Apuseni Mountains, Transylvania," *Nat. Hazards*, Vol. 82, no. 3, pp. 1817–32, 2016.
2. D. Kingston, S. Rasmussen, and L. Humphrey, "Automated UAV tasks for search and surveillance," in Control Applications (CCA), IEEE Conference. 2016. IEEE.
3. N. Ernest, A. Sathyan, and K. Cohen, "Genetic fuzzy single and collaborative tasking for UAV operations," in Multi-Rotor Platform-based UAV Systems, 2020, Elsevier. pp. 217–42.
4. M. M. Trujillo, M. Darrah, K. Speransky, B. DeRoos, and M. Wathen, "Optimized flight path for 3D mapping of an area with structures using a multirotor in Unmanned Aircraft Systems (ICUAS)," in International Conference. 2016. IEEE.
5. E. Zamora, and W. Yu, "Recent advances on simultaneous localization and mapping for mobile robots," *IETE Tech. Rev.*, Vol. 30, no. 6, pp. 490–6, 2013.
6. G. S. Avellar, G. A. Pereira, L. C. Pimenta, and P. Iscold, "Multi-UAV routing for area coverage and remote sensing with minimum time," *Sensors*, Vol. 15, no. 11, pp. 27783–803, 2015.
7. Available: https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011
8. Available: https://www.thenational.ae/business/japan-uavs-take-the-back-breaking-labour-out-of-farming-1.221693
9. Available: https://www.casa.gov.au/standard-page/flying-drones-australia
10. A. Bürkle, F. Segor, and M. Kollmann, "Towards autonomous micro UAV swarms," *J. Intell. Robot. Syst.*, Vol. 61, no. 4, pp. 339–53, 2011.
11. P. O. Pettersson, and P. Doherty, "Probabilistic roadmap based path planning for an autonomous unmanned helicopter," *J. Intell. Fuzzy Syst.*, Vol. 17, no. 4, pp. 395–405, 2006.
12. L. De Filippis, G. Guglieri, and F. Quagliotti, "Path planning strategies for UAVS in 3D environments," *J. Intell. Robot. Syst.*, Vol. 65, no. 1, pp. 247–64, 2012.
13. O. Cetin, I. Zagli, and G. Yilmaz, "Establishing obstacle and collision free communication relay for UAVs with

artificial potential fields," *J. Intell. Robot. Syst.*, Vol. 69, no. 1, pp. 1–12, 2013.

14. S. Hacohen, S. Shoval, and N. Shvalb, "Äpplying probability navigation function in dynamic uncertain environments," *Rob. Auton. Syst.*, Vol. 87, pp. 237–46, 2017.
15. K. S. Camilus, and V. Govindan, "A review on graph based segmentation," *Int. J. image Graph. Signal Process.*, Vol. 4, no. 5, pp. 1–13, 2012.
16. S. M. Persson, and I. Sharf, "Sampling-based A* algorithm for robot path-planning," *Int. J. Rob. Res.*, Vol. 33, no. 13, pp. 1683–1708, 2014.
17. M. Khosravi, and A. G. Aghdam, "Cooperative receding horizon control for multi-target interception in uncertain environments," in Decision and Control (CDC), 53rd Annual Conference, 2014. IEEE.
18. R. W. Beard, T. W. McLain, M. A. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Trans. Robot. Autom.*, Vol. 18, no. 6, pp. 911–22, 2012.
19. L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of robot 3d path planning algorithms," *J. Control Sci. Eng.*, Vol. 2016, pp. 5, 2016.
20. P. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *IEEE Control Systems*, Vol. 34, no. 1, pp. 42–59, 2014.
21. L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of UAV 3D path planning," in Intelligent Control and Automation (WCICA), 11th World Congress, 2014. IEEE.
22. Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowl. Based. Syst.*, Vol. 158, pp. 54–64, 2018.
23. T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr, "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, Vol. 3, no. 1, p. 4, 2019.
24. S. Aggarwal, and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, Vol. 149, pp. 270–99, 2020.
25. D. Li, P. Wang, and L. Du, "Path planning technologies for autonomous underwater vehicles – A review," *IEEE. Access.*, Vol. 7, pp. 9745–68, 2018.
26. H. Yu, K. Meier, M. Argyle, and R. W. Beard, "Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles," *IEEE/ASME Trans. Mechatron.*, Vol. 20, no. 2, pp. 541–52, 2015.
27. J. Ren, W. Du, and F. Du, "A UAV 3-D space dynamic path planning in complex battlefield environment in information Science and Technology (ICIST)," in International Conference. IEEE, 2013.
28. J. Hu, L. Xie, K. Y. Lum, and J. Xu, "Multiagent information fusion and cooperative control in target search," *IEEE Trans. Control Syst. Technol.*, Vol. 21, no. 4, pp. 1223–35, 2013.
29. H. H. Triharminto, A. S. Prabuwono, T. B. Adji, and N. A. Setiawan, "Adaptive dynamic path planning algorithm for interception of a moving target," *International J. Mob. Comput. Multimed. Commun. (IJMCMC)*, Vol. 5, no. 3, pp. 19–33, 2013.
30. F. Andert, and F. Adolf, "Online world modeling and path planning for an unmanned helicopter," *Auton. Robots.*, Vol. 27, no. 3, pp. 147–64, 2009.
31. M. Sanfourche, J. Delaune, G. Le Besnerais, H. De Plinval, J. Israel, P. Cornic, A. Treil, Y. Watanabe, and A. Plyer, "Perception for UAV: Vision-based navigation and environment modeling," *AerospaceLab*, Vol. 4, pp. 1–19, 2012.
32. Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerosp. Sci. Technol.*, Vol. 6, no. 1, pp. 47–55, 2012.
33. X. Zhao, Y. Zhang, and B. Zhao, "Robust path planning for avoiding obstacles using time-environment dynamic map," *Meas. Control*, Vol. 53, no. 2, pp. 214–21, 2020.
34. S. Arora, S. Choudhury, and S. Scherer. "Hindsight is only 50/50: Unsuitability of MDP based Approximate POMDP Solvers for Multi-resolution Information Gathering". arXiv preprint arXiv:1804.02573, 2018.
35. T. Gorecki, H. Piet-Lahanier, J. Marzat, and M. Balesdent, "Cooperative guidance of UAVs for area exploration with final target allocation," *IFAC Proceedings*, Vol. 46, no. 19, pp. 260–5, 2013.
36. J. Cruz, G. Chen, D. Li, and X. Wang, "Particle swarm optimization for resource allocation in UAV cooperative control," in AIAA guidance, Navigation, and Control Conference and Exhibit. 2004. Providence.
37. P. Sujit, A. Sinha, and D. Ghose, "Multiple UAV task allocation using negotiation," in Proceedings of the fifth International Joint Conference on Autonomous Agents and Multiagent Systems, 2006. ACM.
38. B. Liu, Z. Qin, R. Wang, Y. B. Gao, and L. P. Shao, "A hybrid heuristic ant colony system for coordinated multi-target assignment," *Inf. Technol. J.*, Vol. 8, no. 2, pp. 156–64, 2009.
39. Y. Zhang, W. Feng, G. Shi, F. Jiang, M. Chowdhury, and S. H. Ling, "UAV swarm mission planning in dynamic

environment using consensus-based bundle algorithm," *Sensors*, Vol. 20, no. 8, p. 2307, 2020.

40. Y. Chen, D. Yang, and J. Yu, "Multi-UAV task assignment with parameter and time-sensitive uncertainties using modified Two-part wolf pack search algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. 54, no. 6, pp. 2853–72, 2018.

41. G. Wang, Q. Li, and L. Guo, "Multiple UAVs routes planning based on particle swarm optimization algorithm," in Information Engineering and Electronic Commerce (IEEC), 2nd International Symposium, 2010. IEEE.

42. V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Inf.*, Vol. 9, no. 1, pp. 132–41, 2013.

43. N. Özalp, and O. K. Sahingoz, "Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms," in Unmanned Aircraft Systems (ICUAS), International Conference, 2013. IEEE.

44. E. Besada-Portas, L. de la Torre, M. Jesus, and B. de Andrés-Toro, "Evolutionary trajectory planner for multiple UAVs in realistic scenarios," *IEEE Trans. Robot.*, Vol. 26, no. 4, pp. 619–34, 2010.

45. W. Zhu, and H. Duan, "Chaotic predator–prey biogeography-based optimization approach for UCAV path planning," *Aerosp. Sci. Technol.*, Vol. 32, no. 1, pp. 153–61, 2014.

46. Y. Fu, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *IEEE Trans. Syst., Man, Cybern.-A: Syst. Hum.*, Vol. 42, no. 2, pp. 511–26, 2012.

47. R. Zhu, D. Sun, and Z. Zhou, "Cooperation strategy of unmanned air vehicles for multitarget interception," *J. Guid. Control Dyn.*, Vol. 28, no. 5, p. 1068, 2005.

48. T. Y. Sun, C. L. Huo, S. J. Tsai, Y. H. Yu, and C. C. Liu, "Intelligent flight task algorithm for unmanned aerial vehicle," *Expert. Syst. Appl.*, Vol. 38, no. 8, pp. 10036–48, 2011.

49. T. Y. Sun, C. L. Huo, S. J. Tsai, and C. C. Liu, "Optimal UAV flight path planning using skeletonization and particle swarm optimizer," in Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress, 2008. IEEE.

50. Y. Liu, X. Zhang, X. Guan, and D. Delahaye, "Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization," *Aerosp. Sci. Technol.*, Vol. 58, pp. 92–102, 2016.

51. L. Freda, and G. Oriolo, "Vision-based interception of a moving target with a nonholonomic mobile robot," *Rob. Auton. Syst.*, Vol. 55, no. 6, pp. 419–32, 2007.

52. A. Popescu, and D. Popescu. "Moving target interception for a walking robot by fuzzy observer and fuzzy controller". Climbing and Walking Robots: From Biology to Industrial Applications (CLAWAR 2001), p. 363, 2001.

53. X. Yi, A. Zhu, S. X. Yang, and C. Luo, "A bio-inspired approach to task assignment of swarm robots in 3-D dynamic environments," IEEE Transactions on Cybernetics 2016.

54. L. Zhong, Q. Luo, D. Wen, S. D. Qiao, J. M. Shi, and W. M. Zhang, "A task assignment algorithm for multiple aerial vehicles to attack targets with dynamic values," *IEEE Trans. Intell. Transp. Syst.*, Vol. 14, no. 1, pp. 236–48, 2013.

55. J. Ni, and S. X. Yang, "Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments," *IEEE Trans. Neural Netw.*, Vol. 22, no. 12, pp. 2062–77, 2011.

56. X. Fu, J. LI, and X. Gao, "Target allocation in multi-UAV cooperative search with communication constraints," *Acta Aeronautica et Astronautica Sinica*, Vol. 35, no. 5, pp. 1347–56, 2014.

57. M. Khosravi, and A. G. Aghdam, "Stability analysis of dynamic decision-making for vehicle heading control," in American Control Conference (ACC), 2015 IEEE.

58. A. Pierson, Z. Wang, and M. Schwager, "Intercepting rogue robots: An algorithm for capturing multiple evaders with multiple pursuers," *IEEE Robot. Autom. Lett.*, Vol. 2, no. 2, pp. 530–7, 2017.

59. Y. Lan, "Multiple mobile robot cooperative target intercept with local coordination," in Control and Decision Conference (CCDC), 24th Chinese, 2012. IEEE.

60. P. K. Sheridan, P. Kosicki, C. Liu, G. Nejat, and B. Benhabib, "On-line task allocation for the robotic interception of multiple targets in dynamic settings," in Advanced Intelligent Mechatronics (AIM), IEEE/ASME International Conference, 2010. IEEE.

61. X. Fu, H. Feng, and X. Gao, "UAV mobile ground target pursuit algorithm," *J. Intell. Robot. Syst.*, Vol. 68, no. 3, pp. 1–13, 2012.

62. S. Manzoor, S. Lee, and Y. Choi, "A coordinated navigation strategy for multi-robots to capture a target moving with Unknown speed," *J. Intell. Robot. Syst.*, Vol. 87, no. 3, pp. 1–15, 2016.

63. M. Kothari, R. Sharma, I. Postlethwaite, R. W. Beard, and D. Pack, "Cooperative target-capturing with incomplete target information," *J. Intell. Robot. Syst.*, Vol. 72, no. 3, p. 373, 2013.

64. A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, Vol. 48, no. 6, pp. 988–1001, 2003.

65. Z. Ming, Z. Lingling, S. Xiaohong, M. Peijun, and Z. Yanhang, "Improved discrete mapping differential evolution for multi-unmanned aerial vehicles cooperative multi-targets assignment under unified model," *Int. J. Mach. Learn. Cybern.*, Vol. 8, no. 3, pp. 765–780, 2017.

66. R. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory. in micro Machine and human science, 1995," in MHS'95., Proceedings of the Sixth International Symposium, Italy, 1995.

67. J. Kennedy, and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in Systems, Man, and Cybernetics, Computational Cybernetics and Simulation., IEEE International Conference, 1997. IEEE.

68. Y. Shi, "Particle swarm optimization: developments, applications and resources," in Evolutionary computation, Proceedings of the Congress, 2001. IEEE.

69. Y. Shi, and R. Eberhart, "A modified particle swarm optimizer," in Evolutionary Computation proceedings, IEEE World Congress on Computational Intelligence., The IEEE International Conference, 1998. IEEE.

70. G. Oh, Y. Kim, J. Ahn, and H. L. Choi, "PSO-based optimal task allocation for cooperative timing missions," *IFAC-PapersOnLine*, Vol. 49, no. 17, pp. 314–9, 2016.

71. T. Flanagan, C. Thornton, and J. Denzinger, "Testing harbour patrol and interception policies using particle-swarm-based learning of cooperative behavior," in Computational Intelligence for Security and Defense Applications, CISDA. IEEE Symposium, 2009. IEEE.

72. X. Xing, D. Fan, Y. Zhao, and L. Huang, "PSO-based multi UCAVs cooperative attack tasks allocation and its simulation," in Natural computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 12th International Conference, 2016. IEEE.

73. S. Shao, Y. Peng, C. He, and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Trans.*, Vol. 97, pp. 415–30, 2020.

74. S. Girija, and A. Joshi, "Fast hybrid PSO-APF algorithm for path planning in obstacle rich environment," *IFAC-PapersOnLine*, Vol. 52, no. 29, pp. 25–30, 2019.

75. M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artif. Life*, Vol. 5, no. 2, pp. 137–72, 1999.

76. D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K. V. Price, *New ideas in optimization*. London: McGraw-Hill Ltd., 1999.

77. H. B. Duan, "Ant colony algorithms: Theory and applications," *Chin. Sci.*, Vol. 31, no. 4, pp. 167–178, 2005.

78. T. Zaza, and A. Richards, "Ant colony optimization for routing and tasking problems for teams of UAVs," in Control (CONTROL), UKACC International Conference, 2014. IEEE.

79. C. Gao, Z. Zhen, and H. Gong, "A self-organized search and attack algorithm for multiple unmanned aerial vehicles," *Aerosp. Sci. Technol.*, Vol. 54, pp. 229–40, 2016.

80. H. Huang, Y. Wang, H. Zhou, K. Dong, and H. Liu, "Multi-UCAV cooperative autonomous attack path planning method under uncertain environment," in Advanced information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016. IEEE.

81. Z. Zhen, C. Yan, W. Liangdong, and H. Bing, "An intelligent cooperative mission planning scheme of UAV swarm in uncertain dynamic environment," *Aerosp. Sci. Technol.*, Vol. 100, pp. 105826, 2020.

82. Z. Ma, H. Gong, and X. Wang, "An UAV path planning method in Complex Mountainous Area based on a new improved Ant Colony Algorithm," in 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), 2019. IEEE.

83. X. Yang. "Nature-inspired metaheuristic algorithms Firefly algorithm", pp. 79–90, 2011.

84. X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, Vol. 2, no. 2, pp. 78–84, 2010.

85. C. Liu, Z. Gao, and W. Zhao, "A new path planning method based on firefly algorithm," in Computational Sciences and Optimization (CSO), Fifth International Joint Conference, 2012. IEEE.

86. G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "A modified firefly algorithm for UCAV path planning," *Int. J. Hybrid Inf. Technol.*, Vol. 5, no. 3, pp. 123–44, 2012.

87. X. S. Yang. "A new metaheuristic bat-inspired algorithm". Nature inspired cooperative strategies for optimization (NICSO 2010), pp. 65–74.

88. X. S. Yang, and X. He, "Bat algorithm: literature review and applications," *Int. J. Bio-Inspired Comput.*, Vol. 5, no. 3, pp. 141–9, 2013.

89. J. Guo, Y. Gao, and G. Cui, "The path planning for mobile robot based on bat algorithm," *Int. J. Autom. Control*, Vol. 9, no. 1, pp. 50–60, 2015.

90. G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "A bat algorithm with mutation for UCAV path planning," *Scientific World J.*, Vol. 2012, pp. 1–16, 2012.

91. K. Li, Y. Han, F. Ge, W. Xu, and L. Liu, “Tracking a dynamic invading target by UAV in oilfield inspection via an improved bat algorithm,” *Appl. Soft. Comput.*, Vol. 90, p. 106150, 2020.

92. D. Karaboga. “An idea based on honey bee swarm for numerical optimization”, Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

93. D. Karaboga, “Artificial bee colony algorithm,” *Scholarpedia*, Vol. 5, no. 3, pp. 6915, 2010.

94. B. Li, L. Gong, and C. Zhao, “Unmanned combat aerial vehicles path planning using a novel probability density model based on artificial bee colony algorithm,” in Intelligent Control and Information Processing (ICICIP), Fourth International Conference, 2013. IEEE.

95. T. S. Pan, T. K. Dao, and J. S. Pan, “An unmanned aerial vehicle optimal route planning based on compact artificial Bee colony,” in Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceeding of the twelfth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2016, Kaohsiung, Taiwan, Vol. 2, 2017, pp. 21–3.

96. B. Li, L.-G. Gong, and W.-L. Yang, “An improved artificial bee colony algorithm based on balance-evolution strategy for unmanned combat aerial vehicle path planning,” *Scientific World J.*, Vol. 2014, pp. 1–11, 2014.

97. K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” *IEEE Control Syst.*, Vol. 22, no. 3, pp. 52–67, 2002.

98. S. Das, A. Biswas, S. Dasgupta, and A. Abraham, “Bacterial foraging optimization algorithm: Theoretical foundations, analysis, and applications,” *Found. Comput. Intell.*, Vol. 3, pp. 23–55, 2009.

99. S. J. Yang, S. W. Wang, J. Tao, P. Wen, and W. Zhou, “Multi-ucav cooperative task allocation in dynamic environment,” *Electron. Opt. Cont.*, Vol. 19, no. 7, pp. 32–6, 2012.

100. B. Yang, Y. Ding, Y. Jin, and K. Hao, “Self-organized swarm robot for target search and trapping inspired by bacterial chemotaxis,” *Rob. Auton. Syst.*, Vol. 72, pp. 83–92, 2015.

101. Y. Long, Z. Zuo, Y. Su, J. Li, and H. Zhang, “An A*-based bacterial foraging optimisation algorithm for global path planning of unmanned surface vehicles,” *J. Navig.*, Vol. 0, pp. 1–16, 2020.

102. H. Duan, and P. Qiao, “Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning,” *Int. J. Intell. Comput. Cybern.*, Vol. 7, no. 1, pp. 24–37, 2014.

103. B. Zhang, and H. Duan, “Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment,” *IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB)*, Vol. 14, no. 1, pp. 97–107, 2017.

104. C. Li, and H. Duan, “Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function,” *Aerosp. Sci. Technol.*, Vol. 39, pp. 352–60, 2014.

105. M. S. Dao, F. G. De Natale, and A. Massa, “Edge potential functions (EPF) and genetic algorithms (GA) for edge-based matching of visual objects,” *IEEE Trans. Multimed.*, Vol. 9, no. 1, pp. 120–35, 2007.

106. R. Hao, D. Luo, and H. Duan, “Multiple UAVs mission assignment based on modified pigeon-inspired optimization algorithm,” in Guidance, Navigation and Control Conference (CGNCC), 2014. IEEE.

107. B. H. Wang, D. B. Wang, and Z. A. Ali, “A Cauchy mutant pigeon-inspired optimization–based multi-unmanned aerial vehicle path planning method,” *Meas. Control*, Vol. 53, pp. 20–34, 2020.

108. C. Hu, Y. Xia, and J. Zhang, “Adaptive operator quantum-behaved pigeon-inspired optimization algorithm with application to UAV path planning,” *Algorithms*, Vol. 12, no. 1, p. 3, 2019.

109. K. Krishnanand, and D. Ghose, “Detection of multiple source locations using a glowworm metaphor with applications to collective robotics,” in Swarm intelligence symposium, Proceedings 2005 IEEE.

110. K. N. Krishnanand, P. Amruth, M. H. Guruprasad, S. V. Bidargaddi, and D. Ghose, “Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources,” in Robotics and Automation, ICRA, Proceedings IEEE International Conference on, 2006. IEEE.

111. Z. Tang, and Y. Zhou, “A glowworm swarm optimization algorithm for uninhabited combat air vehicle path planning,” *J. Intell. Syst.*, Vol. 24, no. 1, pp. 69–83, 2015.

112. M. S. Couceiro, P. A. Vargas, R. P. Rocha, and N. M. Ferreira, “Benchmark of swarm robotics distributed techniques in a search task,” *Rob. Auton. Syst.*, Vol. 62, no. 2, pp. 200–13, 2014.

113. X. S. Yang, “Swarm intelligence based algorithms: A critical analysis.” *Evol. Intell.*, Vol. 7, no. 1, pp. 17–28, 2014.

114. L. Bayindir, and E. Şahin, “A review of studies in swarm robotics,” *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 15, no. 2, pp. 115–47, 2007.

115. D. P. Rini, S. M. Shamsuddin, and S. S. Yuhaniz, “Particle swarm optimization: technique, system and challenges,” *Int. J. Comput. Appl.*, Vol. 14, no. 1, pp. 19–26, 2011.

116. L. Guangshun, and S. Hongbo, "Path planning for mobile robot based on particle swarm optimization," *Robot*, Vol. 26, no. 3, pp. 222–5, 2008.

117. D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, Vol. 42, no. 1, pp. 21–57, 2014.

118. Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Math. Probl. Eng.*, Vol. 2015, pp. 1–38, 2015.

119. L. N. De Castro, *Fundamentals of natural computing: Basic concepts, algorithms, and applications*. Brazil: CRC Press, 2006.

120. G. G. Wang, H. E. Chu, and S. Mirjalili, "Three-dimensional path planning for UCAV using an improved bat algorithm," *Aerosp. Sci. Technol.*, Vol. 49, pp. 231–8, 2016.

121. C. Zhang, Z. Zhen, D. Wang, and M. Li, "UAV path planning method based on ant colony optimization," in Control and Decision Conference (CCDC), 2010. IEEE.

122. H. B. Duan, X. Y. Zhang, J. Wu, and G. J. Ma, "Max-min adaptive ant colony optimization approach to multi-UAVs coordinated trajectory replanning in dynamic and uncertain environments," *J. Bionic Eng.*, Vol. 6, no. 2, pp. 161–73, 2009.

123. X. S. Yang, and X. He, "Firefly algorithm: Recent advances and applications," *Int. J.arm Intell.*, Vol. 1, no. 1, pp. 36–50, 2013.

124. B. Zhang, and H. Duan, "*Predator-prey pigeon-inspired optimization for UAV three-dimensional path planning*," in International Conference in Swarm Intelligence, 2014. Springer.

125. L. Huang, H. Qu, P. Ji, X. Liu, and Z. Fan, "A novel coordinated path planning method using k-degree smoothing for multi-UAVs," *Appl. Soft. Comput.*, Vol. 48, pp. 182–92, 2016.

126. N. Palmieri, X. S. Yang, F. De Rango, and S. Marano, "Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption," *Neural Comput. Appl.*, Vol. 31, no. 1, pp. 1–24, 2017.

127. A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi. "Metaheuristic Applications in Structures and Infrastructures: 1". Metaheuristic Algorithms in Modeling and Optimization 2013: Elsevier Inc. Chapters.

128. P. Sujit, J. George, and R. Beard. "Multiple UAV task allocation using particle swarm optimization". AIAA Paper, pp. 2008–6837, 2008.

129. J. Pugh, and A. Martinoli, "Inspiring and modeling multi-robot search with particle swarm optimization," in Swarm Intelligence Symposium, 2007. IEEE.

130. K. Chang, Y. Xia, and K. Huang, "UAV formation control design with obstacle avoidance in dynamic three-dimensional environment," *Springerplus.*, Vol. 5, no. 1, pp. 1–16, 2016.

131. M. Turduev, M. Kirtay, P. Sousa, V. Gazi, and L. Marques, "Chemical concentration map building through bacterial foraging optimization based search algorithm by mobile robots," in Systems man and Cybernetics (SMC), IEEE International Conference on, 2010. IEEE.

132. D. Parrott, and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *IEEE Trans. Evol. Comput.*, Vol. 10, no. 4, pp. 440–58, 2006.

AUTHORS

Abhishek Sharma received the bachelor's degree in Electronics and Communication Engineering from ITM-Gwalior, India, in 2012, and the master's degree in Robotics Engineering from the University of Petroleum and Energy Studies (UPES), Dehradun, India, in 2014. He was a Senior Research Fellow in a DST funded project under the Technology Systems Development Scheme. He was an Assistant Professor with the Department of Electronics and Instrumentation, UPES. He is working as a research scientist in research and development department, UPES. His research interests include embedded system, optimization, swarm intelligence and robotics.

Email: abhishek15491@gmail.com

Shraga Shoval was born in Ramat-Gan, Israel. He studied at "Ohel Shem" high school and received his BSc (1985) and MSc (1987) in Mechanical Engineering from the Technion, Israel Institute of Technology. Title of his MSc thesis: Integration of a CNC Lathe and a Robot in a Flexible Manufacturing Cell. Advisor: Prof. Ehud Lenz. Between 1987 and 1990 Shraga worked as a scientist in the Commonwealth Scientific and Industrial Research Organization (CSIRO) – Division of Manufacturing Technologies in Sydney, Australia, developing a robotic system for the processing of colored gemstones. He completed his PhD in the University of Michigan, Ann Arbor, at the department of Mechanical Engineering and Applied Mechanics in 1994. Title of his PhD thesis: Information Transfer in Sensor Based Travel Aid for the Blind. Advisors: Prof. Yoram Koren and Prof. Johann Borenstein. The NavBelt, a navigation device for the blind and visually impaired using mobile robotics technology, that was developed based on this thesis, is one of the first robotic aids for the blind. The NavBelt lead to the development of the GuideCane, that was awarded the best robotics invention by Discover Magazine in 1998.

Email: shraga@ariel.ac.il

Abhinav Sharma received his BTech from H. N. B. Garhwal University, Srinagar, India, in 2009 and the MTech from Govind Ballabh Pant University of Agriculture and Technology, Pantnagar, India, in 2011. He did his PhD from Govind Ballabh Pant University of Agriculture and Technology, Pantnagar, India, in 2016. He is working as an Assistant Professor (Senior Scale) in the Department of Electrical and Electronics Engineering in University of Petroleum and Energy Studies, Dehradun. His field of interest includes adaptive array signal processing, Artificial Intelligence, and Machine Learning, Optimization Techniques and smart antenna.

Corresponding author. Email: abhinavgbpuat@gmail.com

Jitendra Pandey was born in Bakeware, Uttar Pradesh, India. He received his BSc (Biochemistry, 1996) and MSc (Organic Chemistry, 1998), from the University of Kanpur, India. He has received is PhD degree from National Chemical Laboratory Pune, in the area of polymer science and his thesis title was "Degradability of Polymer Composites from Renewable Resources". He worked as business development manager-International Business SRL Ranbaxy Ltd., Andheri-East, India. He is working as a Professor and Dean School of Basic & Applied Science, Adamas University, West Bengal, India. His area of expertise is Polymer nanocomposites, Bio-composites, Natural Nano-particles, Water treatment.

Email: jeetusnu@gmail.com