Journal of the Brazilian Society of Mechanical Sciences and Engineering (2022) 44:56
https://doi.org/10.1007/s40430-022-03362-9

TECHNICAL PAPER

Obstacle avoidance for a swarm of unmanned aerial vehicles operating on particle swarm optimization: a swarm intelligence approach for search and rescue missions

Girish Kumar[1] · **Arham Anwar**[1] · **Abhinav Dikshit**[1] · **Abhirup Poddar**[1] · **Umang Soni**[2] · **Weon Keun Song**[3]

Received: 18 April 2021 / Accepted: 3 January 2022 / Published online: 17 January 2022

Abstract
An approach, based on a multi-plane system, is conceptualized in this work to solve the problem of collision avoidance for a swarm of unmanned aerial vehicles, being used for search and rescue to minimize affecting the searching algorithm. Relevant chronological advancements in the last two decades of the parent algorithm, particle swarm optimization, are summarized. As each optimization algorithm for search and rescue has its own niche area of application, various well-established algorithms such as particle swarm optimization and novel algorithms like layered search and rescue, spiral search and fish-inspired task allocation are compared with each other qualitatively. Simulations with 100 different cases were used to compare the original particle swarm optimization with the additional novel collision avoidance algorithm. The statistical z test was run based on which it was found that the proposed algorithm significantly reduces the number of collisions and does not put a toll on the iterations to convergence. Standardized residuals of all cases indicate minimal error difference in the optimum average fitness value calculated by the particle swarm optimization, with and without the conceptualized anti-collision algorithm.

Keywords Swarm robotics · Search and rescue missions · Unmanned aerial vehicles · Particle swarm optimization · Obstacle avoidance

1 Introduction

A search and rescue mission poses critical challenges as every passing second is important when life is at stake. The survival of affected people is dependent on the success of search and rescue missions. Such missions are often employed when unforeseen events such as natural calamities and other disasters occur. According to the Centre of Research on the Epidemiology of Disasters and United Nations Office for Disaster Risk Reduction, 2021, in the span of the last two decades from 2000 to 2019, a total of 7348 natural hazard-related disasters were reported, leading to 1.23 million deaths, 4.03 billion people affected and a loss of 2.97 trillion US dollars [1]. Search and rescue missions take place to save as many lives as possible, wherein the key objective is to find as many targets as possible in the least amount of time. In many challenging cases, on-ground search becomes infeasible and an aerial approach is opted. Nuclear decay zones, regions prone to natural calamities and military hot zones can particularly be infeasible to penetrate via direct human intervention. The use of unmanned aerial vehicles (UAVs) in these cases serves as a plausible solution.

Technical Editor: Rogério Sales Gonçalves.

✉ Girish Kumar
girish.kumar154@gmail.com

Arham Anwar
arhamanwar_2k17me59@dtu.ac.in

Abhinav Dikshit
abhinavdikshit_2k17me08@dtu.ac.in

Abhirup Poddar
abhiruppoddar_2k17me09@dtu.ac.in

Umang Soni
umangsoni.1@gmail.com

Weon Keun Song
bauman98@naver.com

1 Delhi Technological University, Delhi, India

2 Netaji Subhas University of Technology, Delhi, India

3 School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

Search and rescue (SAR) missions have seen a burgeoned potential of UAVs, wherein the limitations of human search groups are bridged by autonomous search and rescue robot units. However, the scope of single UAV search operations is severely limited to the drone's layered search and rescue (LSAR) algorithm follows the key limitations. A single UAV search and rescue (UAV–SAR) operation has a limited range of coverage for a fixed amount of time. In other words, the scope of single UAV searching is restricted to the path it follows. However, in a group of UAVs employed for search and rescue missions the scope is extended to the combined paths of all UAVs. Moreover, sending a single drone to a hot spot will drastically reduce the success rate of the operation. These limitations are overcome by using multiple UAVs instead of one, operating coherently as a system to perform search and rescue missions.

Many researchers, recently, have proposed SAR algorithms where a swarm of unmanned aerial vehicles is used to optimize the performance of search and rescue operations. Layered search and rescue (LSAR) algorithm for a swarm of SAR-UAVs, for the case where the centre of disaster is accurately known, was proposed by Alotaibi et al. [2]. LSAR algorithm follows the key idea that in natural disasters there is a centre where most of the survivors are located and the algorithm focuses more attention on the known centre and gradually less attention with increasing distance from the known centre of the disaster. A spiral search algorithm for the case of immediate search after big natural disasters, in which UAVs would spiral around each identified point, was suggested by Arnold et al. [3]. Each algorithm has its niche area wherein it works the best.

One popular root algorithm is particle swarm optimization (PSO), which has been modified numerous times to improve its efficiency for SAR missions using a swarm of UAVs, wherein each UAV acts as a PSO swarm particle to search for targets. PSO is popular because of its advantages, which are listed by Lee et al. [4], as relatively low computational complexity, easy to code, less sensitivity to the nature of the objective function. PSO also has a limited number of parameters including only the inertia weight factor and two acceleration coefficients in comparison with other competing heuristic optimization methods, and the impact of parameters on the solutions is considered to be less sensitive compared to other heuristic algorithms. Many PSO implementations, however, have either taken the assumption of no collisions and obstacles or have integrated an obstacle avoidance algorithm which significantly affects the searching algorithm's function by changing the trajectories of swarm particles in directions in which fitness value is modified. In this document, an obstacle and collision avoidance algorithm is proposed which is integrated with particle swarm optimization without theoretically affecting the original algorithm's functioning.

The paper presents an exhaustive literature review in Sect. 2 which summarizes the classification of UAV-SAR cases and lists a few solutions for each case followed by a summary of relevant PSO advancements and modifications as a UAV-SAR algorithm and the research gaps. Section 3 explains the methodology wherein the collision and obstacle avoidance algorithms are explained in detail. Section 4 explains the simulation of the integration of collision and obstacle avoidance algorithm with PSO. Section 5 discusses the results and implications of the model followed by Sect. 6 as a conclusion which contains limitations of the work, future directions and application extensions.

2 Literature review

The literature review is divided into two sections. Initially, Sect. 2.1 explicates the scope of search and rescue problems using unmanned aerial vehicles, provides a summary of relevant search and rescue unmanned aerial vehicle solutions and explains the classification of search and rescue problems. Section 2.2 enlists related works, explains the capacity of particle swarm optimization as an algorithm for search and rescue missions using UAVs, and highlights the necessary adjustment that needs to be addressed for working of PSO in a physical search space using unmanned aerial vehicles as particles of PSO.

2.1 Search and rescue problems

The domain of search and rescue problem (SAR) envelopes the activities of searching for people in distress or imminent danger and the provision of aid to them. Based on the search area location, environment and size, search and rescue missions are classified as: maritime search and rescue (MSAR), combat search and rescue (CSAR), urban search and rescue (USAR) and wilderness search and rescue (WiSAR) [5]. Figure 1 shows the general search and rescue operation workflow. This workflow addresses the factors corresponding to the classification, which are search area location, environment and size, and further decides which category of SAR the case belongs to. If the target position is known, rescue plans are made. If the targets are not known, search execution takes place via the most suitable searching algorithm for the case.

2.1.1 Maritime search and rescue

Maritime search and rescue (MSAR) involve cases where objectives are lost in natural water bodies such as the sea and ocean. When it comes to tackling maritime search and rescue missions, three major constraints limit the usage of unmanned aerial vehicles [6]. First of all, the flight time

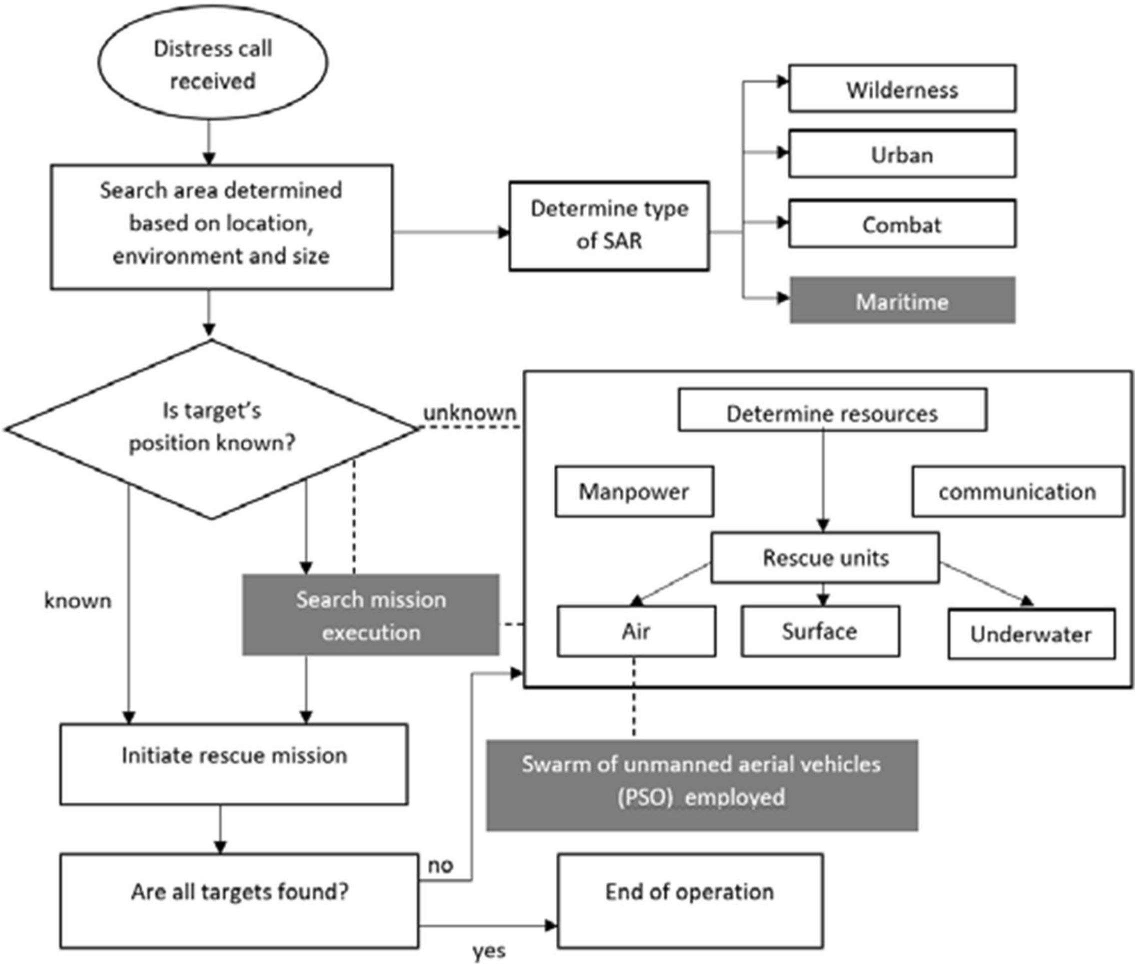


Fig. 1 General operation workflow of search and rescue missions

of commercial unmanned aerial vehicles is limited due to battery or capacity. Moreover, the dynamicity and uncertainty of survivor positions and the need for automated control of fuel service stations make the case furthermore challenging. These challenges were addressed by Lee et al. [6], using the mixed-integer linear programming (MILP) efficiently. Using the cumulative knowledge of last seen locations, cooperative game theory to enable cognitive multi-unmanned aerial vehicles, in an attempt to achieve optimization based on unmanned aerial vehicles with on-board computers, was applied by Rahmes et al. [7]. While the model seems robust for cases where last seen locations are accurately known, the model fails in general cases where this information is not known. A combination of Gaussian mixture models and Fourier transformations for detection of survivors was employed by Dinnbier et al. [8]. However, this algorithm heavily suffers around coastal regions since the algorithm only works for the case where there is no land in the field of vision of UAVs.

2.1.2 Urban search and rescue

Human teams start looking for survivors manually in times of disasters such as an earthquake or terrorist attack. Robots are integrated with human teams since they can efficiently squeeze into tight spaces and reach dangerous areas. Urban search and rescue (USAR) envelopes location, extrication and initial medical stabilization of victims trapped in an urban area. These missions can further be classified as accidental and deliberate, based on the cause. If the scenario is caused by say a terror group, the case is referred to as deliberate USAR which demands a different set of skills from UAVs as compared to simple to an accidental USAR. All other cases correspond to accidental USAR. Technical aspects of autonomous USAR robots were presented by Kohlbrecher et al. [9]. The solution focused on two aspects which were: survivor search and detection, and navigation towards the goal. The first aspect can be solved if the system is as close to the survivors as possible even if the sensors cover only a small portion of the survivor. This proved to be a deterrent to most survivor locations. For navigation towards the goal, the robot observes the survivor it is directly facing. Therefore, the robot has to turn around in each direction for detection. Cui et al. [10] proposed a solution to mitigate the dynamicity which involves real-time image stitching, indoor navigation, vision-based and poses estimation with multiple cooperative MAVs. The implementation methodology had two phases—first, MAVs explore and search the area for the algorithm to aerially photograph and stitch a map together, and second, they search and identify each individual point of interest. The proposed solution, however, is not fully autonomous because a human operator is required to manually stitch the map together for an obstacle-free route for the MAVs. Moreover, in the next phase, an operator is required to identify the victim's house based on the feedback received from the MAV's camera.

2.1.3 Wilderness search and rescue

The search process in which people are lost or are in distress in the wilderness is known as wilderness SAR (WiSAR). Subjected to potentially toxic environments, medical UAVs can be employed in the operations concerning locating targets, emergency medical care and swift transfer of patients. Grissom et al. [11] show how unmanned aerial vehicles also provide information about the area of operation which can be used for future missions. A camera-based position detection system for SAR operations was developed and integrated into UAVs by Sun et al. [12]. The system identified real-time targets, post-targets and also photographed the disaster areas for subsequent operations. Al-Kaff et al. [13] proposed a model WiSAR where operations were handled by mainly protecting human life in risky and unsafe environments using UAVs. The authors presented a human body detection and tracking algorithm. This was accomplished with an on-board sensor on the UAV that captured colour and depth data followed by validation of their work in both real and simulated environments along with an ability to detect multiple survivors.

Nattero et al. [14] compared the real-time performance of multi-chopper systems, developed an autonomous multi-drone SAR strategy and categorized the algorithms based on energy consumed and time-to-completion. Arnold et al. [3] proposed a spiral search method for search and rescue via UAVs after a big natural disaster, which has very high area coverage rates with respect to time.

2.1.4 Combat search and rescue

SAR operations carried out during war are called combat SAR. An auction-based approach, developed by Wie et al. [15], was used to solve SAR task priority problems where tasks were assigned to each robot based on techniques for deciding winner tasks. When a robot was not in a condition to operate, the task was assigned to idle robots. The completion time and required steps were measured along with the energy consumed. The disadvantage of this algorithm is that it is assumed that the robots have knowledge of the previous tasks, environment and the initial location of the survivors. However, this contrasts with one of the most frequent conditions in search and rescue operations, that is survivor locations are neither known nor static. Table 1 contains a brief list of solutions for unmanned aerial vehicle search and rescue missions, classified according to the type of search and rescue, that is, MSAR, USAR, WiSAR and CSAR.

2.2 Particle swarm optimization for unmanned aerial vehicles

A basic variant of the particle swarm optimization (PSO) algorithm works by having a swarm of candidate solutions or particles. Many alterations have been made in the past for a plethora of use cases and environments. Many new metaheuristic algorithms with unique features have been proposed in the recent past, and the proposed multi-plane approach of collision and obstacle avoidance is the extension of these. Some examples of new metaheuristic algorithms include the African vulture optimization [16], which features a unique exploration and exploitation phase split for searching, and the African gorilla optimization [17], which specializes in global optima problems of higher dimensions. Fish-inspired algorithm for multiple-UAV task allocation, or FIAM [18], features low computational task allocation aiming to minimize search time. However, PSO is a robust and widely tested metaheuristic algorithm is used as the base algorithm in this paper. Some of the relevant UAV adaptations of PSO modifications are enlisted in Table 2, while Table 3 enlists some other popular algorithms developed for the particular use case of search and rescue.

2.3 Advantages of the algorithm proposed

There are several studies done in the past on modifications and limitations of the PSO algorithm. However, in light of search and rescue missions with UAVs as PSO particles, collisions avoidance is absent in many proposals. dPSO-U for exploring disaster scenario area with UAVs as PSO particles was simulated and compared to the lawn mower algorithm, which sweeps the entire scenario area, by Garcia et al [19]. Lawn mower algorithm was found to be slower than dPSO-U. These particles can be considered immune to collision if it is assumed that each particle is an abstract entity that can pass through another without hindering the motion of either particle. However, UAVs are neither infinitesimally small nor are abstract entities; thus, in systems where multiple drones

Table 1 Summary of UAV solutions for each classification of SAR

Reference	Type	Algorithm used	Remarks	Advantages
[6]	MSAR	Mixed-integer linear programming	developed a method to efficiently plan the tasks for an unmanned aircraft system (UAS)	Features fuel station solution for battery capacity
[8]	MSAR	Gaussian mixture models and Fourier transforms	Efficient detection of survivors with exceptionally low false positives, albeit reliant on strict assumptions	A robust survivor detection model with very low false positive for the given assumptions
[10]	USAR	Real-time image stitching	Accuracy relies on image stitching and manual detection of survivors	A two-phase searching algorithm that stitches images of MAVs to create a map for an obstacle-free search space—mitigating dynamicity of targets
[3]	USAR	Spiral method	The modular algorithm has potential to be fused with other algorithms to optimize searching; Ideal for congregated collective behavioural positions	Can be used a modular algorithm to fast track convergence
[12]	WiSAR	All-in-one camera-based target detection and positioning algorithm	Camera-based target identification algorithm proposed relies on the assumption of linear scales of the camera and world coordinates	UAVs realize searching and geo-information acquisition in a single flight—features accurate position detection
[13]	WiSAR	Multi-object tracking algorithm and semi-autonomous reactive control	Specially designed to cope with aerial sequences, where the appearance, position and orientation of the objects are continuously changing, and the use of the traditional trajectory-based tracking is not viable	Provides information about the survivors in unfriendly environments and assists the pilot to perform smooth and safe approaching manoeuvres to the detected points
[15]	CSAR	Auction-based novel algorithm	There is a trade-off between fuel consumption and minimum search time for different application domains when using the algorithm	An auction-based approach and a novel prediction approach to dynamic task allocation
[2]	Known centre of disaster	Layered search and rescue	Efficiency of the algorithm relies on whether the centre of disaster is known accurately	A novel task distribution technique for SAR scenarios involving multiple autonomous UAVs

Table 2 Summary of chronological and relevant PSO modifications

Reference	Algorithm	Key results	Chronological advancement	Swarm size	Remarks
Hereford et al. [27]	dPSO	Improved average search time with sensor and position corrections	PSO with basic obstacles and one target	3	Finding the target (the brightest point of light in the search space) in the presence of obstacles
Peng et al. [30]	PSO + dynamic weights	15.5% less Voyage as compared to pure PSO	Dynamic weights for PSO algorithm parameters	20	Model predictive control with partially known environment information
Geng et al. [31]	PSO + contraction factor	Fuel cost lower from 49 to 40 units in the modified algorithm	Developed contraction factor—increased convergence rate and convergence speed	30	Used contraction factor to mitigate convergence into local minima
Loscri et al. [32]	PSO-S	For a varying coverage threshold, the algorithms show a threshold effect around 0.7	Dynamic Search Space	30–80	Integration wtih virtual forces algorithm
Ghamry et al. [21]	PSO-CTPD	Zero collisions for forest fire trajectory planning	Minimized travelling distance with obstacle avoidance	6	Integration with control parameterization and control descretization
Zhang et al. [23]	Per-drone iterated PSO	Coverage by PSO with 800 UAVs was achieved with 200 UAVs using the proposed algorithm	Analysis of drone–cell user coverage and feasible working zone by leveraging drone-to-user and drone-to-base pathloss models	10	maximizing the user coverage while maintaining D2B link qualities, for a given number of drone cells being deployed
Garcia et al. [19]	dPSO	Discovers 25%, 50% and 75% of targets faster than lawn mower algorithm	Tailored for disaster scenario area, UAVs follow a DTN networking approach	6	Finding victim clusters post-disasters in minimum time
Abdollahzadeh et al. [16]	PSO-GA, PSO-HSA	7% less distance traversed compared to PSO	Hybridization with GA and HAS, respectively	100	Reduced traversal time and fuel consumed

operate in close proximity, chances of collisions are high. This brings forth the need for a collision avoidance system that is incorporated in the original PSO algorithm. The proposed algorithm features this advancement of integration of collision avoidance without compromising on the searching efficiency of the original PSO. This modification allows for smooth and faster convergence of UAV particles with significantly reduced collisions.

Many strategies that enforce a minimum distance between UAVs add additional constraints (of minimum distance) to the optimization problem that complicate the original problem [20]. A strategy that incorporates obstacle and collision avoidance without changing the trajectory of the particles and the optimality conditions would provide to be more fruitful [21]. A real-time path planning strategy for UAV based on improved particle swarm optimization was proposed by Cheng et al. [22].

Based on these research gaps, in this paper, integration of a novel obstacle and collision avoidance mechanism into the searching algorithm is proposed in this paper, which theoretically does not affect the searching algorithm significantly and makes the employment of UAVs for search and rescue missions a step closer to practical use.

Table 3 Some other popular algorithms used for SAR via a swarm of UAVs

Reference	Algorithm	Swarm size	Key deliverables	Robot employed	# targets	Objective
[33]	Genetic algorithm	3–6	Proposed Simultaneous Inform and Connect strategy which offers shorter mission time compared to inform-first and connect-first strategies	UAV	–	To minimize the mission completion time, which includes area coverage and network connectivity
[3]	Spiral	1–20	With the development of spiral search algorithm, an area can be searched at very fast rates which increase rapidly with an increase in deployed UAVs	UAV	–	To gather maximum situational awareness data during the first few hours after a major natural disaster UAVs With spiral search, 98.8%
[2]	LSAR	2–128	Developed LSAR algorithm and compared the same with other algorithms in terms of rescued survivors and rescue and execution times with max-sum, auction-based and locust-inspired approaches for multi-UAV task allocation	UAV	2–4096	Minimizing the SAR total time while saving the maximum number of people
[18]	FIAM	4–128	Novel low computational algorithm to address the challenges of task allocation problem	UAV	2–4096	Minimizing mean rescue time

3 Methodology

Section 3.1 provides a brief overview of how the PSO algorithm for searching via unmanned aerial vehicles. Section 3.2 explains the novel collision avoidance algorithm introduced, and Sect. 3.3 explains the implementation of the same. Section 3.4 follows the special cases, and Sect. 3.5 explains the algorithm in steps.

3.1 Particle swarm optimization overview

The PSO is a population-based stochastic technique inspired by bird flocking, fish schooling and swarming theory in particular. It employs a swarm of particles that move through the search space while "adjusting" their position according to their fitness value and that of the other particles. This "adjustment" of position is rendered by recording the best personal position of the particle and that of the swarm called the best global position. These positions, personal and global best, correspond to the location of optima found by each particle and that by the swarm. By using Eq. (1) at every time step, these positions of optima and current positions are utilized to calculate the velocity of the particles mathematically. This velocity, in turn, determines the position of the particle in the next time step using Eq. (2).

$$\overrightarrow{v_i^{t+1}} = m\vec{v_i^t} + cr_1^t\left[\overrightarrow{P^t_{\text{best},i}} - \vec{x_i^t}\right] + sr_2^t\left[\overrightarrow{G^t_{\text{best},i}} - \vec{x_i^t}\right] \quad (1)$$

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_i^t} + \overrightarrow{v_i^{t+1}} \tag{2}$$

where v_i^t represents the velocity of drone i in at time t, x_i^t represents the position of drone i in at time t, P_{best}^t is the best personal position of drone i in from the initialization through time t. G_{best}^t is the best global position of particle i in found from initialization through time t; c and s are the constants that are used to level the contribution of cognitive and social component. r_1^t and r_2^t are the random numbers generated between 0 and 1 generated at time t. m is the inertial weight.

Using the above equations, particles will keep updating positions until a global optimum solution is reached. However, being a metaheuristic algorithm, PSO does not guarantee that an optimal solution will ever be reached. Though the above algorithm is attributed to particles, it can also be used for drones operating in a large search space as they can simply be abstracted as particles in the swarm. A modified version of PSO having with adaptive PSO constants, developed by Shi et al. [23], is used in this work.

3.2 Collision avoidance algorithm for UAV particle swarm optimization application

The idea of the unmanned aerial vehicles coherently operating in unique planes is exploited in this conceptualization. If each UAV operates in a plane whose vertical distance from other horizontal planes (in which other drones are operating) is sufficiently large, it will not collide with other UAVs. Such ideas have been presented in the literature [24]. However, their motive was not of proposing a collision avoidance model. This research attempts to make a collision avoidance algorithm based on a multiple-plane approach which offers the benefits of no collision while changing the altitude of the drones and no significant change in the original PSO algorithm by changing the trajectories of the drones for collision avoidance. The following assumptions were made while proposing the collision avoidance algorithm:

3.2.1 Assumptions for the surroundings

- Search is being performed in a relatively open area with a low density of obstacles per unit area like a mine quarry, marine environment, etc. Even if the area has a significant amount of obstacles, their height should be limited so that UAVs can hover over the obstacles.
- The environment is static, i.e. the conditions of the environment are not changing and the obstacle is stationary.
- Dimensions of the search space is known even though anything about the search space is not known, the number of obstacles, location of victims, etc. However, a rough qualitative idea of the density of obstacles can be known from the area of application like a forest is supposed to have more obstacles than a desert.

3.2.2 Unmanned aerial vehicle-related assumptions

- UAVs have short-range communication capabilities which make communication among the UAVs possible. Some UAVs among the swarm have long-range communication capability or at least have a beacon nearby to relay information about the ongoing SAR to a central station, where first responders are.
- UAVs are of moving rotor type. Fixed-wing UAVs are unsuitable for this algorithm as they lack hovering capability.
- UAVs are equipped with an obstacle detection system to detect obstacles at short ranges. UAVs can use LiDAR, SONAR, computer vision techniques, etc. to gauge the dimensions of the obstacles.

3.3 Implementation of multi-plane system

If the UAVs travel in different altitudes, it is almost certain that they cannot collide, but if there is some obstacle in the plane of operation of the drones, it would have to change its plane of operation and shift to a new plane. As soon as the obstacle is detected the PSO algorithm is stopped and the swarm switches to the anti-collision algorithm.

UAVs initially hover at a particular location; after encountering an obstacle, drones move to the nearest free plane and use it to cross the obstacle from above. All other UAVs, which are not facing any obstacle, remain stationary to avoid collision while some of the drones switch planes. After the drone crosses the obstacle and returns to its original plane, swarm switches to PSO to continue the search.

The number of the free planes is dependent on the number of the obstacles in the surroundings and the number of the UAVs operating in the swarm. A general formula is given in Eq. (3) for number of planes P in the system,

$$P = n + [n/a] \tag{3}$$

$$F = [n/a] \tag{4}$$

where n = number of UAVs in the swarm, a is a constant decided on the expected number of obstacles in the surroundings, [.] is the smallest integer function, F = number of free planes.

The difference in the altitude of the planes is such that the thrust exerted by the wind blown by the rotors is not affecting the motion of the other UAVs below them. The drones are distributed equally amidst the occupied in and equally distributed such that the number of occupied planes between any two free planes is same, as shown in

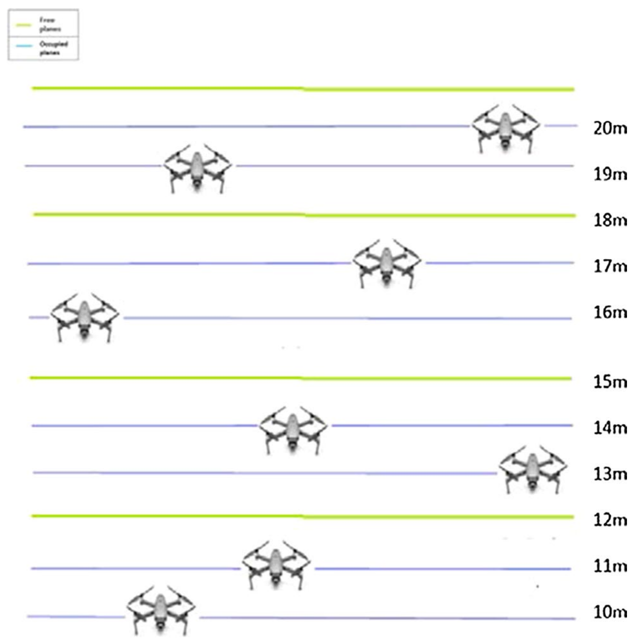


Fig. 2 An illustration of the distribution of free planes among occupied planes

Fig. 2. This is done to minimize the time of travel of each drone to the free plane and also increase the accessibility of the free planes. The number of occupied planes between each free plane is given by:

$$N_O = \left[\frac{P}{[n/a]}\right]$$

The range at which the UAV detects the obstacle highly depends on the obstacle detection system installed, but the distance at which it has to switch to the anti-collision algorithm depends on the size and manoeuvrability of the UAV, as the anti-collision algorithm will be using this information to be able to plan the trajectory.

3.4 Special problems and solutions

The conceptualization presented in the above sections fail in certain scenarios or at least becomes inefficient. Some additional features are accommodated in the algorithm for these special conditions to make the system more robust. These conditions and their solutions are mentioned in Table 4:

3.5 Mathematical expression of the proposed algorithm

The UAVs in the modified particle swarm optimization algorithm constantly switch planes in order to avoid collisions and obstacles, the behaviour of which is decided by the following equations. In order to eliminate zero error for varying heights, z values are adjusted using the following equations

$$\overrightarrow{v_i^{t+1}} = m\overrightarrow{v_i^t} + \mathrm{c}r_1^t\left[\overrightarrow{P^t_{\mathrm{best},i}} - \overrightarrow{x_i^t}\right] + sr_2^t\left[\overrightarrow{G^t_{\mathrm{best},i}} - \overrightarrow{x_i^t}\right] \tag{5}$$

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_i^t} + \overrightarrow{v_i^{t+1}} \tag{6}$$

$$F = [n/a] \tag{7}$$

$$P = n + [n/a] \tag{8}$$

$$Z_i^* = Z_i - \left(P_i * h\right) \tag{9}$$

$$S_i = \left\{ \begin{array}{l} 0 \text{ if}\left(O_z - Z_i\right)^2 + \left(O_z - Y_i\right)^2 + \left(O_z - X_i\right)^2 < d^2 \\ 1 \text{ if}\left(O_z - Z_i\right)^2 + \left(O_z - Y_i\right)^2 + \left(O_z - X_i\right)^2 > d^2 \end{array} \right\} \tag{10}$$

where v_i^t, represents the velocity of ith drone at tth iteration, x_i^t represents the position of ith drone at tth iteration, P^t_{best}, is the best personal position of drone i till tth iteration for the objective function, G^t_{best} is the best global position of the swarm till tth iteration for the objective function t; c and s are the constants that are used to level the contribution of cognitive and social component, r_1^t and r_2^t are the random numbers generated between 0 and 1 generated at time t, m is the inertial weight, Z_i represents actual vertical position of

Table 4 Additional novel features of proposed algorithm for accommodating exceptional cases

Condition/situation	Remedy
If the lowest plane encounter many obstacles or if the terrain crosses a plane	The whole system of the planes shift in altitude by the height of the difference in altitude of two consecutive planes
If an obstacle is very tall (crosses more than three planes)	The whole system of the planes shift in altitude by the thrice height of the difference in altitude of two consecutive planes
In the event in which if 2 or more drones encounter same obstacle	The drones above will shift their position in their plane and let the drones in the lower planes cross over the obstacle

the ith UAV particle at tth iteration, Z_i^* represents zero error adjusted vertical position of the ith UAV at tth iteration, n = number of UAVs in the swarm, a is a constant decided on the expected number of obstacles in the surroundings, [.] is the smallest integer function, F = number of free planes, P_i represents plane position of ith drone, d represents the constant threshold value of Euclidean distance between the particle and obstacle that counts as a collision, O_z represents z position of the obstacle, O_y represents y position of the obstacle, O_x represents x position of the obstacle, h represents the constant height between 2 consecutive planes, S_i represents the switch plane function for the ith drone; if the value is 0, then the plane is not switched; if the value is 1, then drone shifts the plane to the nearest free plane.

Arrays are created to store plane information in the following syntax to achieve this task

$$\text{Total\_Planes} = \left[\text{True/False, drones\_in\_plane, Plane\_Index} \right]$$

wherein the first variable can hold two types of values—True and False. True represents that the plane is not free and that there are drones in the plane. False represents that there are no drones in the plane and the plane is free for drones to switch to. Second variable stores the ID number of drones in plane. Plane Index represents the position of the plane. For example, if the 2nd and 27th drones are in 7th plane, then the 7th element of the array would look like [True, [3, 7], 7].

3.6 Significance of threshold value *d*

Threshold value d is the maximum Euclidean distance between a UAV particle and its obstacle that counts as a collision. This threshold value depends on the physical attributes of the UAV, search area and obstacle size. For instance, a UAV that requires a safe spherical radius of operation as twice its diagonal length x will have the d value as 3×.

3.7 Algorithm steps

Based on the earlier discussion and equation used in section three, the algorithm with steps is presented in this subsection:

R1 Decide the number of drones and a factor for the environment.
R2 Calculate the number of total and free planes according to the equations using Eqs. (3) and (4) in the and velocity of the drones according to Eqs. (1) and (2) for initialization
R3 Allocate planes to drones and decide the base height (altitude of the bottom plane above ground) of the multiplane system. Also store the altitude of each of the drone in an array according to the plane of the respective drone.
R4 Evaluate the fitness value for each of the UAV for population in R3.
R5 Update velocity and positions for all the Drones using Eq. (1) and (2) and start motion of UAVs
R6 Continuously check for obstacles via obstacle detection during motion for each drone while in motion and
R7 If obstacle is encountered, then go to R8; otherwise, go to R22.
R8 Stop the motion of all the drones.
R9 Set variable m = number of drones detecting obstacles in their path.
R10 If m is not 0, then go to R11; otherwise, go to R19.
R11 Check whether groups of UAVs are facing the same obstacle or not; if yes, then go to R12; otherwise, go to R15
R12 For each group of shift planes for drone with higher altitude until its path is clear. Cross the obstacle and return to the original plane and put q=0 and put variable p=0
R13 Repeat R12 for next highest altitude UAV until the stack of UAVs is clear
R14 Assign number of drones that faced same obstacle and got their path cleared to p and set counter, $c = p$.
R15 Set $m = m - q - p$ and go to R10.
R16 For each one of the UAVs not facing the same obstacle, send them to a free plane with higher altitude until the path is clear. Cross the obstacle and return to the original plane and put variable p and q = 0
R17 Set q = number of UAVs facing different obstacle whose path is cleared
R18 Set $m = m - q - p$ and go to R10.
R19 If $c > 3$, then go to R20; otherwise, go to R21.
R20 Go to R3.
R21 Resume the motion of all Drones and go to R6.
R22 Evaluate the fitness function for each UAV and calculate personal and global best
R23 Is the convergence criteria reached? If yes, then go to R24; otherwise, go to R5.
R24 End.

3.8 Flowchart

Figure 3 represents a flowchart of the novel algorithm proposed and discussed in Sect. 3.5.

4 Simulation

This section focuses on the description of the simulation, developed using MATLAB to visualize a set of drones using the PSO with the proposed collision avoidance algorithm. The first of the three subdivisions discusses the kind of objective functions used and the rationale behind using those

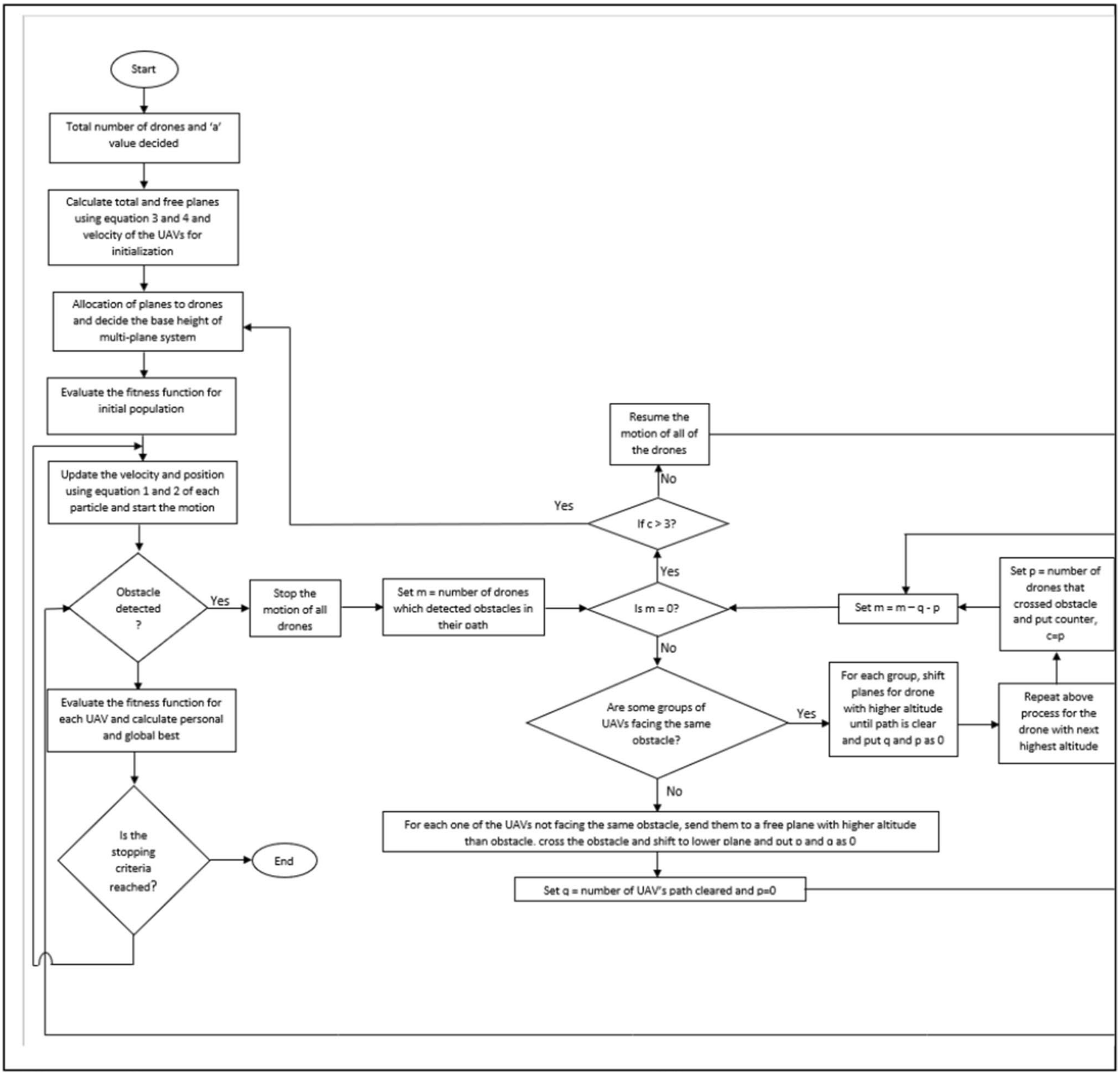


Fig. 3 Flowchart of the novel algorithm

functions, the second part mentions the specifications of the search domain and its bounds, and the third part discusses various considerations taken into account regarding UAVs and the anti-collision algorithm. The fourth part discusses how collisions were calculated in the simulations.

4.1 Objective functions for benchmarking

Finding a general objective function that can mimic the actual search and rescue mission in a simulation was a difficult task as each kind of search and rescue may use different types of inputs as fitness values. In the real world, various kinds of wireless technologies, which may aid in finding the victims, are available at disposal [25]. A few of these technologies are IEEE 802.11 standards [26]. Each of the drones can use, for example, strength of the distress signal being emitted by the survivors or use image processing for gauging the likeliness of finding the victims [27]. In either of the aforementioned ways (though not limited to the two), each drone is getting an input fitness value which can be used by the swarm to develop the strategy. Thus, a real-life search and rescue problem can also be called as an optimization problem in which the global optimum is the position of the victim.

Based on this argument, it was decided to use mathematical benchmarking functions, commonly used by the researchers as the objective functions for the simulations in addition to the typical constrained optimization mathematical functions.

4.2 Search area and bound

For this research, it was decided to use Rastrigin and Schwefel functions as the main objective functions as they are commonly used by researchers for benchmarking [28]. Moreover, randomized constrained optimization functions are used with different parameters to add for an exhaustive dataset and to imitate contour as shown in Fig. 6. The constants are randomized to generate random contours.

For the simulation, the contour of the function can be considered as the search space and the bounds as the limits of the search area where the drones move from point to point in search of the optimum position. For the sake of visualization, the three-dimensional surfaces of the functions can also be considered as the terrain over which UAVs fly, checking the fitness value at every location as they travel. The drones calculate the value of the objective function at each coordinate to get the fitness value. The distribution of the drones in the various planes in the beginning and their convergence over the minima at $x=0$, in the end, is illustrated in Figs. 4 and 5.

In Fig. 6: $Minf(x,y) = (1-x)^2 + 100\left(y - x^2\right)^2$

subjected to: $(x-1)^2 - y + 1 \leq 0$

$x + y - 1 \leq 0$

$-1.5 < x < 1.5$

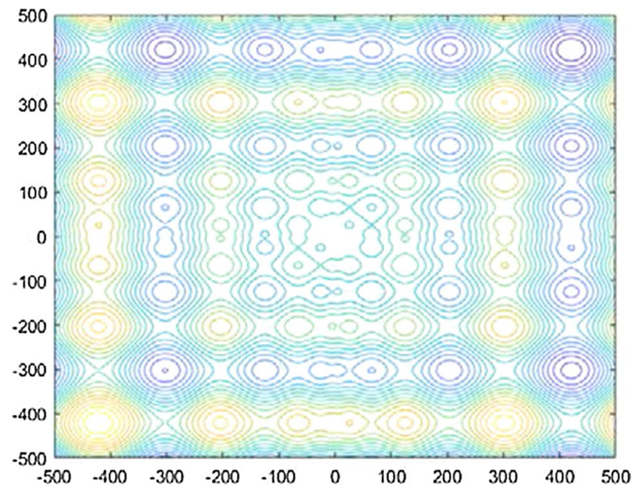


Fig. 5 Contour of Schwefel function; gradient of yellow to purple denotes variation in depth wherein purple region is the deepest part and yellow region is the highest part (colour figure online)

The constants are varied to get random constrained objective functions.

4.3 UAVs in the simulation

For the simulation, the following considerations were taken into for the surroundings and the UAV.

- If the final position of the UAV, calculated by the algorithm for the next iteration, is out of the bounded space, then the UAV's final position will be changed to the point

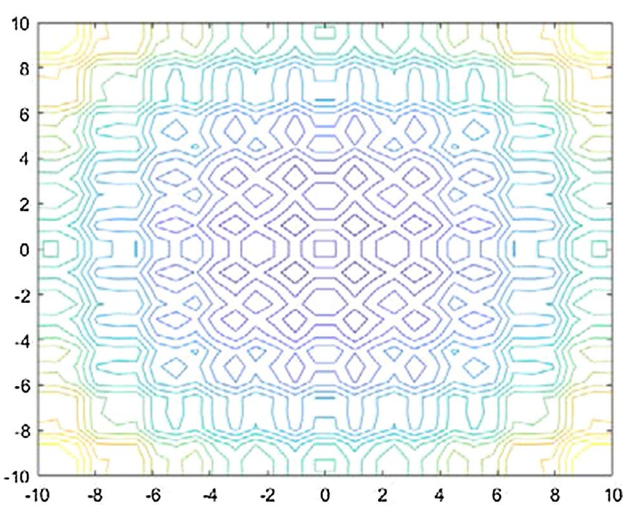


Fig. 4 Contour of Rastrigin function; gradient of yellow to purple denotes variation in depth wherein purple region is the deepest part and yellow region is the highest part (colour figure online)

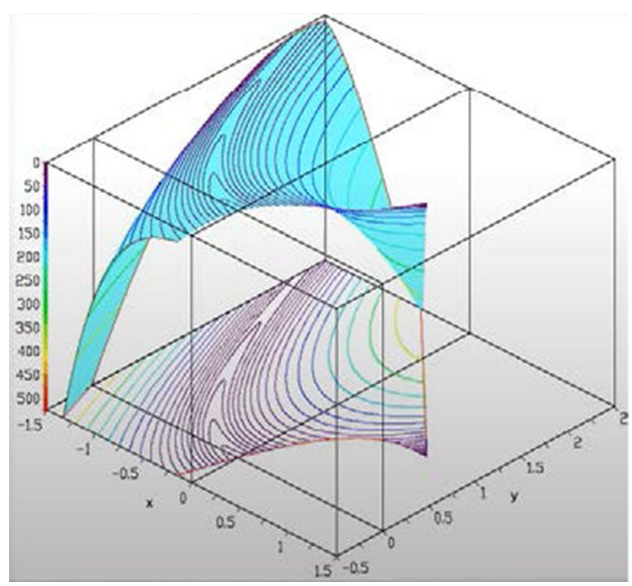


Fig. 6 Contour of randomized constrained optimization benchmarking function

in the domain, that is, on the line joining initial and the earlier final position (outside the bounded space) and the border of the space simultaneously.

- UAVs enter the search space from a launch site to the search space. UAVs then assume their positions, evenly distributed throughout the search space, before starting the actual search. This strategy helps in mitigating the problem of UAVs getting stuck in local optima.
- The "*a*" factor for the environment was taken as infinity to make the number of free planes equal to zero. This was done as there are no obstacles, other than the terrain, for the drones and they automatically shift the multi-plane system to an altitude that does not intersect with the terrain.

The UAVs use the combined PSO and collision avoidance algorithm in coherence as mentioned in the flowchart and algorithm steps of the methodology. The results obtained from the simulation and other inferences are presented in the later sections (Fig. 7).

4.4 Number of collisions

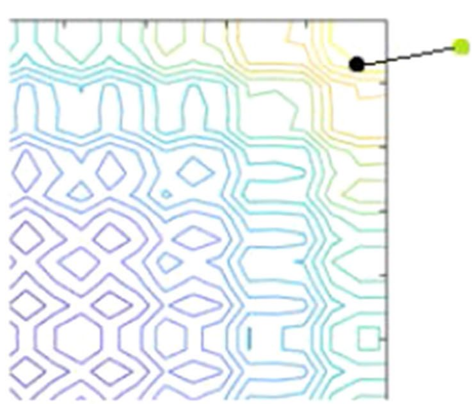
To calculate the number of collisions, cases were developed by varying swarm size, iterations, objective functions, height between the planes and the threshold value of Euclidean distance for the count of collisions. A total of 100 cases are made by randomly choosing the objective function among the ones listed in Sects. 4.1 and 4.2, randomly choosing swarm size between 10 and 1000, randomly choosing total iterations between 35 and 200, randomly varying height between the planes. Collisions were calculated had the UAVs been operating on original PSO and for the cases when UAVs were operating on the proposed algorithm. One sample case is shown in Fig. 8. The objective function for this case is the Rastrigin function, swarm size is 100, iterations are 100, height between planes is 0.1, and threshold Euclidean distance is 0.0001. The total original PSO collision is 17719, whereas total collisions with the proposed algorithm are 0. Collisions are calculated for each iteration using the threshold Euclidean distance; that is, if the Euclidean distance between a particle and an obstacle is less than the threshold value, then the particle is interpreted to have physically collided. The threshold distance can be varied according to the dimensions of. Similarly, 100 different cases with 200 data points are simulated, on the dataset of which the statistical z test is run to check the hypothesis that the proposed collision avoidance algorithm significantly reduced the number of collisions.

5 Results and discussions

In this section, the results of the simulated environment, described in the previous section, are presented. For the evaluation of the performance of the proposed modified UAV-PSO, several comparisons on different critical indicators, such as the probability of collisions and convergence plots, were made. All the comparison results clearly illustrate the effectiveness and advantages of the proposed improvements.

5.1 Probability of collisions

UAVs using PSO, without the proposed collision and obstacle avoidance system, are prone to a high probability



(a) **(b)**

Fig. 7 illustrates the case adjustment for the cases where the final position of a drone calculated by PSO is outside the boundary for the simulation. **a** represents one such case where black dot is the actual position of the UAV before the iteration and green dot is the calculated position after the iteration. Since the green dot is outside the boundary, the simulation changes the final position as shown in **b** where the red dot represents the intersection of the line joining initial (black dot) and the earlier final position (green dot) and the border (colour figure online)

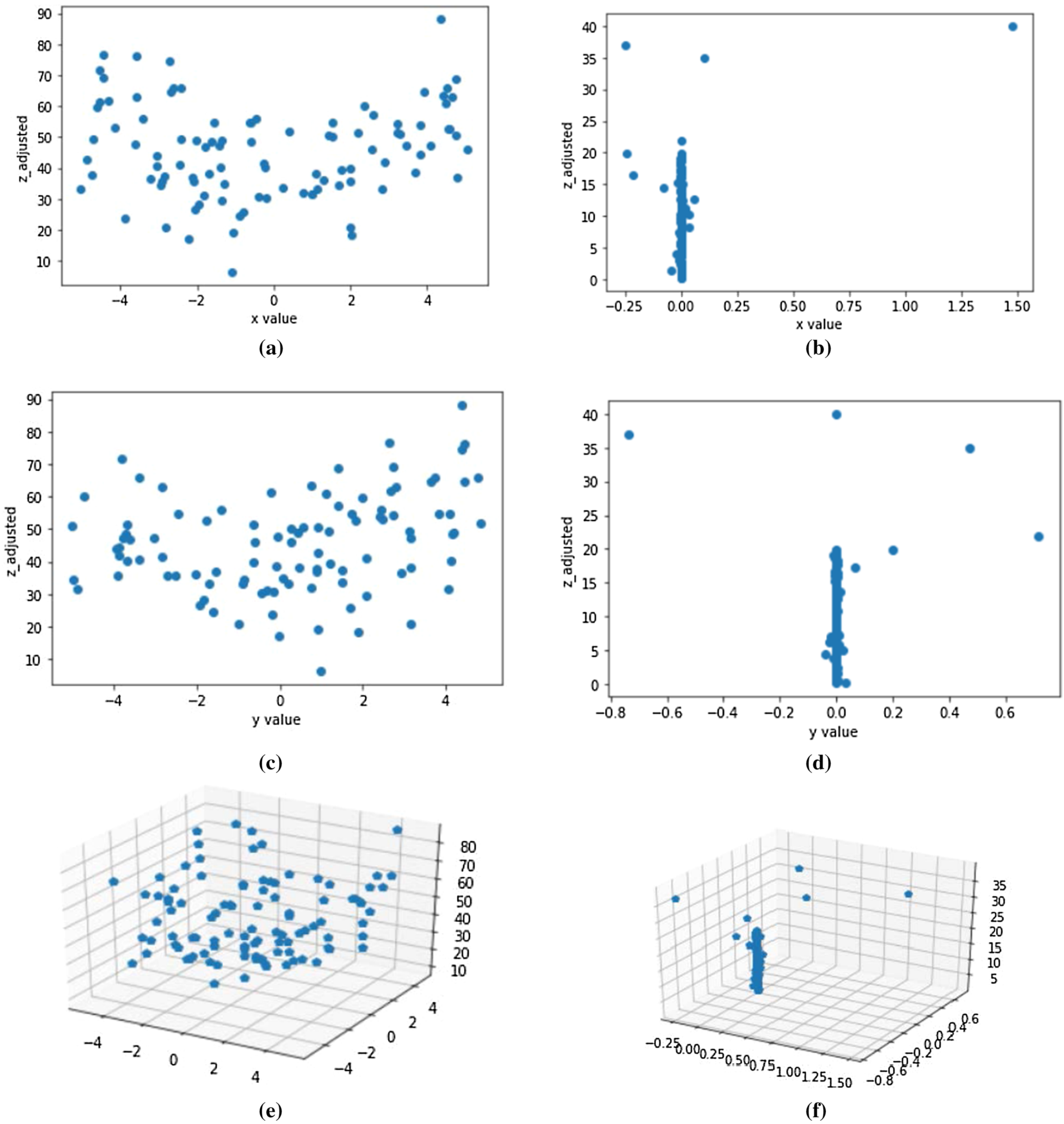


Fig. 8 **a**, **b**, **c** represents the initial state of the UAV particles before the proposed algorithm is run, whereas figure **d**, **e**, **f** represent the final state of UAV particle after all iterations. **a** and **b** represent the front view, **c** and **d** represent the side view, **d** and **f** represent 3d views

of collisions in physical search spaces which renders the use of unmanned aerial vehicles redundant. The proposed modifications reduce the probability of collisions between any two unmanned aerial vehicles to almost zero as the different UAVs operate in different non-intersecting planes shown in Fig. 8. However, challenging environments, such as spaces not having enough vertical room for the collision avoidance planes to exist and still the number of planes is not reduced, can increase the probability of collisions between the unmanned aerial vehicles as well. Moreover, the probability of the collision, while being low, also depends on how accurately the UAV's on-board obstacle detection system works.

The proposed obstacle avoidance system works for the case of static obstacles. For dynamic obstacles, the probability of avoiding collisions with obstacles decreases because the unmanned aerial vehicles are prone to collisions with obstacles in motion, after detection or while changing the planes. If a UAV detects dynamic obstacles, this algorithm can be used in conjunction with the dynamic path planning algorithms. The UAV can switch to a dynamic path planning algorithm for dodging a moving obstacle and use the obstacle avoidance, conceptualized in this work, for stationary obstacles. However, these dynamic path planning algorithms were not included in the current version.

5.2 Statistical test for collisions

To perform the statistical comparison, 100 samples were created by changing objective function, swarm size, iterations, the threshold of Euclidean distance between the obstacle and particle to count as collision, and the height between the planes.

The statistical test used for comparison is the z-test. This Z-test is valid as the sample satisfies necessary conditions for the Z-test, i.e. data points are independent of each other, sample size greater than 30 and dataset is randomly selected from the population. Taking the null hypothesis that difference between the mean collisions per total iterations of PSO without collision avoidance algorithm and with collision avoidance algorithm is zero, i.e.

Null hypothesis $H_0 : \mu_1 - \mu_2 = 0$
Alternate hypothesis $H_1 : \mu_1 - \mu_2 > 0$

where μ_1 is mean total collisions per total iterations of samples with pure PSO, μ_2 is mean total collisions per total iterations of samples run on PSO with proposed collision avoidance.

In applying the statistic z-test, significance level = 0.05 is used. A modular collision avoidance algorithm is proposed to amend with PSO, the philosophy of which lies in the concept that each particle is given its own individual space while the PSO algorithm is run. This is achieved by a multi-plane approach wherein particles switch planes in difficult situations. The statistical Z-test was run on the mean of the total number of collisions in the original PSO and modified version of POS. p-value for the z test comes out to be $8.640335 \times 10^{-75.}$ With the results of the statistical z-test, it is concluded that there is significant evidence that the proposed approach significantly reduces the collisions of UAVs. The results of the statistical z test indicate giving each particle its own individual space to perform the search without altering the parent algorithm leads to reduced collisions by a large margin (Fig. 9).

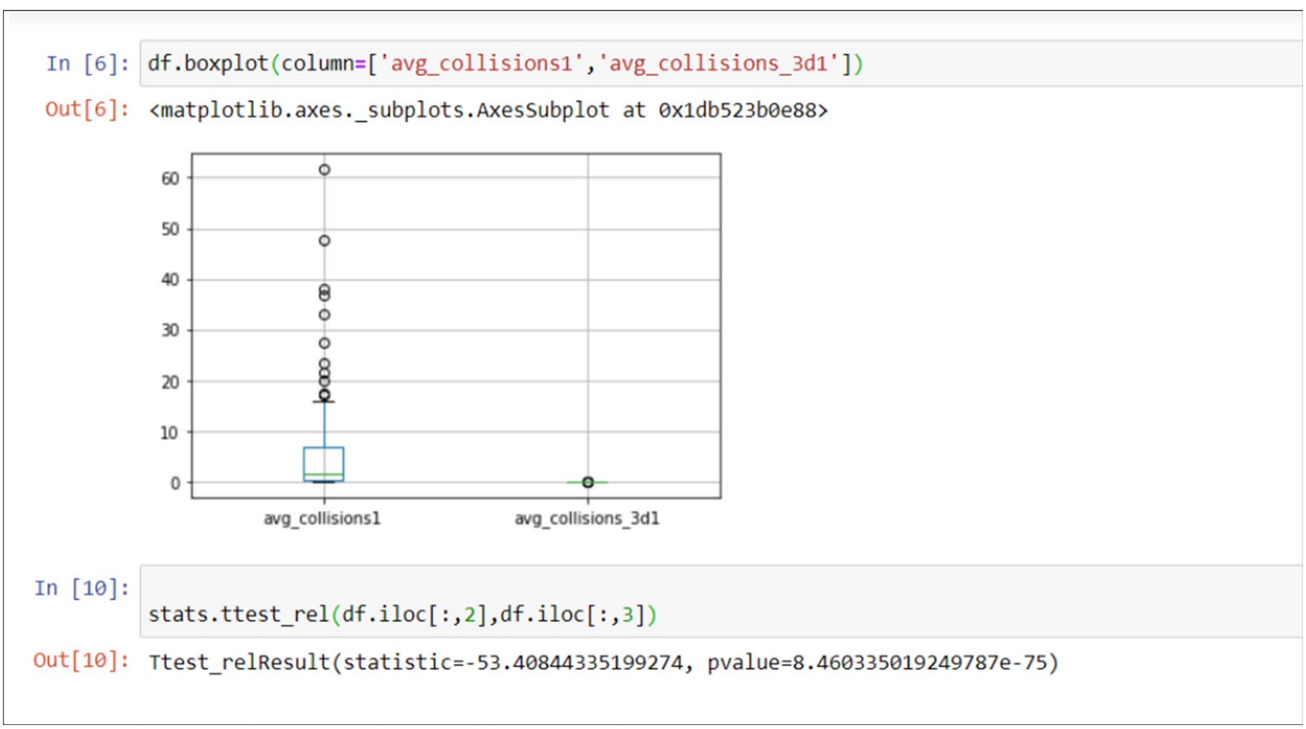


Fig. 9 Results of the statistical z test

5.3 Comparison of convergence

As shown in plot for average convergence in Fig. 10, the collision avoidance does not affect the convergence, there is no significant difference in the iterations to convergence in this conceptualization and modified PSO developed by Shi and Eberhart. The slight difference in the values calculated by the PSO with and without collision avoidance can be attributed to the random numbers, r_1^t and r_2^t, multiplied in Eq. (1) of Sect. 3. The value of the fitness value reaches within a tolerance of 0.001 to the global optima in around 40 and 220 iterations in approximately for both the objective functions. The exact value of the number of iterations to convergence changes for different runs of the algorithm.

Standardized residual plots give an idea of error in the global optima found by the algorithm more accurately as the values become asymptotic in the convergence plot as shown in Fig. 10.

The residuals show that there is no significant difference between the error in the values calculated by PSO and PSO with the anti-collision algorithm. The formula used for the calculation of standardized residuals, for each of the iterations, is mentioned below:

$$\mathrm{SR} = \frac{g_i - e}{\mathrm{S.D.E.}}$$

where SR = standardized residual, SDE = standardized error, g_i = Global minimum at the ith iteration and e = expected absolute minima.

5.4 Computational power and cost

For the case of zero obstacles in the search space, the computational power requirement for the proposed variant is almost equal to the computational power required of original particle swarm optimization. However, there is a requirement for additional memory to store the altitude of operation of each UAV. Practically, depending on the objective function, the UAVs might have to take images to process the fitness of a location, and cloud computing might be necessary for image processing. In such a case, the UAV at the higher altitude might need a camera with higher resolution to capture the location underneath accurately and better connectivity to send the data of larger size to the cloud server, increasing the overall cost.

For a case with obstacles, time for "crossing" the obstacle gets added to the total search time. This time depends on the density of static obstacles. As the number of obstacles increases, the time taken by the UAVs to complete the mission also increases.

5.5 Time complexity

Using the big-O notation, the time complexity of the original PSO is calculated as $O(m * N)$ where m is the population size and N is the number of dimensions. For the use case of search and rescue, N is generally 2 or 3, but at the cost of either additional time complexity of the native collision avoidance algorithm or at the cost of damage due to collisions. Original PSO cannot be implemented without a collision avoidance algorithm, which can significantly increase the time complexity and affect the searching algorithm path.

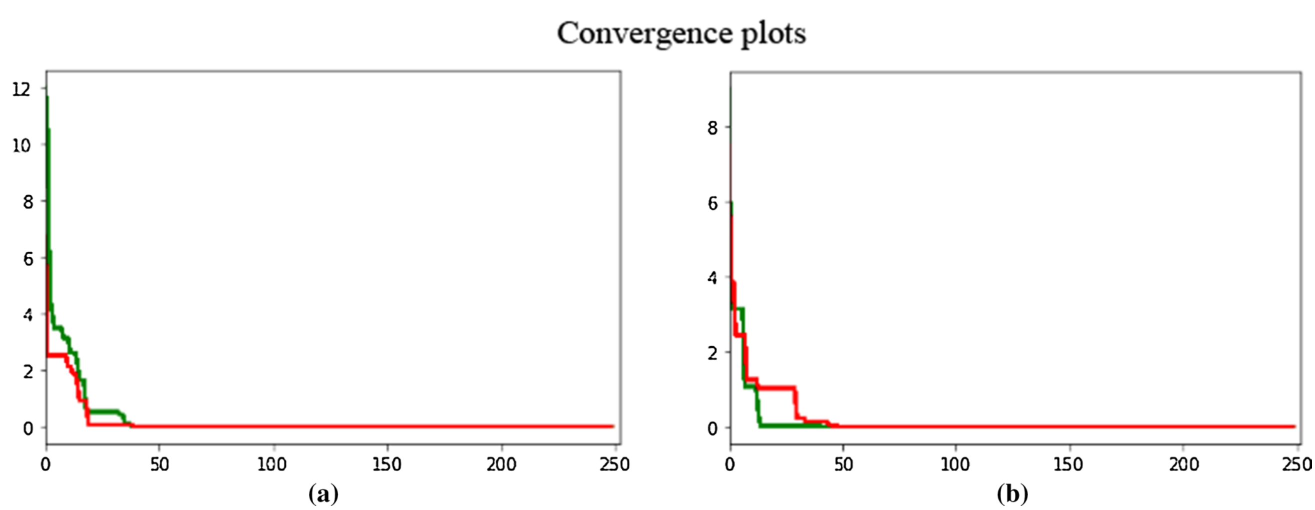


Fig. 10 **a** represents average fitness value plotted against number of iterations, **b** represents standardized residuals plotted against number of iterations, keeping swarm size and iterations constant, where green line depicts original PSO and red line depicts modified variant (colour figure online)

In the proposed algorithm, using the big O notation, time complexity comes out to be $O(m * P)$ where m is the population size and P is the number of planes according to the formula in Eq. (8), without compromising the parent PSO algorithm.

6 Conclusion

A novel collision avoidance algorithm is proposed for the application of a swarm of UAVs operating on particle swarm optimization for search and rescue operations. The collision avoidance algorithm is simulated for a total of 100 cases with 200 data points which were generated by varying swarm size, iterations, height between the planes, threshold Euclidean distance for collisions and objective functions. As the *p*-value of the statistical z test is 8.46×10^{-75}, results of the statistical z test run on the simulation dataset to favour the alternate hypothesis that the proposed algorithm significantly reduces the total number of collisions. The reason behind this result is that each UAV is allowed to move in its own independent space using a multi-plane approach without altering the parent algorithm. Moreover, the results of the simulations indicate that the new algorithm does not affect the efficacy of the original algorithm as the average of iterations to convergence and the trend of standardized residuals is the same for both original PSO and proposed algorithm.

6.1 Application areas

Potential applications of the proposed algorithm include various classes of search and rescue missions, wherein obstacles are mostly static. For example, maritime search and rescue cases are appropriate for the proposed algorithm as the environment for drones to operate in has minimum dynamic obstacles. Moreover, the use case of the proposed algorithm can be extrapolated to mining fields, exploring hazardous areas and other searching applications wherein a swarm of unmanned aerial vehicles can be used to search for the desired targets. The multi-plane approach of collision and obstacle avoidance can also be extrapolated to the newer metaheuristic algorithms featuring unique advantages.

6.2 Limitations and future works

The application of the proposed modified PSO model is limited to a simple search space with sufficient aerial space for the UAVs to switch planes. For a congested space, the algorithm is likely to fail or at least become highly inefficient. The collision avoidance which was integrated in the algorithm is a simple one and still requires provisions that would also prevent collisions with complex dynamic obstacles, such as birds. Till the time this new feature is incorporated, collision avoidance in this conceptualization can be used in combination with dynamic collision avoidance algorithm developed by other researchers. This way a bargain can be obtained between the advantages of both the algorithms. Also, near the final iterations of PSO, huge clusters form around the evident global minima. To avoid cluster formation, convergence algorithms can be proposed. The proposed algorithm in integration with the original PSO is prone to the original PSO's limitations of potentially falling in the local search trap; however, this can be solved in future works by integrating the proposed collision avoidance with hybrid versions of PSO and other metaheuristic algorithms as in the latest research works [29].

The model can also be extrapolated to search missions of various settings since the original searching algorithm remains theoretically unaffected by the inclusion of the collision avoidance algorithm. For example, in the field of mining, this implementation can be employed to search for the optimum mining zones in quarries. The use of UAVs would significantly reduce the search time. The proposed model can be extended to search on ocean beds as well where submarines can be treated as PSO particles working with the collision avoidance algorithm proposed in this paper.

In this work, the collision and obstacle avoidance algorithm is introduced for PSO. As a future line of work, the collision avoidance algorithm proposed in this document can be integrated with other algorithms. Moreover, the collision and avoidance algorithm will be extended to address the obstacles of high complexity. Further development to address dynamic obstacles can significantly widen the application domain of this algorithm but maintaining simplicity will still a challenge.

Author contributions All authors contributed to the study conception. Material preparation, coding and analysis were performed by AA, AD and AP. The first draft of the manuscript was written by GK, US and WKS. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Code availability MATLAB and Jupyter Notebook software are used in this work.

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

References

1. CRED, UNDRR, https://cred.be/sites/default/files/CredCrunch61-Humancost.pdf
2. Alotaibi E, Alqefari S, Koubaa A (2019) LSAR: multi-UAV collaboration for search and rescue missions. IEEE Access 7:55817–55832
3. Arnold R, Yamaguchi H, Tanaka T (2018) Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. J Int Humanit Action 3:1–8
4. Lee K, Park J (2006) Application of particle swarm optimization to economic dispatch problem: advantages and disadvantages. In: 2006 IEEE PES power systems conference and exposition
5. Ghazali S, Anuar H, Zakaria S, Yusoff Z (2016) Determining position of target subjects in maritime search and rescue (MSAR) operations using rotary wing unmanned aerial vehicles (UAVs). In: 2016 international conference on information and communication technology (ICICTM)
6. Lee S, Morrison J (2015) Decision support scheduling for maritime search and rescue planning with a system of UAVs and fuel service stations. In: 2015 International conference on unmanned aircraft systems (ICUAS)
7. Rahmes M, Chester D, Hunt J, Chiasson B (2018) Optimizing cooperative cognitive search and rescue UAVs. In: Proceedings SPIE10643, autonomous systems: sensors, vehicles, security, and the internet of everything
8. Dinnbier N, Thueux Y, Savvaris A, Tsourdos A (2017) Target detection using Gaussian mixture models and fourier transforms for UAV maritime search and rescue. In: 2017 International conference on unmanned aircraft systems (ICUAS)
9. Kohlbrecher S, Meyer J, Graber T, Petersen K, Klingauf U, von Stryk O (2014) Hector open source modules for autonomous mapping and navigation with rescue robots. In: RoboCup 2013: robot world cup XVII. pp 624–631
10. Cui J, Phang S, Ang K, Wang F, Dong X, Ke Y, Lai S, Li K, Li X, Lin F, Lin J, Liu P, Pang T, Wang B, Wang K, Yang Z, Chen B (2015) Drones for cooperative search and rescue in post-disaster situation. In: 2015 IEEE 7th international conference on cybernetics and intelligent systems (CIS) and IEEE conference on robotics, automation and mechatronics (RAM).
11. Grissom C, Thomas F, James B (2006) Medical helicopters in wilderness search and rescue operations. Air Med J 25:18–25
12. Sun J, Li B, Jiang Y, Wen C (2016) A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes. Sensors 16:1778
13. Al-Kaff A, Gómez-Silva M, Moreno F, de la Escalera A, Armingol J (2019) An appearance-based tracking algorithm for aerial search and rescue purposes. Sensors 19:652
14. XXXX (2021) Coverage algorithms for search and rescue with UAV drones. In: XIII AI*IA symposium on artificial intelligence, AIRO workshop
15. Wei C, Hindriks K, Jonker C (2016) Dynamic task allocation for multi-robot search and retrieval tasks. Appl Intell 45:383–401
16. Abdollahzadeh B, Gharehchopogh F, Mirjalili S (2021) African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng 158:107408. https://doi.org/10.1016/j.cie.2021.107408
17. Abdollahzadeh B, Soleimanian Gharehchopogh F, Mirjalili S (2021) Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Int J Intell Syst 36(10):5887–5958. https://doi.org/10.1002/int.22535
18. Alhaqbani A, Kurdi H, Youcef-Toumi K (2020) Fish-inspired task allocation algorithm for multiple unmanned aerial vehicles in search and rescue missions. Remote Sens 13:27
19. Sánchez-García J, Reina D, Toral S (2019) A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario. Future Gener Comput Syst 90:129–148
20. Duan H, Luo Q, Shi Y, Ma G (2013) Hybrid particle swarm optimization and genetic algorithm for multi-UAV formation reconfiguration. IEEE Comput Intell Mag 8:16–27
21. Ghamry K, Kamel M, Zhang Y (2017) Multiple UAVs in forest fire fighting mission using particle swarm optimization. In: 2017 international conference on unmanned aircraft systems (ICUAS)
22. Tsung-Ying S, Chih-Li H, Shang-Jeng T, Chan-Cheng L (2008) Optimal UAV flight path planning using skeletonization and particle swarm optimizer. In: 2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)
23. Shi W, Li J, Xu W, Zhou H, Zhang N, Zhang S, Shen X (2018) Multiple drone-cell deployment analyses and optimization in drone assisted radio access networks. IEEE Access 6:12518–12529
24. Hauert S, Winkler L, Zufferey J, Floreano D (2008) Ant-based swarming with positionless micro air vehicles for communication relay. Swarm Intell 2:167–188
25. Morgenthaler S, Braun T, Zhao Z, Staub T, Anwander M (2012) UAVNet: a mobile wireless mesh network using unmanned aerial vehicles. In: 2012 IEEE globecom workshops, 2012, pp 1603–1608
26. Carpin M, Rosati S, Khan M, Rimoldi B (2015) UAVs using bayesian optimization to locate WiFi devices. https://arxiv.org/abs/1510.03592
27. Hereford J, Siebold M, Nichols S (2007) Using the particle swarm optimization algorithm for robotic search applications. In: 2007 IEEE swarm intelligence symposium
28. Chou P, SauBor H (2011) Modified particle swarm optimization for sphere, rastrigin, schwefel and schafer fuctions. In: Proceedings of 2011 cross strait quad-regional radio science and wireless technology conference
29. Stacey A, Jancic M, Grundy I (2003) Particle swarm optimization with mutation. In: The 2003 congress on evolutionary computation. CEC '03
30. Peng Z, Li B, Chen X, Wu J (2012) Online route planning for UAV based on model predictive control and particle swarm optimization algorithm. In: Proceedings of the 10th world congress on intelligent control and automation
31. Geng Q, Zhao Z (2013) A kind of route planning method for UAV based on improved PSO algorithm. In: 2013 25th Chinese control and decision conference (CCDC)
32. Loscri V, Natalizio E, Mitton N (2014) Performance evaluation of novel distributed coverage techniques for swarms of flying robots. In: 2014 IEEE wireless communications and networking conference (WCNC)
33. Hayat S, Yanmaz E, Brown T, Bettstetter C (2017) Multi-objective UAV path planning for search and rescue. In: 2017 IEEE international conference on robotics and automation (ICRA)