Research Article

Hybrid PSO-HSA and PSO-GA algorithm for 3D path planning in autonomous UAVs

B. Abhishek[1] · S. Ranjit[1] · T. Shankar[1] · Geoffrey Eappen[1] · P. Sivasankar[2] · A. Rajesh[1]

Received: 18 April 2020 / Accepted: 6 September 2020 / Published online: 10 October 2020

Abstract

Unmanned aerial vehicles (UAVs) are a quintessential example of automation in the field of avionics. UAVs provide a platform for performing a wide variety of tasks, but in each case the concept of path planning plays an integral role. It helps to generate a pathway free of obstacles, having minimum length leading to lesser fuel consumption, lesser traversal time and helps in steering the aircraft and its corresponding antenna power signature safely around the hostile antenna to avoid detection. To optimize path planning to incorporate all the above-mentioned constraints, this paper presents two new hybrid algorithms particle swarm optimization (PSO) with harmony search algorithm and PSO with genetic algorithm. The hybrid algorithms perform both an exploratory and exploitative search, unlike the existing algorithms which are biased, towards either an exploitative search or an exploratory search. Furthermore, the hybrid algorithms are compared to the existing optimization algorithms and in all cases the hybrid algorithms give a minimum of 7% better result against PSO with up to a 40% better result against Invasive Weed optimization algorithm for a fixed computational time, suggesting better real-time applications.

Keywords Unmanned aerial vehicles · Particle swarm optimization · Harmony search algorithm · Genetic algorithm · Optimization · Path planning

1 Introduction

The use of the unmanned aerial vehicle (UAV) dates back to 1849, they were first used by Austrians solely for military purposes, that is, for bombing and reconnaissance. And the decades following the 1849 bombing of the Italian city of Venice by Austrians saw an increase of interest in the field of UAV. Several nations including the USA and Russia invested heavily in the technology leading to widespread technological advancement in the field of UAV. It was no more just a remote-controlled unreliable, expensive robot for the purpose of ferrying goods, with the introduction of semi-automation it became a state-of-the-art device, with its application ranging from military utility to farming. The UAVs became known for low cost, optimum size and additional manoeuverability owing to absence of manual pilot. To carry forward the technical evolution of an UAV, the concept of path planning was introduced, making UAV automated, helping them function efficiently, without being under human surveillance. Hence making manual controller redundant.

Path planning is a term used in robotics for the process of breaking down a desired movement task into discrete motions that satisfy movement constraints and possibly optimize some aspect of the movement. Path planning is classified either as local path planning or global path planning. In a local path planning, a robot navigates through the world map with obstacles in steps and calculates its next position dynamically, satisfying one or more predefined constraints. In global planning, the entire navigation path is planned by the drone prior to its movement towards the goal. This type of global planning is referred to

✉ T. Shankar, tshankar77@gmail.com | [1]VIT University, Vellore 632014, India. [2]NITTTR, Chennai, India.

SN Applied Sciences (2020) 2:1805 | https://doi.org/10.1007/s42452-020-03498-0

as offline planning in literature. The paper aims to optimize multi-constrained global path planning, by presenting two new hybrid meta-heuristic optimization algorithms. The algorithms presented are an amalgamation of two different existing optimization algorithms, one tuned for explorative search and the other tuned for exploitative search. An algorithm specifically dedicated to exploitative search may not always ensure an optimum path as complicated problems tend to terminate at a local minima/maximum and do not carry out a thorough analysis of the search space. Whereas using an explorative algorithm leads to inefficiency in exploitive search, that is, the exact global minima/maxima is not obtained. Hence, we choose to integrate two different algorithms, each tuned for a different purpose.

The main contribution of the hybrid algorithm in this paper is to provide path planning so as to generate a pathway free of obstacles, having minimum length leading to lesser fuel consumption, lesser traversal time and assist in steering the aircraft and its corresponding antenna power signature safely around the hostile antenna to avoid detection. The hybrid algorithms perform both an exploratory and exploitative search, unlike the existing algorithms which are biased, towards either an exploitative search or an exploratory search. In the hybrid algorithms suggested, explorative search has been carried out using PSO in both the cases. While exploitative search in first case is done using HSA and in the second case using GA. The hybrid meta-heuristic algorithms aim to reduce the distance traversed by the drone, the fuel consumed and the time taken. The pathway generated subsequently avoids all the obstacles and maintains a safe distance from prohibited hostile areas.

The hybrid algorithms constructed are favoured over the existing optimization algorithms because of the following reasons:

- The hybrid algorithm retains the robustness of PSO, enabling the particles (individuals) to move over the entire search space by updating position and velocity at the end of each generation.
- HSA and GA have the shortcoming of being restricted to a region but specialized in depth search, therefore aptly complement PSO.
- Moreover, the convergence to global minima/maxima is also fast.

In addition, to satisfy the manoeuverability constraints it is essential to obtain a smooth path rather than discrete points interconnected by line segments. For this purpose, we use B-spline curve smoothening method which specializes in producing an efficient path with few control variables. The rest of the paper is organized as follows: Sect. 2 describes the works related to path planning in UAV using optimization algorithms. In Sect. 3, the fitness function has been detailed. Section 4 explains the methodology using HSA algorithm and Sect. 4.1.1 elucidates the clustering using the PSO algorithm. The hybrid HSA-PSO algorithm is given in Sect. 4.1.2. The results and discussion of the proposed and existing algorithm are described in Sect. 5, and Sect. 6 concludes the major findings of the proposed algorithm.

2 Related work

A detailed literature review has been carried out, analysing several optimization algorithms to know their respective pros and cons, thereby helping us coin two new meta-heuristic algorithms which are both explorative and exploitative in nature. Most of the path planning algorithms can be categorized either as swarm-based or bio-inspired algorithms [1]. Algorithms falling into the former category include Genetic Algorithms (GA) [1], Cuckoo Search Algorithm (CSA) in [2], Ant Colony Optimization (ACO) in [3, 4], Particle Swarm Optimization (PSO) [1], Artificial Bee colony (ABC), Firefly algorithm (FA), Bees algorithm (BA), Honey Bees mating optimization algorithm (HBMO) in [5], Differential Evolution (DE) in [6], etc. While, Memetic Algorithm (MA), Neural Networks (NN), Whale Optimization Algorithm (WOA), Invasive Weed Optimization (IWO) in [7], etc., fall in the latter category.

The proposed algorithms have been compared against PSO, ABC, BA, FA, SFLA, IWO, Imperial Competitive Algorithm (ICA) and Simulated Annealing (SA). These algorithms have been proved to be very efficient and are being used in various optimization problems. Particle Swarm Optimization (PSO) in [8–10] mimics swarm behaviour in birds and fishes to guide the particles to search for global optimal solutions. Particles fly around in a multidimensional search space and each particle adjusts its position according to its own experience and according to the experience of its neighbouring particles. Each particle's fitness is evaluated and their personal best solution along with the global best solution in the swarm for that generation is updated. Convergence would be achieved by moving all particles towards the particle with the best solution until the stopping criteria is met. PSO is easy to implement, very efficient and faster but it is parameter sensitive.

In [11], the authors proposed Artificial Bee Colony (ABC) method for navigation. The ABC algorithm is successful in obtaining optimal solutions in the field of wireless communication [1]. It is inspired from the behaviour of honey bees in finding sources of nectar. This method has three types of bee–employed bee, the onlooker bee and the scout bee. Each type has a specific role. The onlooker bees search for food sources based on a fixed equation. The scout bees explore the search space randomly. All the probable food sources are evaluated on the basis of amount of nectar and worst solutions are abandoned. All the bees in the population memorize the positions corresponding to best solutions. The main advantages of this algorithm are: it is easy to implement and has only two parameters to adjust. The need for a high amount of objective function evaluations is a disadvantage.

Papers [12, 13] explore Bees Algorithm (BA). BA is very similar to ABC algorithm. The bee population is divided into scouts and foragers. Scouts explore the search space for best food sources. They bring back the information to the hive where they pass on the information through a "waggle" dance. The information includes direction, distance and quality of the food source. Higher the amount of food available, more number of foragers are sent to collect nectar efficiently. The remaining bees are sent to explore for other food sources. Firefly algorithm (FA) has been developed to mimic fireflies, which produce light to attract mating partners, to attract prey and as a warning mechanism. The brightness of light produced is inversely proportional to the distance between two fireflies. In this algorithm [14], all fireflies are considered to be unisex. This means that, a firefly will be attracted to another firefly regardless of its sex. The fireflies in the population or swarm move towards the firefly having brighter intensity light till the stopping criteria is met.

Whale optimization algorithm (WOA) in [15] was also studied. It is one of the most recent algorithms used in path planning. It was inspired by the hunting strategies of whales. The main steps involved are: the bubble-net method and spiral updating position. WOA can handle multi-objective functions and has a faster convergence rate. Authors in [16] make use of shuffled frog leaping (SFLA) optimization algorithm for navigation of a robot. Shuffled frog leaping algorithm is a memetic algorithm, which is based on frog leaping. The elements (frogs) in SFLA algorithm contain both local and global information, therefore allowing the solution to converge on global minima. The SFLA algorithm makes use of a mechanism similar to PSO for exploitative search while mechanism similar to complex evolution algorithm is used for explorative search. In this case, each element or frog or member of population carries memes which is a unit of cultural evolution. The algorithm specializes in solving multi-constrained optimization problem with fast convergence and has a higher likelihood of converging into a global optimal solution [17].

Another bio-inspired algorithm is the Invasive Weed Optimization (IWO) [18]. The term 'weed' refers to an unwanted plant whose presence is a threat to the cultivated plants. Invasive Weed optimization (IWO) is an algorithm developed to mimic the colonization of weeds. Weed are invasive and adaptive in nature. Initially, weeds are distributed randomly over the search space. The number of seeds produced by the weeds is directly proportional to the fitness function, i.e., better the fitness, higher is the number of seeds produced and vice versa. The number of weeds and seeds are maintained at a constant. Over iterations, weaker weeds are eliminated until the stopping criteria is met [19].

Imperialist competitive algorithm (ICA) is one of the recent meta-heuristic algorithms based on a socio-politically inspired optimization strategy [20]. The initial population is called empire and each individual agent is called a country. Countries in this algorithm are like chromosomes in GA and particles in PSO. The fitness of each country decides its power. The countries are categorized into imperialists and colonies. Imperialistic competitions take place among the countries. During these competitions, weak countries collapse and powerful ones take possession of their colonies. As the iterations go on, the solution converges to a state where only one empire exists and its colonies are in the same position and have the same cost as the imperialist countries. All these actions take place in three steps—assimilation, revolution and competition. Assimilation makes the colonies of each empire get closer to the imperialist state in the search space. Revolution brings about random changes in the position of some of the countries in the search space. Hybrid bio-inspired algorithm has been successfully implemented for the wireless communication and has an upper edge over the conventional bio-inspired algorithm [18, 20, 21]. In this paper, hybrid PSO-HAS and PSO-GA algorithms are proposed for the UAV path planning optimization.

3 Proposed fitness function

Fitness function indicates the proximity of the obtained solution to the given ideal condition. It can be treated as a figure of merit. In this paper, our aim is to find the global minimum for the objective function, which means lesser the value of objective function for a solution, more is the fitness of the solution. To capture the various constraints specific to our problem, a new objective function has been formulated. The various parameters constituting the multi-objective function are:

1. The distance traversed between the source and destination should be as small as possible. Therefore, the objective function should be directly proportional to the distance traversed as the aim is to obtain minima.

$$F \propto L \tag{1}$$

The length of the path is given in Eq. (2) as follows:

$$L = \sum \sqrt{\left(\mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2\right)} \tag{2}$$

where F = Fitness function, L = Distance traversed, dx = Infinitesimal length traversed by the drone along x axis, dy = Infinitesimal length traversed by the drone along y axis, dz = Infinitesimal length traversed by the drone along z axis.

2. The collision of the drone with obstacles is represented as violations (V) and is directly proportional to F. If the flight path intercepts any of the obstacle at any point in map, V will attain a value given by Eq. 6 which will drive the value of F up indicating an unfit solution, away from global minima.

$$F \propto V \tag{3}$$

Equation 4 is used to calculate the distance of each point on the pathway to a specific obstacle.

$$d_i = \sqrt{\left((x_i - x_{\text{obs}})^2 + (y_i - y_{\text{obs}})^2 + (z_i - z_{\text{obs}})^2\right)} \tag{4}$$

Subsequently the points chosen on the map representing the flight path attain a value in accordance with Eq. 5.

$$v_i = \left\{ \begin{array}{cc} 0, & \left(1 - \frac{d_i}{r_{\text{obs}}}\right) < 0 \\ 1 - \frac{d_i}{r_{\text{obs}}}, & \left(1 - \frac{d_i}{r_{\text{obs}}}\right) > 0 \end{array} \right\} \tag{5}$$

Average of v_i, with respect to a particular obstacle is obtained. Subsequently average of v_i with respect to other obstacles is obtained and summed to obtain V as depicted in Eq. 6.

$$V = \sum_{j=1}^{\text{nobs}} \sum_{i=1}^{100} \frac{v_i}{100} \tag{6}$$

where V = Violation, d_i = Distance of a point on drone's trajectory from an obstacles centre, in total the pathway has been segregated into 100 points, x_i, y_i, z_i = coordinates of a point on drone's trajectory, $x_{\text{obs}}, y_{\text{obs}}, z_{\text{obs}}$ = coordinates of an obstacles centre, r_{obs} = radius of an obstacle.

3. To take into account the proximity of the drone to prohibited or hostile area antennas, we make use of Friis equation. It calculates the average power being received (Pr) by the drone's antenna from the antenna in the hostile region. Pr should be as small as possible, hence is directly proportional to the objective function (F).

$$F \propto \text{Pr} \tag{7}$$

The radio propagation model used is Log-distance path loss model. Thus, the power received by the drone's antenna is given by Eq. 8.

$$P_r = \frac{P_t * G_t * G_r * \lambda^2}{(4 * \pi * d_0)^2} * \left(\frac{d_0}{d}\right)^L \tag{8}$$

where P_t = Power emitted by the transmitter, G_t = Gain of the transmitter, G_r = Gain of the receiver, λ = Wavelength of operation, d_0 = Reference distance, d = distance between transmitter and receiver, L = Path loss exponent.

Now combining Eqs. (1), (3), (7) and (8) the following objective function is obtained.

$$F = L(1 + \beta * V) * \frac{1}{T^\alpha} * \text{Pr} \tag{9}$$

where β and α = the parameters β and α are the tuning parameters and are basically uniform distributed random number between [0 1], initiated after the 1st iteration based on the performance of the algorithm.

4 Methodology

First, a 3D map filled with obstacles has been created. The shape of the obstacles chosen are spherical and cuboidal. The spherical spheres are used to mimic enemy's radars, where a drone has to stay out of, to remain undetected by hostile forces. The cuboidal shapes mimic birds or other drones which act as a hindrance to the drone's path. The obstacles have been randomly scattered over the map. The map has a dimension of 20 × 20 × 20 units, with a starting point of (−10, −10, −10) and an ending point of (10, 10, 10) for all simulations.

4.1 Proposed hybrid algorithms

4.1.1 Hybrid PSO/HSA algorithm for UAV path planning

Harmony search algorithm (HSA) draws inspiration from harmony improvisation by musicians, by trying out various combinations of pitches stored in their memory. Harmonic Memory Considering Rate (HMCR) and Pitch adjusting Rate (PAR) are two basic parameters in the HSA algorithm which control the component of solutions and even affect convergence speed. The algorithm begins by initializing the Harmonic Memory (HM) with random solutions within the prefixed boundaries.

For evaluation of the fitness function, a random solution is chosen by following any of the three rules:

- Choosing any value from the HM.
- Choosing an adjacent value from the HM.
- Choosing a random value from the possible value range.

The PSO–HSA hybrid algorithm is as follows.

```
Begin
Step1: Map Model. Generate the 3D model map with obstacles and determine the number of decision variables ($nvar$), contained by swarm's individuals (also called harmonics). The number of decision variables depicts the points to be plotted on the 3D map. Also select the lower bound ($v_{min}$) and upper bound ($v_{max}$) of variables.
Step2: PSO parameters.
        •  Maximum iteration for PSO (MaxIt_PSO)
        •  Swarm size (nPop)
        •  Inertia weight (w)
        •  Inertia weight damping ratio (wdamp)
        •  Personal Learning Coefficient ($c_1$)
        •  Global Learning Coefficient ($c_2$)
        •  Velocity variance parameter
    HSA parameters
        •  Maximum iteration for HSA (MaxIt_HSA)
        •  Number of new harmonies (nNew)
        •  Harmony memory consideration rate (HMCR)
        •  Pitch adjustment rate (PAR)
rand depicts a random number in the range 0 to 1 and randi depicts a random number drawn from standard normal distribution.

Step3: Initialization. Randomly generate a population (pop) with individual particles (also called harmonics) containing the fields:
        •  Velocity (V)
        •  Position(P)
        •  Evaluate Cost depending on position (Cost)
        •  Best Solution obtained by the individual (pBest)
        •  3D Model Plotting of the path
        Best solution (gBest) irrespective of the individual is obtained across the generation. The population (pop) is depicted by $nPop \times 1$ matrix. Both HSA and PSO refer to the same memory or population, therefore termed as particle harmony memory (also called population or swarm).
Step4: for i = 1 to MaxIt_PSO
      •  Organize the memory according to ascending order of cost.
      •  Determine best particle from memory
      •  Determine worst particle from memory
      •  Carry out local search using embedded HSA
       for j = 1 to MaxIt_HSA
        •  Create a new population (newpop) of harmonics and randomly assign position
         for k = 1 to nNew
          for m= 1 to nvar
            if rand<=HMCR
            Randomly select a harmonic from pop, let it be $n^{th}$ and assign its $m^{th}$ variable's value to the corresponding new harmonic variable.
             $pop(n).P(m) = newpop(k).P(m)$
            end if
            if rand<=PAR
             Mutate the new harmonic's  decision variable.
             $newpop(k).P(m) = newpop(k).P(m) + 0.02(v_{max} - v_{min}) * randi$
            end if
          end for
          Evaluate the cost.
         end for
        • Merge new harmonies with the existing harmonic memory.
        • Sort  the newly created harmonic memory in ascending order of cost of harmonics
        • Truncate the memory to original size by removing bad harmonies.
        • Update the best solution for each harmony
        • Update the global best solution
      end for
      • Update velocity and position of particles to carry out global search
      for j = 1 to nPop
        • Update velocity
          $pop(j).V = w * pop(j).V + c_1 * rand * (pop(j).pBest.P - pop(j).P) + c_2 * ...$
          $rand * (gBest.P - pop(j).P)$
        • Update position
          $pop(j).P = pop(j).P + pop(j).V$
        • Calculate the corresponding cost
        • Update personal best
        • Update global best for the generation
      end for
      Apply B-spline curve smoothening method to the global best solution and plot it on 3D map
    end for
Step5: Visualize. Plot the variation in global best   cost with each generation
```

4.1.2 Hybrid PSO/GA Algorithm for UAV path planning

To enhance the efficiency of our solution, we embed GA within PSO, to perform a more adept local search for the optimal solution while PSO is used for global search. The PSO–GA hybrid algorithm is as follows:

```
Begin
Step1: Map Model. Generate the 3D model map with obstacles and determine number of decision variables (nvar), contained by swarm's individuals (also called particles). The number of decision variables depicts the points to be plotted on the 3D map. Also select the lower bound ($v_{min}$) and upper bound ($v_{max}$) of variables.

Step2: PSO parameters
        •  Maximum iteration (MaxIt_PSO)
        •  Swarm size (nPop)
        •  Inertia weight (w)
        •  Inertia weight damping ratio (wdamp)
        •  Personal Learning Coefficient ($c_1$)
        •  Global Learning Coefficient ($c_2$)
        •  Velocity variance parameter
    GA parameters
        •  Maximum iteration (MaxIt_GA)
        •  Crossover percentage ($p_c$)
        •  Mutation percentage ($p_m$)
        •  Number of offspring ($N_c$)
        •  Number of Mutants ($N_m$)
        •  Range factor for crossover ($\gamma$)
        •  Mutation rate ($m_u$)
        •  Selection Pressure ($\beta$)
randn depicts an array of nvar random numbers drawn from standard normal distribution

Step3: Initialization. Randomly generate a population (pop) of particles (also called phenotype) containing the fields:
        •  Velocity (V)
        •  Position(P)
        •  Evaluate Cost depending on position (Cost)
        •  Best Solution obtained by the individual (pBest)
        •  3D Model Plotting of the path
Best solution (gBest) irrespective of the individual is obtained across the generation. The population (pop) is depicted by $nPop \times 1$ matrix. Both GA and PSO refer to the same population, therefore termed as therefore termed as particle genetic memory (also called population or swarm).

Step4: for i = 1 to MaxIt_PSO
      •  Organize the memory according to ascending order of cost.
      •  Calculate best particle from memory (BestCost)
      •  Calculate worst particle from memory (WorstCost)
      •  Carry out local search using embedded GA
    for j = 1 to MaxIt_GA
        •  Assign relative fitness ($Cost_{rel}$) to each individual in the population for carrying out roulette selection
           Sum = 0
          for p = 1 to nPop
           $pop(p).Cost_{rel} = e^{\left(\frac{-\beta * pop\,(p).Cost}{WorstCost}\right)}$
        $Sum = Sum + pop(p).Cost_{rel}$
      end for
      for p = 1 to nPop
      $pop(p).Cost_{rel} = \frac{pop(p).Cost_{rel}}{Sum}$
      end for
    •  Carry out Crossover
      for m = 1 to $N_c/2$
        •  Select two parents $p_1$ and $p_2$ from pop using roulette selection
        •  Perform crossover to get offspring $o_1$ and $o_2$
        for n = 1 to nvar
          •  Randomly select α from the range -γ to γ+1.
          •  Update variable of $o_1$
               $o_1.P(n) = \alpha * p_1.P(n) + (1 - \alpha) * p_2.P(n)$
          •  Update variable of $o_2$
               $o_2.P(n) = \alpha * p_2.P(n) + (1 - \alpha) * p_1.P(n)$
        end for
      end for
    •  Carry out Mutation
      for m = 1 to $N_m$
        •  Select an individual $p_1$ randomly from pop
        •  Perform mutation to get $m_1$
        for n = 1 to nvar
          •  $\sigma = 0.1 * (v_{max} - v_{min})$
          •  Update variable of $m_1$
               $m_1.P(n) = p_1.P(n) + \sigma * randn$
        end for
      end for
    •  Merge the offspring obtained with the existing genetic memory.
    •  Sort the newly created memory in ascending order of cost of particles
    •  Truncate the population by removing unfit individuals
    •  Update the best solution for each particle
    •  Update the global best solution
  end for
  •  Update velocity and position of particles to carry out global search
  for j = 1 to nPop
    •  Update velocity
         $pop(j).V = w * pop(j).V + c_1 * rand * (pop(j).pBest.P - pop(j).P) + c_2 * \ldots$
         $rand * (gBest.P - pop(j).P)$
    •  Update position
         $pop(j).P = pop(j).P + pop(j).V$
    •  Calculate the corresponding cost
    •  Update personal best
    •  Update global best for the generation
  end for
Apply B-spline curve smoothening method to the global best solution and plot it on 3D map
end for
Step5: Visualize. Plot the variation in global best cost with each generation.
```

4.2 Equations governing proposed PSO/HSA are described as under

The velocity of each particle in the swarm is updated using the following equation [7]:

$$v_i(t+1) = w\, v_i(t) + c_1 r_1 [\hat{X}_i(t) - x_i(t)] + c_2 r_2 [g(t) - x_i(t)] \tag{10}$$

where $i = 12 \ldots N, w$ is a parameter called the *inertia weight* and are positive constants, referred to as *cognitive* and *social* parameters, respectively, and are random numbers, uniformly distributed in [0,1]. The inertia weight in (3) is employed to manipulate the impact of the previous history of velocities on the current velocity.

The scheme for adaptive acceleration coefficients, assuming the social velocity model:

$$c2(t) = c2,\min + c2,\max/2 + c2,\max - c2,\min/2 + e - mi(t)\ - 1\ /(e - mi(t)\ + 1) \tag{11}$$

This strategy focuses on exploration in the early stages of optimization, while encouraging convergence to a good optimum near the end of the optimization process by attracting particles more towards the neighbourhood best (or global best) positions. The values of $c1(t)$ and $c2(t)$ at time step t are calculated as:

$$c1(t) = (c1,\min - c1,\max)\ * t/\,nt + c1,\max \tag{12}$$

$$c2(t)\ =\ (c2,\max\ - c2,\min) *\ t\ /nt + c2,\min \tag{13}$$

A number of theoretical studies have shown that the convergence behaviour of PSO is sensitive to the values of the inertia weight and the acceleration coefficients. These studies also provide guidelines to choose values for PSO parameters that will ensure convergence to an equilibrium point. The first set of guidelines are obtained from the different constriction models suggested by Clerc and Kennedy. For a specific constriction model and selected φ value, the value of the constriction coefficient is calculated to ensure convergence. For an unconstricted simplified PSO system that includes inertia, the trajectory of a particle converges if the following conditions hold:

$$1\ > w > 1.2 \tag{14}$$

$$(\varphi 1 + \varphi 2) - 1 > 0 \tag{15}$$

and if:

$$0 < w < 1 \tag{16}$$

Then

$$\varphi 1\ =\ c1U(0,1) \text{ and } \varphi 2 = c2U(0,1) \tag{17}$$

The acceleration coefficients $c1$ and $c2$ serve as upper bounds of $\varphi 1$ and $\varphi 2$.

Equation (14) and (15) can be combinedly be rewritten as in (18)

$$1 > w > 12\,(c1 + c2) - 1 > 0 \tag{18}$$

Therefore, if w, c1 and c2 are selected such that the condition in equation holds, the system has guaranteed convergence to an equilibrium state. The heuristics above have been derived for the simplified PSO system with no stochastic component. It can happen that, for stochastic $\varphi 1$ and $\varphi 2$ and a w that violates the condition stated in equation, the swarm may still converge. The stochastic trajectory illustrated in Fig. 16.6 is an example of such behaviour. The particle follows a convergent trajectory for most of the time steps, with an occasional divergent step. Van den Bergh and Engelbrecht show in that convergent behaviour will be observed under stochastic $\varphi 1$ and $\varphi 2$ if the ratio is as in (19).

$$\varphi\text{ratio} = \varphi\text{crit}/(c1 + c2) \tag{19}$$

is close to 1.0, where

$$\varphi\text{crit} = \varphi|0.5/\varphi - 1\ <\ w,\ \varphi \in (0, c1\ + c2] \tag{20}$$

It is even possible that parameter choices for which φratio = 0.5 may lead to convergent behaviour, since particles spend 50% of their time taking a step along a convergent trajectory.

The usage of harmony memory is important, as it is similar to the choice of the best-fit individuals in genetic algorithms (GA). This will ensure that the best harmonies will be carried over to the new harmony memory. To use this memory more effectively, it is typically assigned as a parameter in (21):

$$r \text{ accept} \in [0, 1], \tag{21}$$

called harmony memory accepting or considering rate. If this rate is too low, only few best harmonies are selected and it may converge too slowly. If this rate is extremely high (near 1), almost all the harmonies are used in the harmony memory, then other harmonies are not explored well, leading to potentially wrong solutions. Therefore, typically, we use

$$r \text{ accept} = 0.7 \sim 0.95 \tag{22}$$

The second component is the pitch adjustment determined by a pitch bandwidth range and a pitch adjusting rate rpa. Though in music, pitch adjustment means

Table 1 Parameter values for the 3D map

Parameter	Value
Starting point	(–10, –10, –10)
Ending point	(10, 10, 10)
Map's centre point	(0, 0, 0)
Number of spherical obstacles	15
Number of cuboidal obstacles	3
Obstacle placement	Random

to change the frequencies, it corresponds to generate a slightly different solution in the Harmony Search algorithm. In theory, the pitch can be adjusted linearly or nonlinearly, but in practice, linear adjustment is used. So, we have

$$xnew = xold + \text{brange} * \varepsilon \tag{23}$$

where xold is the existing pitch or solution from the harmony memory, and xnew is the new pitch after the pitch adjusting action.

The probability of randomization is

$$P\text{ random} = 1 - r\text{ accept} \tag{24}$$

and the actual probability of adjusting pitches is

$$P\text{pitch} = r\text{ accept} * rpa \tag{25}$$

5 Simulation results and analysis

In this section, existing methodologies like particle swarm optimization (PSO), simulated annealing (SA), shuffled frog leap algorithm (SFLA), invasive weed optimization (IWO), artificial bee colony optimization (ABC), imperialist competitive algorithm (ICA) and firefly algorithm (FA) have been used for comparison to the hybrid algorithms designed. In this section, results obtained from all nine methodologies have been analysed and compared. For the comparison to be fair, all the simulations were carried out using MATLAB R2016a on an intel core i7 processor with 2.2 GHz CPU and 8 GB RAM. Tables 1 and 2 show the parameter values for PSO-HSA algorithm [21]. Table 3 shows the parameter values for PSO-GA algorithm [21].

A three-dimensional map containing several obstacles and prohibited areas is created for implementation and subsequent comparison of different path planning methodologies. In each case, the starting point is (–10, –10, –10) and the end point is (10, 10, 10). Moreover, each algorithm is run for the same amount of iterations and with the same number of individuals. Figure 1 shows the flightpath of drone obtained by using PSO algorithm in a

Table 2 Parameter values for proposed hybrid PSO-HSA algorithm

Parameter	Value
Maximum iteration (PSO)	100
Inertia weight	0.729
Inertia weight damping ratio	0.98
Personal learning coefficient	1.494
Global learning coefficient	1.494
Population size	100
Velocity variance parameter	0.942
Maximum iteration (HSA)	20
Harmony memory size	100
Harmonic memory consideration rate	0.9
Pitch adjustment rate	0.1

Table 3 Parameter values for proposed hybrid PSO-GA algorithm

Parameter	Value
Maximum iteration (PSO)	100
Inertia weight	0.729
Inertia weight damping ratio	0.98
Personal learning coefficient	1.494
Global learning coefficient	1.494
Population size	100
Velocity variance parameter	0.942
Maximum iteration (GA)	20
Crossover percentage	0.9
Mutation percentage	0.3

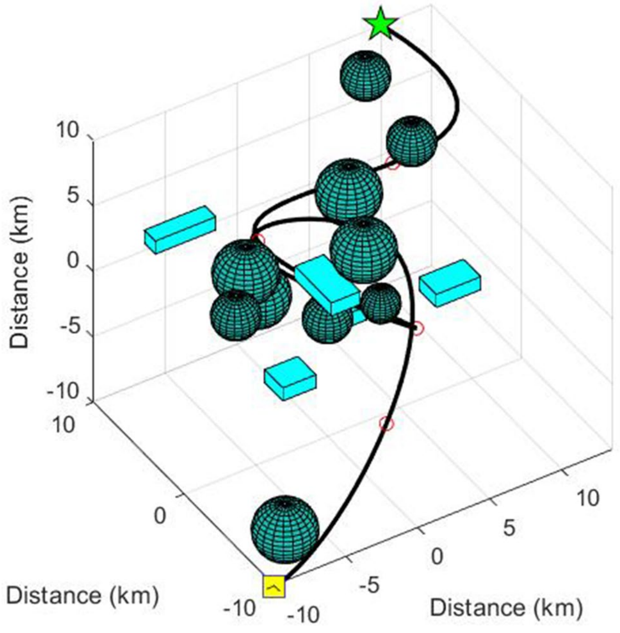


Fig. 1 Path obtained for PSO algorithm

simulated obstacle-filled environment [7, 12, 13]. The PSO is a swarm-based algorithm, the individuals of the population learn with each iteration. The algorithm has been tuned for explorative search by setting the initial velocity possessed by the particles high. The velocity afterwards undergoes dampening to carry out exploitative search. The result henceforth obtained lies in the global optima region but lacks in depth search. Figure 2 shows the flightpath of drone obtained by using SA algorithm in a simulated obstacle-filled environment. Simulated annealing is modelled after the physical phenomenon of annealing, which refers to the heating and subsequent cooling of the material to remove internal stresses. The temperature and temperature reduction rate decide the type of search undertaken. For an explorative search, the temperature is kept high but due limited computational time and slow temperature reduction rate the solution henceforth obtained by SA algorithm does lie in the region of global optima but does not give an exact optimal solution.

Figure 3 shows the flightpath of drone obtained by using SFLA algorithm in a simulated obstacle-filled environment. SFLA is a memetic meta-heuristic algorithm modelled after the behaviour frogs. SFLA algorithm is adept in both exploitative and explorative search but requires high computational time as the population is divided into memeplexes with each memeplex simultaneously running its own PSO-like method for exploitative search, subsequently the frogs are shuffled between memeplexes and random frogs are generated for explorative search. Hence due to limitation in computational time, the solution obtained by SFLA algorithm lies in the region of global optima but lacks in depth search. Figure 4 shows the flightpath of drone obtained by using IWO algorithm in a simulated obstacle-filled environment. IWO is a meta-heuristic algorithm modelled after colonization of invasive weeds. The algorithm involves generating offspring based on fitness of a weed, the offspring thus generated are

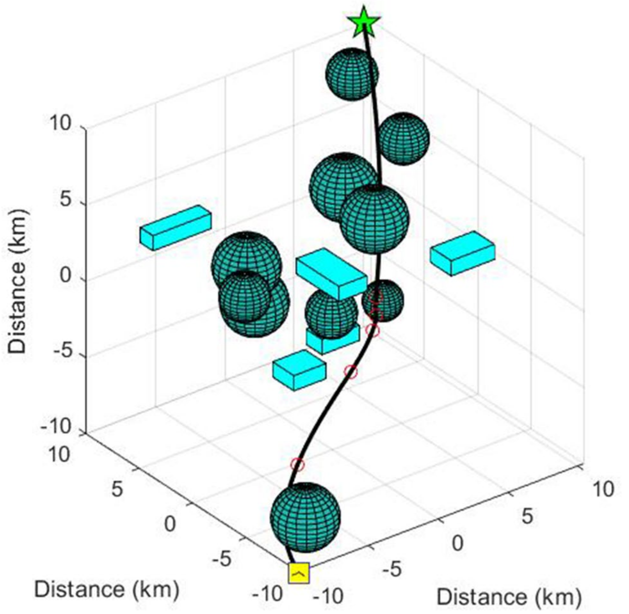


Fig. 3 Path obtained for SFLA algorithm

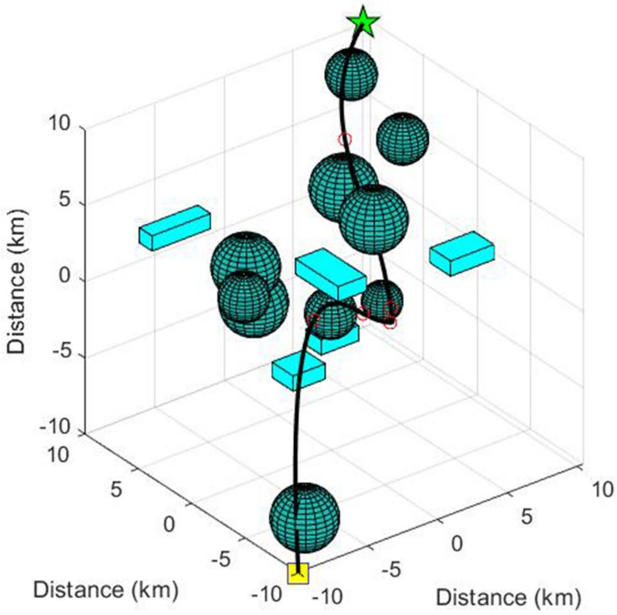


Fig. 2 Path obtained for SA algorithm

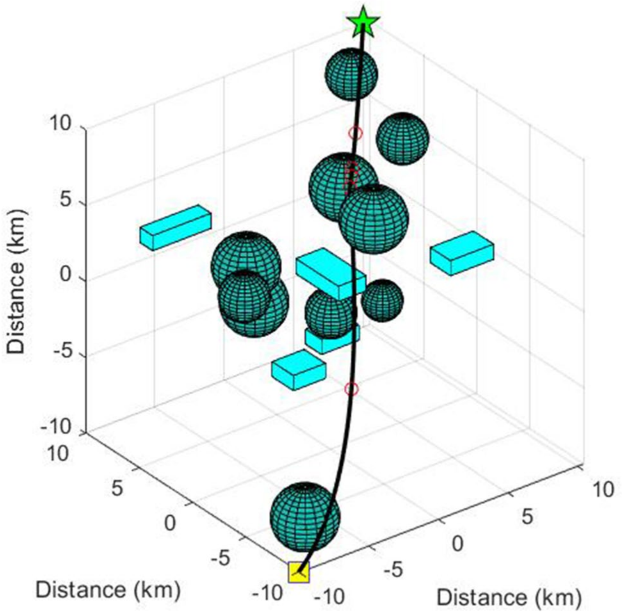


Fig. 4 Path obtained for IWO algorithm

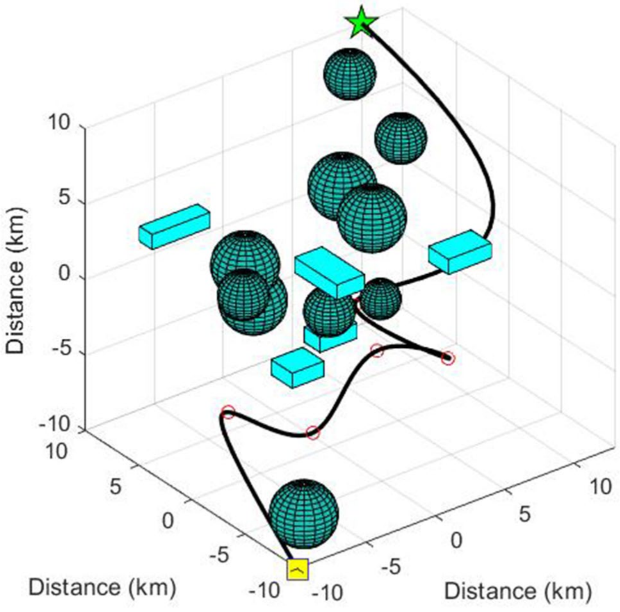


Fig. 5 Path obtained for ABC algorithm

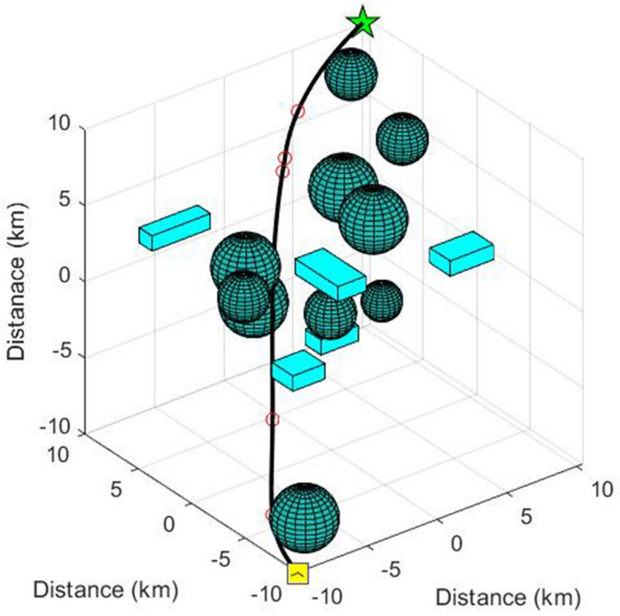


Fig. 6 Path obtained for ICA algorithm

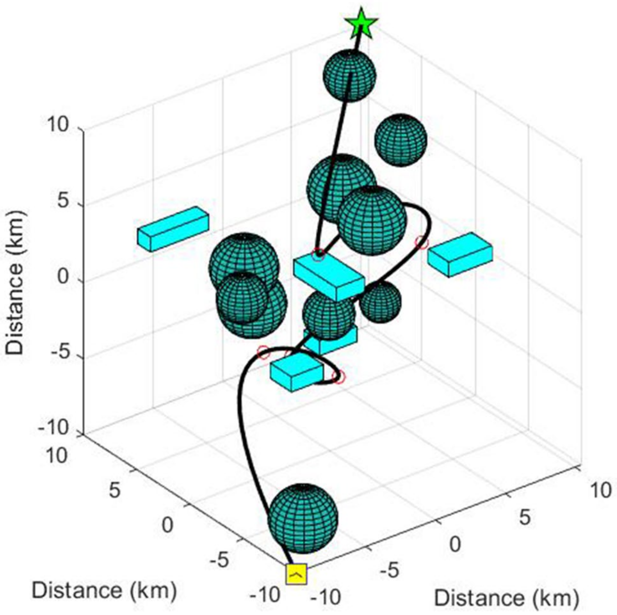


Fig. 7 Path obtained for BA Algorithm

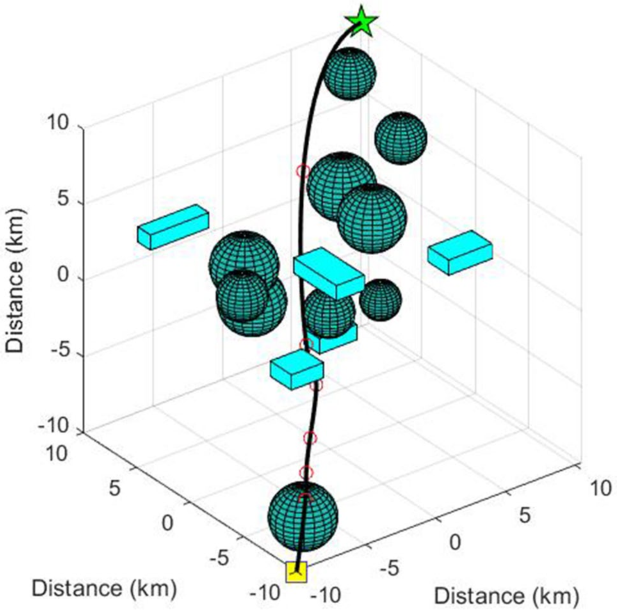


Fig. 8 Path obtained for FA Algorithm

randomly distributed about the parent with the standard deviation reducing with each iteration helping in quicker convergence. Here in the solution obtained fails to achieve global optima as the fittest weed produces corresponding number of seeds contrary to a fixed number, leading to much higher deviation from the solution and subsequent premature termination due to limitation of computational time.

Figure 5 shows the flightpath of drone obtained by using ABC algorithm in a simulated obstacle-filled environment [15]. ABC is swarm-based meta-heuristic algorithm modelled after the behaviour of bees. The bees have been segregated into three groups, employed bees, scout bees and onlooker bees. The employed bees and onlooker bees are used for exploitative search whereas the scout bees are used for explorative search. Due to a limitation on the

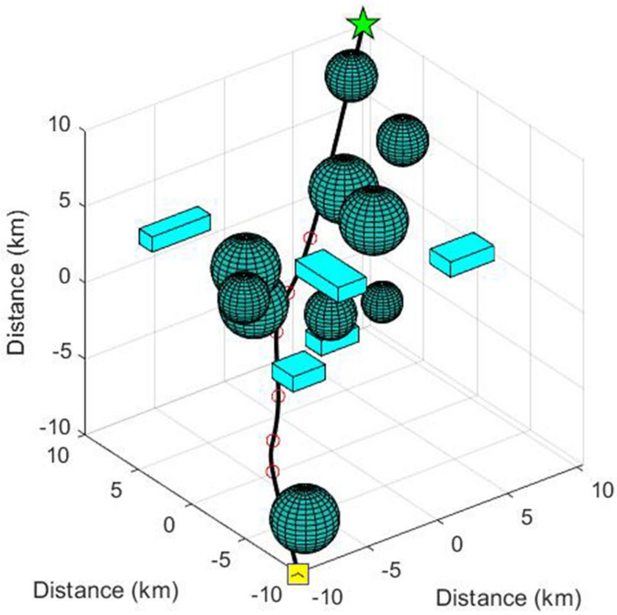


Fig. 9 Path obtained for PSO-HSA Algorithm

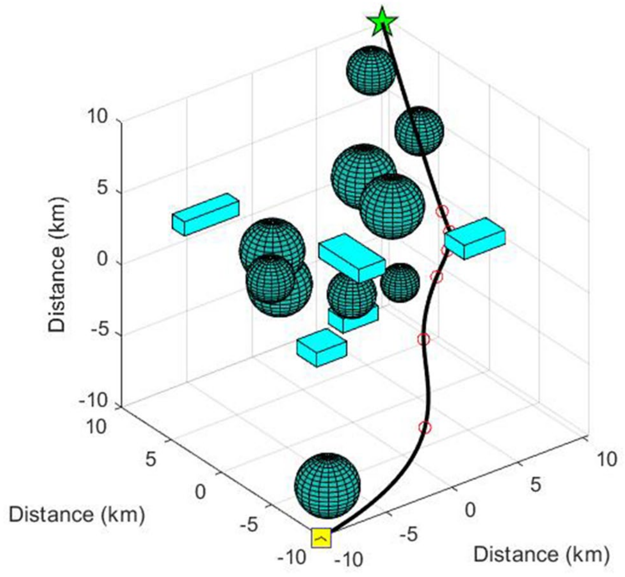


Fig. 10 Path obtained for PSO-GA Algorithm

population, a choice must be made between exploration and exploitation. Here in more individuals were allocated for exploitative search, hence the solution obtained gets prematurely terminated at local optima and the complete search space is not explored. Figure 6 shows the flightpath of drone obtained by using ICA algorithm in a simulated obstacle-filled environment. ICA is an evolutionary algorithm inspired by imperialistic competition. The individuals are modelled after countries which subsequently form empires. The empire consists of colonies and an imperialist which is the fittest individual. Gradually with each iteration the colonies assimilate with the fittest empire, subsequently leading to convergence. ICA focuses on exploitative search which is carried out by the colonies, hence the solution obtained does not lie in the global optimum region due to premature termination at local optima.

Figure 7 shows the flightpath of drone obtained by using BA algorithm in a simulated obstacle-filled environment [20, 21]. BA is a variation of ABC, modelled after foraging behaviour of honey bees. The algorithm balances both exploitative and explorative search but has to be tuned in favour of one due to limited computational time and population. The scout bees carry out exploration and forager bees carry out exploitation. Here in the focus is on exploitation that is forager bees are more in number leading to a solution which does not lie in global optimum region due to premature termination at local optima. Figure 8 shows the flightpath of drone obtained by using FA algorithm in a simulated obstacle-filled environment [1]. FA is meta-heuristic algorithm modelled after the social behaviour of fireflies. The less fit fireflies are attracted towards much fitter or brighter fireflies, leading to convergence. The focus in FA is on exploitative search as the less fit fireflies tend to converge at the fittest solution instead of carrying out exploration. Thus, the solution obtained is prematurely terminated at local minima.

Figure 9 shows the flightpath of drone obtained by using the proposed hybrid PSO-HSA algorithm in a simulated obstacle-filled environment. The hybrid algorithm strikes a perfect balance between exploitation and exploration, with each being allocated its own specific algorithm. The PSO has been tuned for exploration, by imparting the individuals with a high velocity component leading to much higher variation in the new solution generated. The HSA meanwhile focuses on exploitation around a solution by carrying out crossover. Hence, the solution obtained lies in the global optimum region with a much fitter solution because of depth search carried out by HSA. Figure 10 shows the flightpath of drone obtained by using the proposed hybrid PSO-GA algorithm in a simulated obstacle-filled environment. The hybrid algorithm strikes a perfect balance between exploitation and exploration, with each being allocated its own specific algorithm. The PSO has been tuned for exploration, by imparting the individuals with a high velocity component leading to much higher variation in the new solution generated. The GA meanwhile focuses on exploitation around a solution by carrying out crossover and mutation. Hence, the solution obtained lies in the global optimum region with a much fitter solution because of depth search carried out by GA. On analysing all nine pathways generated, we observed that the use of hybrid algorithm leads to an optimal path

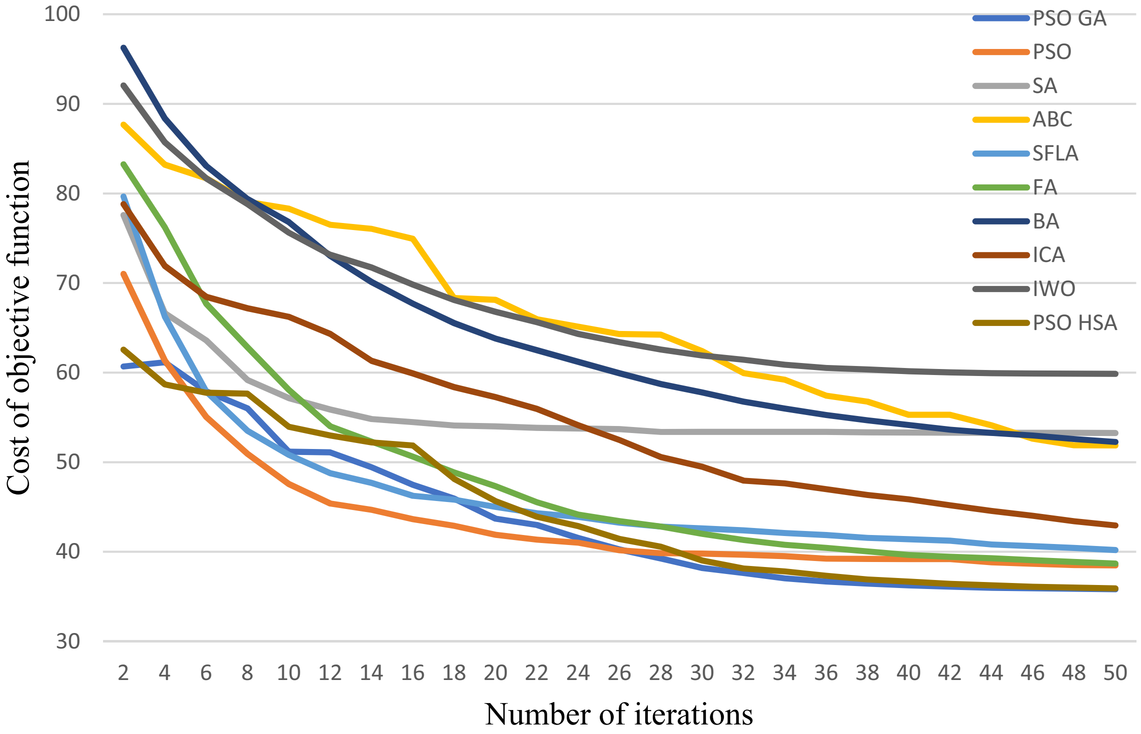
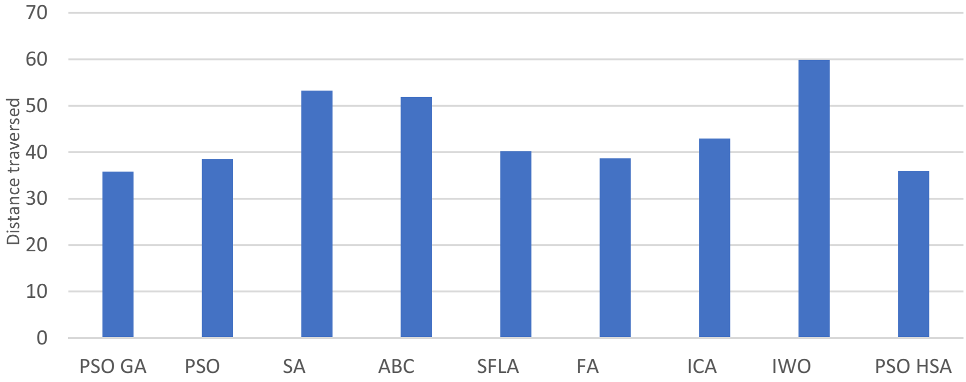


Fig. 11 Comparison of variation in cost

Fig. 12 Distance traversed in various algorithms



satisfying all the constraints as seen in Figs. 9 and 10. Whereas other algorithms give a sub-optimal solution. In some cases, there is a premature convergence at a local minima. In other cases, even though the algorithm correctly isolates the global minimum region in the search space, it is not able to give an exact answer.

Figure 11 has been obtained by averaging results of 10 simulations, with each algorithm being run for 50 iterations. The variation in fitness value with respect to iterations carried out by different algorithms is highlighted in Fig. 11. We observe that hybrid algorithms give a better cost compared to the others, which can be translated to a path that is shorter as shown in Fig. 12. The power received by the drone has been highlighted in Fig. 13, and it could be inferred that with the optimum distance traversed using PSO-HSA, the drone receives high power as compared to other algorithms except SFLA. However, SFLA travels more distance as compared to PSO-HAS. The variation is due to the trade-off between the exploratory and exploitative search. Moreover, Table 4 highlights the improvement in the proposed algorithm over the existing methods and it is found that the PSO-HSA has displayed more than 10% improvement as compared to the SFLA algorithm.

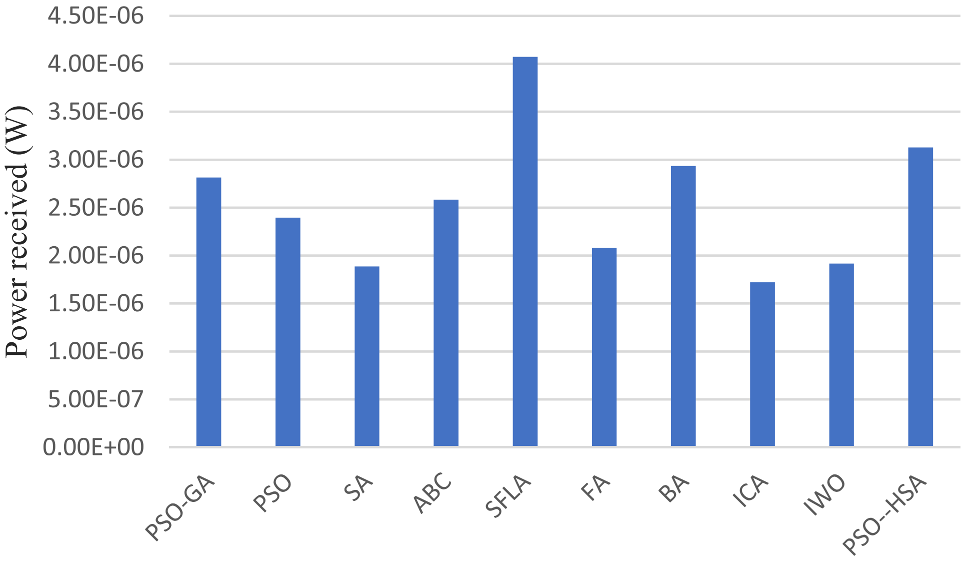


Fig. 13 Power received by drone for different algorithms

Table 4 Percentage improvement in cost of proposed hybrid algorithms over the existing algorithms. All values are in percentage (%)

Algorithm	IWO	SA	ABC	ICA	SFLA	FA	PSO
PSO-HSA	40.01	32.63	31.09	16.44	10.75	7.27	7.01
PSO-GA	40.15	32.74	31.18	16.57	10.9	7.42	7.1

6 Conclusion

This paper proposes two hybrid meta-heuristic algorithms, one based on PSO with HSA, another based on PSO with GA for UAV three-dimension path planning in complicated obstacle-filled environment. From Table 12, we can infer that the proposed hybrid algorithms give a better optimized path, thereby reducing traversal time and the fuel consumed. The path obtained avoids all obstacles and helps in steering the aircraft and its corresponding antenna power signature safely around the hostile antenna to avoid detection. There is an improvement in around 7% when compared to PSO, with up to 40% improvement against IWO. Furthermore, to satisfy turning constraints B-spline curve smoothening method is used.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Roberge Vincent, Tarbouchi Mohammed, Labonté Gilles (2012) Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans Ind Informat 9(1):132–141
2. Xin-She Y, Suash D (2009) Cuckoo search via L´evy flights. In: World congress on nature & biologically inspired computing
3. Michael B, Michael M, Nicole W, Xiao-Hua Y (2010) Ant colony optimization algorithm for robot path planning. In: 2010 international conference computer design and applications (ICCDA), August 2010
4. Gan Y, Qu F, Sun F, He W, Zhang P (2017) Research on path planning for mobile robot based on ACO. In: 2017 29th Chinese control and decision conference (CCDC)
5. Rashmi P, Sujata B, Sharmilla T (2011) Navigational path planning of multi-robot using honey bee mating optimization algorithm (HBMO). Int J Comput Appl
6. Xing Z, Jie C, Bin X, Hao F (2011) Online path planning for UAV using an improved differential evolution algorithm. In: The international federation of automatic control, September 2
7. Mohanty PK, Parhi DR (2014) A new real time path planning for mobile robot navigation using invasive weed optimization algorithm. In: ASME 2014 gas and turbine India conference, December 2014
8. Lai L, Qu S (2012) Path planning for unmanned air vehicles using an improved artificial bee colony algorithm. In: Proceedings of the 31st Chinese control conference, July 2012
9. Ahmed HD, Abdulkader J, Mariam K (2018) Using the bees algorithm for wheeled mobile robot path planning in an indoor dynamic environment. Cogent Eng
10. Saffari MH, Mahjoob MJ (2010) Bee colony algorithm for real-time optimal path planning of mobile robots. In: 2009 fifth international conference on soft computing, computing with words and perceptions in system analysis, decision and control, January 2010

11. Shahab Alam M, Usman Rafique M, Umer Khan M (2015) Mobile robot path planning in static environments using particle swarm optimization. Int J Comput Sci Electron Eng (IJCSEE)
12. Hassanzadeh I, Madani K, Badamchizadeh MA (2010) Mobile robot path planning based on shuffled frog leaping optimization algorithm. In: 2010 IEEE conference automation science and engineering (CASE), October 2010
13. Mohanty PK, Parhi DR (2014) A new efficient optimal path planner for mobile robot based on invasive weed optimization algorithm. Front Mech Eng, 9: 317-330
14. Baolei L, Lanjuan L, Qinhu Z, Danjv L, Yufeng Z, Junhua Z, Xinling S (2014) Path planning based on firefly algorithm and Bezier curve. In: 2014 IEEE international conference on information and automation (ICIA), October 2014
15. Zheng Z, Karl S, Andrew L, Fangpo H, Youhang T (2015) Imperialist competitive algorithm for AUV path planning in a variable ocean. In: Applied artificial intelligence, April 2015
16. Dao TK, Pan TS, Pan JS (2016) A multi-objective optimal mobile robot path planning based on whale optimization algorithm. In: 2016 IEEE 13th international conference on signal processing (ICSP), IEEE, pp 337–342
17. Mihir KR, Deepak BBVL (2015) PSO based system architecture for path planning of mobile robot in dynamic environment. In: 2015 global conference communication technologies (GCCT), December 2015
18. Mehrabian AR, Caro L (2006) A novel numerical optimization algorithm inspired from weed colonization. Ecol Inform 1(4):355–366
19. Wang Y, Caia F, Wang Y (2017) Application of particle swarm optimization in path planning of mobile robot. In: AIP Conference Proceedings
20. Abdurrahim S, Emre K, Emin K (2015) Optimal path planning for UAVs using genetic algorithm. Unmanned Aircraft Systems (ICUAS), 09 July 2015
21. Eappen G, Shankar T (2020) Hybrid PSO-GSA for energy efficient spectrum sensing in cognitive radio network. Phys Commun 101091

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.