Research Square

Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

Multiple UAVs Three-Dimensional Navigation Using a Hybrid Optimization Algorithm

Upma Jain
Graphic Era University

Ram Kishan Dewangan (ramadew7@gmail.com)
Thapar University

Shailendra Tiwari
Thapar University

Jay Prakash
National Institute of Technology Calicut

Research Article

Keywords: UAV, Whale optimization algorithm, sine cosine algorithm, navigation, Deterministic, Meta-heuristic, Hybrid Algorithm

Posted Date: April 14th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1545802/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Noname manuscript No.
(will be inserted by the editor)

Multiple UAVs Three-Dimensional navigation Using a Hybrid Optimization Algorithm

Upma Jain · Ram Kishan Dewangan · Shailendra Tiwari · Jay Prakash

Received: date / Accepted: date

Abstract This paper aims to propose a hybrid approach for multiple Unmanned Aerial Vehicle navigation. This is an NP-hard problem since the robots have to find the optimal safe path without colliding with other robots and obstacles in three dimensional search space. The proposed approach enhances the exploration capabilities of whale optimization algorithm then hybrid this improved whale optimization algorithm with the sine cosine algorithm to improve the overall exploitation capabilities. The efficiency of proposed hybrid approach is compared with other meta-heuristic algorithms for multi-UAV navigation. Results obtained through simulation ensure the validity of the proposed approach.

Keywords UAV · Whale optimization algorithm · sine cosine algorithm · navigation · Deterministic · Meta-heuristic · Hybrid Algorithm

1 Introduction

Robotics has applied extensively in various real-life applications. Use of robots for searching the toxic gas leak, coal mines, fire explosive and planetary exploration has reduced the risk of human lives. The most crucial challenge for any robot is to design a path that will allow it to travel from its starting point to its destination without colliding with obstacles or other robots [1–4]. A path contains the arrangement of multiple points in a continuous grid between the start point and final destination point. Designing such a path is among one of the most important problem in multi-robot system. navigation for multi-robot system has been

Graphics Era University, Dehradun
E-mail: upma.jain88@gmail.com

Thapar Institute of Engineering and Technology, Patiala
E-mail: ramadew7@gmail.com

Thapar Institute of Engineering and Technology, Patiala
E-mail: shailendra@thapar.edu

NIT Calicut
E-mail: jayprakash@nitc.ac.in

studied well in the literature [1], recent work indicated the applicability of swarm intelligence in the navigation [5–7]. The problem of navigation is categorized as:

1. Online navigation: while travelling towards the target, develops the path [6, 7, 5].
2. Offline navigation: path is generated before a robot starts moving [6–8].

navigation for multi-UAV requires the generation of aerial trajectories for multiple aerial robots. navigation in 3D is a complex problem and falls under the category of NP-Hard problem. A path contains the sequence of multiple points in a continuous grid between the start point and final destination point. For autonomous mobile robots, 3D navigation algorithms include randomly exploring algorithms [9], visibility graphs [10], and heuristic search algorithms:: A* and D* [11, 12], Meta heuristic methods such as GWO [13–15], PSO [16, 17], GSO [18] *etc.*, and deterministic search algorithm 'Dijkstra' [19] navigation can be thought of as an optimization issue in which the goal is to identify the best path from the starting point to the destination. The following are the general steps in navigation:

1. The first stage is to portray the workspace, which comprises a demonstration of the workspace's free and occupied places. A continuous grid is commonly used to represent the workspace.
2. The next stage is to plan a path of linked points from an initial place to a specific goal site that is free of conflicts. A deterministic or non-deterministic algorithm is used to accomplish this.
3. The final stage is to create a smooth trajectory from the intended path that can be used to execute many UAVs in real time. The B-spline is used to create a continuous and smooth path from the chosen locations.

2 literature

Initially, the single robot navigation was well studied by the researchers [20, 5, 8, 6, 7]. As the technology advanced and robots have been applied in various real life applications including search [21], planetary exploration[21], gaming [21], farming [21] and service robots [21]. Therefore, multi-robot system had caught the attention of researchers.

Significant progress in the field of single and multi-robot navigation have been done in the last three decades by using traditional and heuristic methods such as graph and Voronoi diagram based methods [10], potential field based methods [22, 23], neural network [3, 25], simulated annealing [26], A* algorithm [11] and evolutionary algorithms [20].

The necessity for three-dimensional (3D) navigation is gaining attention among researchers as UAV agility improves. navigation in 3D is a complex problem and falls under the category of NP-Hard problem. A path contains the sequence of multiple points in a continuous grid between the start point and final destination point. Various methods have been proposed by researchers to deal with the problem of UAV navigation such as A* algorithm [11] , artificial potential field method [22, 23], genetic algorithm [24], particle swarm optimization [16, 17] grey wolf optimizer [13–15, 27] and so on. They can be classified as traditional and meta-heuristic optimization based methods. Various 2D navigation algorithms for autonomous

mobile robots include randomly exploring algorithms [12], visibility graphs [10], heuristic search algorithms : algorithms that use meta heuristics, such as GWO [13–15,27], PSO [29,16], GSO [18] *etc.*, A* and D* [11], and deterministic search algorithm 'Dijkstra' [19] were extended to apply in 3D.

Classical techniques, on the other hand, have a high time complexity and are more prone to get locked in a local optimum. As a result, meta-heuristic techniques to 3D navigation have grown in popularity.

As mentioned earlier, meta-heuristics have applied extensively in the field of 3D navigation due to their ability to produce optimal results with lesser complexity. In this work, a mixture of the modified whale optimization algorithm and the sine-cosine algorithm is used to solve the problem of multi-UAV navigation.

A hybrid of WOA [28] and SCA [29] is proposed for multiple Unmanned Aerial Vehicle navigation. The proposed approach enhances the exploration capabilities of WOA then hybrid this improved with the SCA optimization algorithm. To improve the exploitation around the search space, SCA is hybrid with improved WOA.

3 Problem Formulation

The function of initial and final positions of UAVs that generates trajectories for each UAV is defined as the multi-UAV 3D navigation problem:

$fun(\text{initial}_{positions_i}, final_{positions_i}) \rightarrow trajectory_i$

Where initial represents the starting point (x_m, y_m, z_m) of the i_{th} UAV and $final_{positions}$ represents the target point (x_t, y_t, z_t) of the i_{th} UAV and trajectory represents the feasible conflict free path for i_{th} UAV.

The primary goal of navigation is to find the best collision-free path with the shortest cost path length (σ_i). The initial position matrix for m UAVs is given as follows:

$$posn = \begin{bmatrix} posn_1^1 & posn_2^1 & \dots & \dots & posn_m^1 \\ posn_1^2 & posn_2^2 & \dots & \dots & posn_m^2 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ pos_1^n & pos_2^n & \dots & \dots & pos_m^n \end{bmatrix} \tag{1}$$

Where $posn_i$ denotes the position i^{th} UAV in n dimensional space. Since the search space is 3D, therefore, here $N = 3$. The purpose is to minimize the path length (σ_i) for each UAV by the optimization problem that follows:

$$(\sigma_1^*, \sigma_2^*, \dots, \sigma_m^*) = \arg \min_{\sigma_1, \sigma_2, \dots, \sigma_m} \sum_{j=1}^{m} cost_j \sigma_j \tag{2}$$

subject to constraint

$$\chi_{ij}(\sigma_i, \sigma_j) = 0 \qquad \forall\, i, j = 1, 2, \dots, m \tag{3}$$

Where i and j denotes the different UAVs. σ_i represents the path of ith UAV and is calculated as follows:

$$\sigma_i = \sqrt{(x_s - x_f)^2 + (y_s - y_f)^2 + (z_s - z_f)^2} \tag{4}$$

The cost of the path σ_j is represented by $cost_j$, and the violation of comparable path restrictions between the trajectories of i_{th} and j_{th} UAVs is represented by χ_{ij}.

3.1 Scenario Construction

The first stage in multi-UAV navigation is to build the graph-based structure around the UAV or a numerical model. Three scenarios (maps) were examined for the simulation method in this work. Table 1 shows the boundary points for each of the three scenarios. All barriers and UAVs are included in the boundary representation, which includes a starting and ending boundary location. Table 2 depicts the start and goal positions of UAVs, whereas Table 3 specifies the obstacle representation for various scenarios. − relates to the absence of obstacles in Table 3.

Table 1: Boundary points depiction in 3D scenario.

Scenario	Starting Point of the Boundary Point	Ending Point of the Boundary Boundary
Scenario1	(0, -5,0)	(10,20,6)
Scenario2	(0,0,0)	(20,5,6)
Scenario3	(0, -5,0)	(10,20,6)

Table 2: Representation of the UAV's start and goal points.

Scenario	Starting Point	Goal Point
Scenario1	(2,10,2),(1,-4,1),(9.2,17,3),(9.2,10,3),(0.1,10,2)	(1,-4,1),(0.1,17,3),(9,-4,1),(0.9,-4,5),(9,10,2)
Scenario2	(0,1,5),(0,2,5),(0,3,5),(19,4,5),(19,5,5)	(19,0,5),(19,5,5),(19,4,5),(0,3,5),(0,1,5)
Scenario3	(2,10,2),(1,-4,1),(9.2,17,3),(9.2,10,3),(0.1,10,2)	(1,-4,1),(0.1,17,3),(9,-4,1),(0.9,-4,5),(9,10,2)

4 Methodology for UAV navigation

4.1 UAV Path Generation

The modelling of the workspace as a distinct environment is the first stage in navigation. The environment is divided into equal-sized cells, which are referred to as 'grid '. Because it is ideal for graph representation, the discrete environment representation was chosen. A UAV path is described as a series of continuous points that run from the source point to the target point. To generate the path for UAV, it is expected that UAV will choose one of the six potential ways to go from a given place. For example, from a point $N(x, y, z)$, the possible six neighboring points to choose are as follows:

Table 3: Obstacle visualization in 3D scenario.

Obstacle	Scenario1	Scenario2	Scenario3
1	(0,2,0) - (10,2,5,1.5)	(3.1,0,2.1) - (3.9,5,6)	(0,2,0) - (10,2,5,1.5)
2	(0,2,4.5) -(10,2.5,6)	(9.1,0,2.1) - (9.9,5,6)	(0,2,4.5) -(10,2.5,6)
3	(0,2,1.5) -(3,2.5,4.5)	(15.1,0,2.1) - (15.9,5,6)	(0,2,1.5) -(3,2.5,4.5)
4	(7,2,1.5) - (10,2.5,4.5)	(0.1,0,0) - (0.9,5,3.9)	(7,2,1.5) - (10,2.5,4.5)
5	(3,0,2.4) - (7,0.5,4.5)	(6.1,0,0) - (6.9,5,3.9)	(3,0,2.4) - (7,0.5,4.5)
6	(0,15,0) - (10,20,1)	(12.1,0,0) - (12.9,5,3.9)	(0,15,0) - (10,20,1)
7	(0,15,1) - (10,16,3.5)	(18.1,0,0) - (18.9,5,3.9)	(0,15,1) - (10,16,3.5)
8	(0,18,4.5) - (10,19,6)	-	(0,18,4.5) - (10,19,6)
9	-	-	(0,-2,0) - (10,-1.5,1.5)
10	-	-	(0,-2,3) - (10,-1.5,5.5)
11	-	-	(0,7,0) - (10,7.5,0.5)
12	-	-	(0,7,2) - (10,7.5,5.5)
13	-	-	(0,11,0) - (10,11.5,2.5)
14	-	-	(0,11,4) - (10,11.5,5.5)
15	-	-	(0,-2,1.5) - (3,-1.5,3)
16	-	-	(6,-2,1.5) - (10,-1.5,3)

$$\begin{aligned} &P(x+1,y,z), Q(x-1,y,z), R(x,y+1,z), \\ &S(x,y-1,z), T(x,y,z+1), U(x,y,z-1) \end{aligned} \tag{5}$$

To get to the next point in the 3D workspace, 1 has been added or subtracted.

Following Algorithm 1, have been used to initialize the path matrix 'P'. Here source point is 'S' and 'N' represents the upper bound on maximum number of points this matrix could have.

The Algorithm 1 is as follows:

Algorithm 1 Random neighbor selection based initial solutions generation.

1: Initialize: source point 'S' and path matrix 'P' and set
2: $curnt - point =$ 'S';
3: Ngbr $n =$ getNeighbors(S);
4: **for** all (j= 1 to m) **do**
5: Select n such that $n(j) \cup n(j+1) = \phi$
6: Select $rand(n(j))$;
7: $P = P \cup n(j)$;
8: $curnt - point = n(j)$; // update current node
9: **if** ($curnt - point! = goal \| Num - points < N$) **then**
10: go to step 3.
11: **else**
12: go to step 5.
13: **end if**
14: **end for**
15: set $visited - points = unique(visited - points)$
16: Return the path 'P'.

4.2 Proposed Hybrid IWOA-SCA algorithm for multi-UAV navigation

A hybrid of improved WOA and SCA is proposed to for multi UAV navigation. The idea is to utilize the strength of both the approaches. Improved WOA enhances the exploration capabilities by creating a more diverse initial population set. Then exploitation capabilities are improved by merging the shrinking encircling behavior and spiral update behavior of the WOA with the sine cosine algorithm. Parameters of sine cosine are set to ensure the exploitation of the search space.

4.2.1 Whale Optimization Algorithm

Mirjalili and Lewis [28] proposed WOA, a new meta-heuristic algorithm that simulates humpback whale foraging. Humpback whales hunt krill or small fish near the surface by swimming around them in a diminishing circle and blowing characteristic bubbles along a circular or '9'-shaped path. Three different types of behaviour WOA mathematically modelled encircling prey, spiral bubble-net assaulting, and searching for prey. Because the location of the ideal solution is unknown in advance, it operates by treating the current best solution as prey. Other agents attempt to update their position in relation to the best search agent (best solution) after discovering the best search agent.

During the adaptation of multi UAV navigation, the following equations from Whale optimization algorithm have been presented to emulate the spiral bubble net feeding mechanism of the humpback whales:

$$D = |C * posn_p(k) - posn_i(k)| \tag{6}$$

$$posn_i(k+1) = posn_p(k) - A * D \tag{7}$$

Where D represents a vector whose value depends on the best solution, k is the current iteration, $posn_p$ denotes the position vector of the best solution, and $posn_i$ is a solution of ith position vector. Since the search space is 3D, hence the size of each vector is 3.

A and C are two coefficient vectors, and are calculated as follows:

$$A = 2 * a * randm_1 - a \tag{8}$$

$$C = 2 * randm_2 \tag{9}$$

The value of a decreases linearly from 2 to 0 over the course of iterations, and $randm_1$ and $randm_2$ are vectors generated randomly in the course of uniform distribution between [0,1].

Exploitation phase The mathematical model of bubble net feeding mechanism incorporates two different behavior namely, shrinking encircling and spiral update.

1. Encircling behaviour shrinking: This is accomplished by lowering the value of a. It's worth noting that A has a random value between $[-a, a]$. The new position of an agent can be defined between the original position of the agent and the best searching agent by assigning random values of A in the interval [-1, 1].

2. Spiral update behavior: A spiral equation, *i.e.*, based on the distance between the search agent and the best agent, is presented to simulate the humpback whale's helix-shaped movement:

$$posn_i(k+1) = D \cdot \exp(pl) \cdot \cos(2\pi l) + posn_p(k) \tag{10}$$

Where D signifies the distance between the i_{th} and the best solution, and p denotes the logarithmic spiral's shape. The symbol l stands for a random number which take values between [-1, 1], and . represents the element-wise multiplication.
Humpback whales hunt the prey by swimming around it along a spiral path within a shrinking circle simultaneously. To model this, a probability of 50 is chosen to employ any of the behavior either spiral path or shrinking circle. The mathematical model to update agents position is given as follows:

$$posn_i(k+1) = \begin{cases} (posn_p(k)(i) - A * D) & if(prob < 0.5) \\ (D \cdot \exp(pl) \cdot \cos(2\pi l) + posn_p(k) & if(prob >= 0.5) \end{cases} \tag{11}$$

Where $prob$ is a random number that varies between [0, 1].
Exploration phase New locations of search agents are Scenarioped around a randomly selected search agent instead of the optimal search agent to maintain exploration. To accomplish this, the value of A is set to be more than 1 or less than -1. This approach enables the WOA to place a greater emphasis on global search.
The following is the mathematical model:

$$D = |C * posn_{rand}(k) - posn_i(k)| \tag{12}$$

$$posn_i(k+1) = posn_{rand}(k) - A * D \tag{13}$$

Where $posn_{rand}$ is the position of the randomly chosen solution.

4.2.2 Improved whale optimization algorithm

At first, the random initial solutions are divided into multiple groups. These groups are then fed into WOA as initial population and solutions obtained from WOA for multiple groups are merged and used as the initial population in WOA. This helps in enhancing the exploration capability of WOA since the obtained solutions are distributed around multiple good solutions rather than a single best solution.

4.2.3 Sine cosine algorithm

Mirjalili [29] in 2016 presented the Sine Cosine algorithm, which is a population-based optimization technique. This technique uses a mathematical model based on sine and cosine to oscillate solutions outward or inwards towards the ideal result. To balance the exploration and exploitation of the search space during optimization, this technique used several adaptive and random variables.
To simulate the sine-cosine model for multi-UAV navigation, the following equations have been proposed.

$$posn_i(k+1) = posn_i(k) + rand_1 * \sin rand_2 * x \tag{14}$$

$$posn_i(k+1) = posn_i(k) + rand_1 * \cos rand_2 * x \tag{15}$$

Where x is given as follows:

$$x = |rand_3 * posn_p(k) - posn_i(k)| \tag{16}$$

Above two equations are combined to be work together as :

$$posn_i(k+1) = \begin{cases} (posn_i k + rand_1 * \sin rand_2 * x & if(rand_4 < 0.5) \\ posn_i(k) + rand_1 * \cos rand_2 * x & if(rand_4 >= 0.5) \end{cases} \tag{17}$$

Where $rand_4$ is a random number between [0, 1].
There are four main parameters in the equation $rand_1, rand_2, rand_3$ and $rand_4$. $rand_1$ identify the next position region either inside the solution and destination or outside it. $rand_2$ dictates how far the next move should be. The parameter $rand_3$ is used to put a random weight to emphasize ($rand_3 > 1$) or deemphasize ($rand_3 < 1$) the effects of destination. The parameter $rand_4$ allows equal switching between sine and cosine.
Exploration and exploitation To achieve exploitation it is necessary to reposition a solution inside or outside of the region between a solution and destination. This is achieved by defining the range of sine and cosine functions between [-2 ,2]. $rand_2$ is defined between [0, 2π]. To ensure this range of $rand_1$ is defined as follows:

$$rand_1 = a_1 - a_1 * (k/T) \tag{18}$$

The constant a_1 is used here. The current iteration is k, and the maximum number of iterations is T.
Hybrid IWOA-SCA To mathematically model hybrid of WOA and SCA for multi-UAV navigation initial paths generated through the random path generation method are considered as initial solutions. And there fitness is evaluated on the basis of defined objective function. The following equation is used to update the position of solutions at the end of the process:

$$posn_i(k+1) = \begin{cases} (rand_1 * \sin(rand_2) * x + y_k & if(rand_4 < 0.5) \\ rand_1 * \cos(rand_2) * x + y_k & if(rand_4 >= 0.5) \end{cases} \tag{19}$$

Where part-3 (*i.e.*, the value of y_k) the above equation can be written as:

$$y_k = D \cdot \exp(pl) \cdot \cos(2\pi l) + posn_p(k) \tag{20}$$

The algorithm for navigation using Hybrid approach is represented in Algorithm 2.

5 Implementation Procedure

The performance of the proposed hybrid algorithm has been analyzed by performing several experiments with five homogeneous UAVs on PC with $i5$ and 3.4 GHz CPUs. Execution of the proposed hybrid algorithm is performed on the scenarios described in the third section.

Algorithm 2 Hybrid Approach based planning for multi-UAV.

```
Input (Map, initial and goal location of UAVs)
Initialize parameters: no. of solutions: k, max iterations: it_max
Genereate the k random solutions and use the Eqs (1) and (2) to calculate the fitness value
  of each UAV.
while it != it_max do
   Find the best solution posn_p.
   for all (i = 1 to k) do
      Update A and C using Eq. (8) and (9).
      if (prob < 0.5) then
         if (|A| < 1) then
            Update the next position of current solution by Eq.(11).
         else
            Choose a random solution (posn_rand).
            Update the next position of current solution by equation (13).
         end if(prob >= 0.5)
         if (r4 < 0.5) then
            Update by sine rule from eq. (19)
         else
            Update by cosine rule from eq. (19)
         end if
      end if
   end for
end while
Output the most optimized path based on Eq. (17).
```

5.1 Parameter Setting for 3D navigation

The suggested hybrid approach is used to plan the paths of many UAVs in a 3D workspace in this part. To analyse the performance of the proposed method, three different scenarios have been chosen, which are presented in Table 1 of Section 3.

Table 4: Parameter setting for the meta-heuristic techniques.

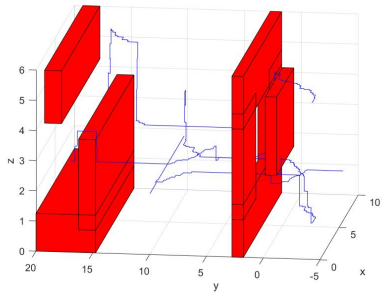
Algorithm	Parameters	Values
BBO	Emigration Prob. (μ)	[1, 0]
	Immigration Prob. (λ)	1-μ
	Probabilistic Mutation	0.1
	Populations	20-30
	Max. Iterations	25-50
IBA	Frequency (Q)	0.5
	Pulse Emission Rate (r)	0.5
	Loudness (A)	0.5
	Populations	20-30
	Max. Iterations	25-50
PSO	Inertia Weight (w)	1
	Inertia Weight Damping Ratio	0.98
	Personal Learning Coefficient (c_1)	1.5
	Global Learning Coefficient (c_2)	1.5
	Populations	20-30
	Max. Iterations	25-50
WOA	Parameter (a)	[0, 2]
	Random (l)	[-1, 1]
	Constant (b)	1
	Random (r)	[0, 1]
	Populations	20-30
	Max. Iterations	25-50
SCA	Random 1 (r_1)	[0, 1]
	Random 2 (r_2)	[0,2* π]
	Random 3 (r_3)	[-1, 1]
	Random 4 (r_4)	[0, 1]
	Populations	20-30
	Max. Iterations	25-50
GWO	Parameter (a)	[2, 0]
	Populations	20-30
	Max. Iterations	25-50

Meta-heuristic algorithms are defined by their ability to deal with socially competent, naive, and cognitively capable agents. The ability to learn from others in the herd is referred to as social competency. It necessitates effective communication among the robots in the herd. Each person's cognitive ability is their ability to learn from their experiences. Individuals in naive ability complete a random flying/wandering activity with no special instruction.

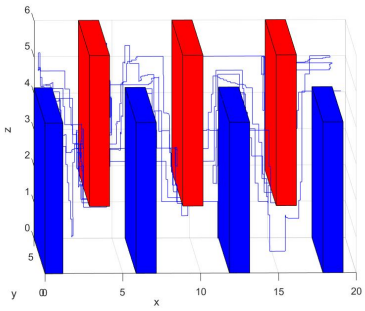
To tackle a specific problem, each meta-heuristic algorithm has its unique set of attributes. Theoretically comparing meta-heuristic algorithms is tricky. For each meta-heuristic method developed in the manuscript, we provide a tabular summary (Table 4) of the underlying parameters considered.

Each of the three cases is subjected to the hybrid approach, and the results are compared to those of nine other algorithms: Dijkstra (deterministic), GWO, SCA, BBO, GSO, WOA, IBA, and PSO. The UAV motions for each Scenario are shown in Fig. 1, while the execution time for the Dijkstra algorithm and various meta-heuristic algorithms for Scenario 1, 2, and 3 are reported in Table 5, 6 and 7, respectively.

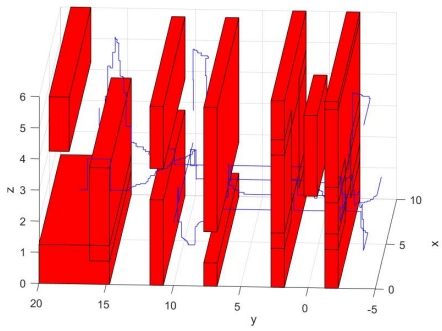
The execution time and average fitness costs for all three scenarios are shown in Figures Fig. 2 and 3. It can be shown that the proposed hybrid algorithm produces results that are similar to those of Dijkstra's method. Furthermore,



(a) UAV path generation for Scenario 1



(b) UAV path generation for Scenario 2



(c) UAV path generation for Scenario 3

Fig. 1: The suggested hybrid method used to calculate UAV movements for Scenario 1, Scenario 2, and Scenario 3.

the suggested hybrid method outperforms the findings of other optimization techniques investigated in the study.

The UAV movement and the algorithm-generated path are depicted in Fig. 4, 5, 6 and 7. The trajectory is the outcome of path smoothing and the proposed hybrid optimization technique, which makes it more flyable for aerial vehicles while also avoiding sharp turns in expected paths.

The suggested method is compared to the Dijkstra, GWO, SCA, BBO, GSO, WOA, IBA, and PSO algorithms in terms of execution time, resulting in a feasible path from source to destination. For all Scenarios, the algorithm is tasked with constructing the path between the two ends. Table 5, 6 and 7 indicate that arbitrary algorithms give deterministic results, however there is a lot of mythology surrounding the time necessary for execution of any meta-heuristic. The tables Table 5, 6 and 7 compare the execution timings of different algorithms with varying start and finish conditions within the workspace. There is no need to specify multiple population sizes or iterations because the Dijkstra technique is deterministic.

It can be deduced from Table 5, 6 and 7, that the proposed hybrid method outperforms other meta-heuristic and deterministic algorithms and requires less time. In most circumstances, the suggested hybrid algorithm outperforms

Table 5: For varied population sizes and maximum iterations, the execution time (in seconds) and best cost (distance travelled in cm) of various algorithms for Scenario 1.

Algorithm	**Pop. Size**	**Iterations**	**Best Cost** (UAV_1) **(cm)**	**Best Cost** (UAV_2) **(cm)**	**Best Cost** (UAV_3) **(cm)**	**Best Cost** (UAV_4) **(cm)**	**Best Cost** (UAV_5) **(cm)**	**Overall Time (sec)**
Dijkstra	-	-	179	274	267	228	90	122.53
IBA	20	25	235	320	325	234	92	58.28
BBO	20	25	231	284	309	228	92	60.08
GSO	20	25	215	295	285	240	94	52.75
PSO	20	25	201	308	295	236	94	56.63
GWO	20	25	217	290	293	234	92	52.14
SCA	20	25	209	308	295	234	92	56.23
WOA	20	25	205	308	313	232	92	65.13
Hybrid	**20**	**25**	**197**	**304**	**296**	**236**	**92**	**51.42**
Dijkstra	-	-	179	274	267	228	90	122.53
IBA	25	40	209	282	295	234	92	97.75
BBO	25	40	193	294	349	228	92	97.38
GSO	25	40	193	288	289	230	92	72.23
PSO	25	40	191	302	281	230	92	66.91
GWO	25	40	211	302	275	228	94	65.23
SCA	25	40	201	298	285	234	92	70.33
WOA	25	40	207	286	279	228	92	75.33
Hybrid	**25**	**40**	**191**	**286**	**277**	**234**	**92**	**58.55**

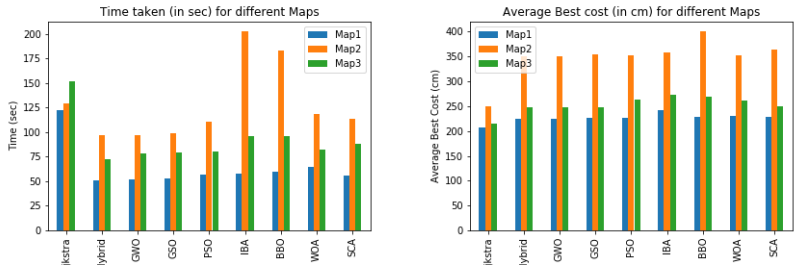
Table 6: For varied population sizes and maximum iterations, the execution time (in seconds) and best cost (distance travelled in cm) of various algorithms for Scenario 2.

Algorithm	**Pop. Size**	**Iterations**	**Best Cost** (UAV_1) **(cm)**	**Best Cost** (UAV_2) **(cm)**	**Best Cost** (UAV_3) **(cm)**	**Best Cost** (UAV_4) **(cm)**	**Best Cost** (UAV_5) **(cm)**	**Overall Time (sec)**
Dijkstra	-	-	239	259	239	239	269	129.30
IBA	20	25	283	311	345	345	509	112.03
BBO	20	25	361	411	427	443	363	101.72
GSO	20	25	319	313	329	345	461	98.88
PSO	20	25	335	337	365	355	369	98.27
GWO	20	25	331	337	347	355	385	97.42
SCA	20	25	329	333	337	389	433	113.33
WOA	20	25	337	337	321	375	393	118.50
Hybrid	**20**	**25**	**315**	**327**	**325**	**369**	**419**	**96.77**
Dijkstra	-	-	239	259	239	239	269	129.30
IBA	30	40	357	381	329	375	423	202.92
BBO	30	40	349	395	365	415	609	183.06
GSO	30	40	295	309	323	361	353	110.09
PSO	30	40	291	327	323	367	347	111.03
GWO	30	40	331	325	303	327	343	109.00
SCA	30	40	319	311	317	381	323	192.13
WOA	30	40	325	323	337	331	317	205.78
Hybrid	**30**	**40**	**301**	**305**	**291**	**321**	**413**	**106.63**

the BBO method in terms of cost and time, and outperforms other algorithms in both parameters.

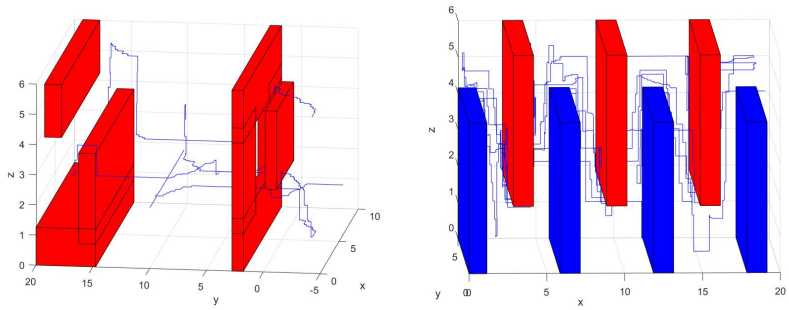
Table 7: For varied population sizes and maximum iterations, the execution time (in seconds) and best cost (distance travelled in cm) of various algorithms for Scenario 3.

Algorithm	**Pop. Size**	**Iterations**	**Best Cost (UAV_1) (cm)**	**Best Cost (UAV_2) (cm)**	**Best Cost (UAV_3) (cm)**	**Best Cost (UAV_4) (cm)**	**Best Cost (UAV_5) (cm)**	**Overall Time (sec)**
Dijkstra	-	-	181	276	289	234	90	152.23
IBA	20	25	249	288	407	334	92	95.61
BBO	20	25	263	310	379	306	92	96.23
GSO	20	25	223	313	367	248	92	79.06
PSO	20	25	235	308	401	280	94	80.19
GWO	20	25	227	326	351	242	96	78.31
SCA	20	25	197	338	369	250	92	82.52
WOA	20	25	239	318	341	318	94	82.05
Hybrid	**20**	**25**	**227**	**312**	**353**	**258**	**92**	**72.83**
Dijkstra	-	-	181	276	289	234	90	152.23
IBA	25	40	245	316	385	252	92	146.80
BBO	25	40	205	320	429	306	92	106.56
GSO	25	40	259	286	337	254	92	85.08
PSO	25	40	221	316	355	258	92	87.30
GWO	25	40	251	308	325	256	92	84.58
SCA	25	40	257	298	337	256	94	88.11
WOA	25	40	217	320	337	272	94	86.14
Hybrid	**25**	**40**	**183**	**300**	**371**	**274**	**92**	**78.31**

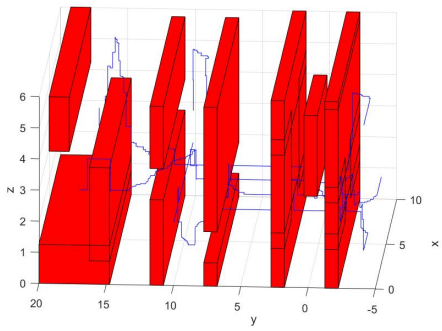


(a) Time taken for different algorithms (b) Average best cost for different algorithms

Fig. 2: Analysis of execution time and average best cost for Scenario 1, Scenario 2, and Scenario 3, using 25 iterations and a population size of 20.

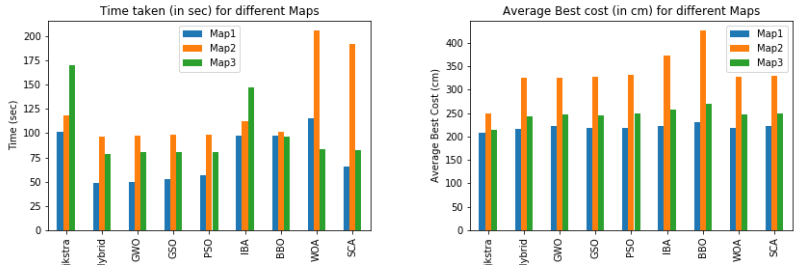


(a) Path generated for Scenario1 (b) Path generated for Scenario2



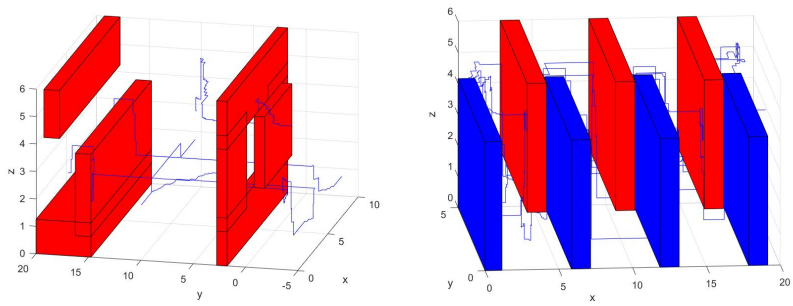
(c) Path generated for Scenario3

Fig. 5: Path constructed using proposed hybrid technique for iterations: 25 and population size: 20 for Scenario 1, Scenario 2, and Scenario 3.

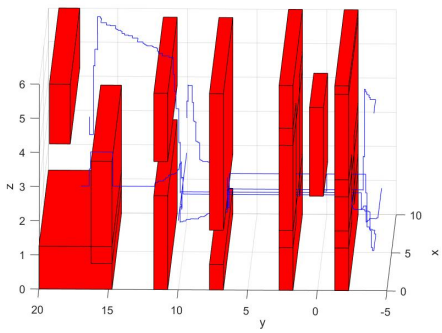


(a) Time taken for different algorithms (b) Average best cost for different algorithms

Fig. 3: Analysis of execution time and average best cost for Scenario 1, Scenario 2, and Scenario 3, using 40 iterations and a population size of 25.

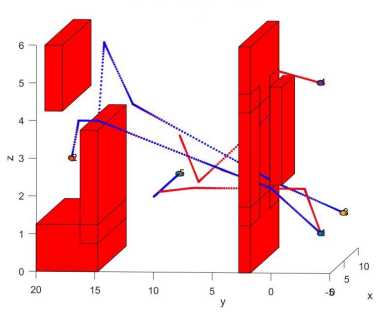


(a) Path generated for Scenario1 (b) Path generated for Scenario2

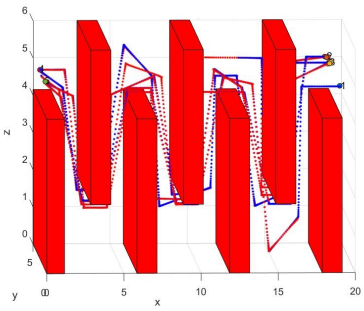


(c) Path generated for Scenario3

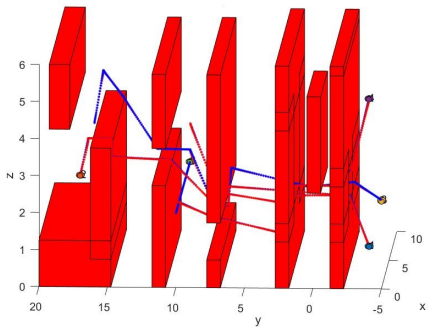
Fig. 7: Path constructed using proposed hybrid technique for iterations: 40 and population size: 25 for Scenario 1, Scenario 2, and Scenario 3.



(a) UAV movements for Scenario1



(b) UAV movements for Scenario2



(c) UAV movements for Scenario3

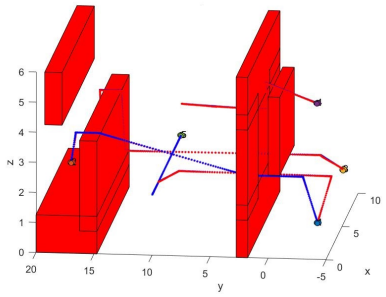
Fig. 4: The Proposed algorithm used to calculate the UAV movements for Scenario 1, Scenario 2, and Scenario 3 for iterations: 25 and population size: 20.

5.2 Convergence analysis of the Hybrid algorithm

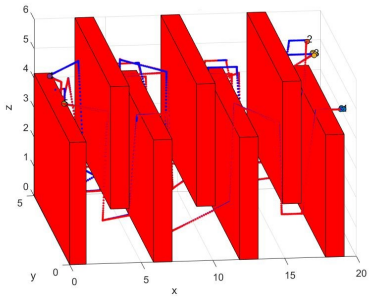
The proposed algorithm works without any constraint of workspace and resulted with a near-optimal solution in fewer iterations, as shown by Table 8, 9 and 10. As, more resistance is put in the workspace more the iteration required by the algorithm for convergence. It is generally observed that proposed approach start producing the results after fifteen to twenty iterations, which is less likely to increase later.

The proposed hybrid approach gives a near-optimal output in fewer iterations without any workspace limits, as shown in Table 8, 9 and 10. When there is more resistance in the workplace, convergence necessitates more iterations. Even if the number of repetitions is unlikely to grow, results are reached within 15 to 20 iterations.

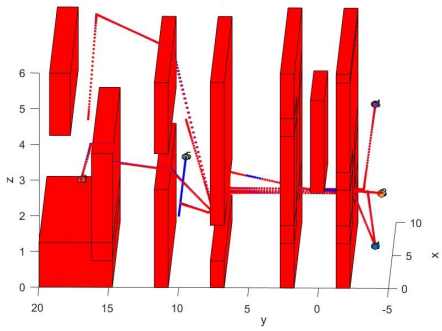
The comparison of the average best cost of the various algorithms after ten iterations is depicted by Fig. 8. From this figure, it can be observed that proposed algorithm is matching with the other meta-heuristic algorithms at some points. Further it can also be observed that in the very beginning iterations some of the UAVs do not reach the destination. We can also infer from these



(a) UAV movements for Scenario1



(b) UAV movements for Scenario2



(c) UAV movements for Scenario3

Fig. 6: The Proposed algorithm used to calculate the UAV movements for Scenario 1, Scenario 2, and Scenario 3 for iterations: 40 and population size: 25.

results that initial average best cost is higher in case of BBO under Scenario 1 and cost is also higher in the IBA under scenarios 2 and 3.

As demonstrated in Fig. 8, the difference in the average best cost of various algorithms increased considerably after 10 iterations, and several elements of the proposed hybrid method overlap with previous meta-heuristic algorithms. Some UAVs fail to reach their target in the first iteration, resulting in a higher initial average best cost in BBO in Scenario 1 and a higher initial average best cost in IBA in Scenario 2 and Scenario 3.

From the results obtained, it can be observed that the with the simple workspace, like scenario 1, the execution time of all the algorithms is compare-able. While with some complex workspace, like scenario 2 and 3, the proposed algorithm achieved better results with less time complexity as compared to others. Hence the proposed algorithm is better and use-full for all real-word multi-UAV applications.

When examining a simple workspace like Scenario 1, the results obtained show that the algorithm's execution time is equivalent. In Scenario 2 and Scenario 3, the suggested hybrid method has a lower time complexity and produces better outcomes for complicated workspaces. As a result, our approach is applicable to multi-UAV applications in the real world.

Table 8: Scenario 1's average best cost (in cm) per iteration count

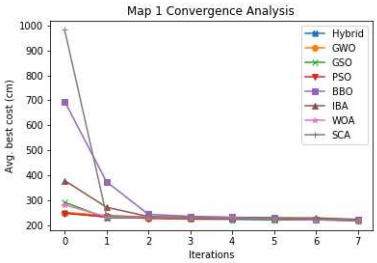
Iteration	**Hybrid**	**GWO**	**GSO**	**PSO**	**BBO**	**IBA**	**WOA**	**SCA**
1	250.80	250.80	291.60	246.80	695.78	378.48	280.8	982.26
5	231.20	238.80	228.40	233.60	373.46	271.60	233.6	238.80
10	230.80	226.40	229.60	227.20	243.20	234.40	230.8	232.00
15	224.00	225.60	225.60	225.80	235.20	231.60	229.6	230.00
20	223.20	225.60	224.80	225.40	231.60	230.80	229.2	226.80
25	222.00	224.50	221.60	224.40	229.20	229.20	224.8	224.40
30	221.20	222.00	221.60	222.00	222.80	228.80	222.4	223.60
35	217.60	218.00	220.40	219.60	221.20	223.20	219.2	218.40

Table 9: Scenario 2's average best cost (in cm) per iteration count

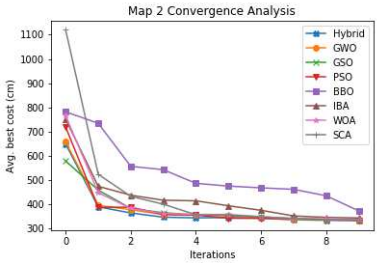
Iteration	**Hybrid**	**GWO**	**GSO**	**PSO**	**BBO**	**IBA**	**WOA**	**SCA**
1	648.76	657.90	578.22	718.56	781.49	751.49	765.18	1120.09
5	389.00	395.00	458.28	387.00	734.23	473.80	447.66	523.47
10	363.55	376.89	382.20	387.60	556.29	437.00	382.20	432.37
15	346.20	356.97	364.60	357.00	542.46	416.20	361.80	399.80
20	343.40	355.80	355.80	354.60	485.96	413.80	357.40	356.20
25	344.60	351.80	353.60	341.80	474.64	393.40	357.40	355.21
30	342.20	345.00	346.20	341.40	467.36	374.60	348.60	347.40
35	335.40	337.00	338.20	339.40	461.26	351.40	339.40	341.80
40	333.40	335.40	336.20	337.40	434.55	345.40	338.40	336.60
45	331.60	331.80	333.80	333.60	372.35	343.00	336.20	334.60

Table 10: Scenario 3's average best cost (in cm) per iteration count

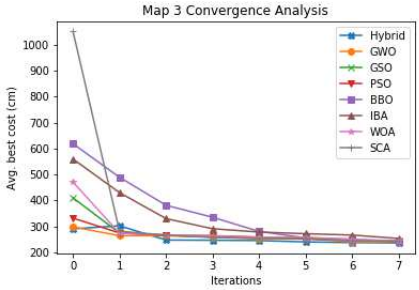
Iteration	**Hybrid**	**GWO**	**GSO**	**PSO**	**BBO**	**IBA**	**WOA**	**SCA**
1	290.40	298.00	409.89	331.81	618.76	559.81	470.69	1052.04
5	302.00	264.00	274.40	274.40	489.40	430.20	272.80	281.20
10	247.20	263.60	265.80	265.60	381.33	330.00	267.20	266.00
15	246.40	258.00	258.00	264.00	335.20	290.80	263.20	258.40
20	245.20	251.20	256.40	258.00	280.40	278.00	258.40	254.40
25	240.00	251.60	252.20	252.40	255.60	271.60	257.60	254.00
30	237.20	240.00	246.00	246.60	250.00	266.40	250.40	244.00
35	236.00	240.00	243.00	241.60	241.20	253.20	243.60	243.60



(a) Convergence analysis for Scenario1



(b) Convergence analysis for Scenario2



(c) Convergence analysis for Scenario3

Fig. 8: Convergence analysis utilising the hybrid algorithm for Scenario 1, Scenario 2, and Scenario 3.

5.3 Comparison with different Meta-heuristic Techniques

The proposed approach comes with the following advantages like flexibility, mechanism without any derivation, and tries to achieve the global optima. Here, flexibility is applicability of the proposed approach to different problem without changing the structure of the algorithm. As compared to the gradient based optimization approach, the proposed approach optimizes the problems stochastically, which makes it derivation free. So, the proposed approach can be used in all the real world situations where derivation finding is costly affair. The third advantages also comes with the stochastic nature of the proposed algorithm. The stochastic nature helps the proposed approach to avoid the local optimum solutions. Also, this advantages makes the proposed solution highly suitable for solving highly nonlinear, multi-variable, multi-modal function optimization problems. Due to the above benefits, proposed approach is suitable for many real-world problems.

Flexibility, derivation-free methods, and avoidance of local optima are some of the benefits of the suggested methodology. The suggested hybrid algorithm's flexibility refers to its capacity to be used to a variety of issues without requiring any changes to the algorithm's structure. Furthermore, the proposed hybrid has a system that is devoid of derivatives. The suggested hybrid optimises problems stochastically, as opposed to gradient-based optimization approaches. There is

no need to calculate the derivative of search locations throughout the optimization phase. This approach will be useful for real-world applications involving expensive or uncertain generated data. Finally, because to the stochastic character of the suggested strategy, local optimization avoidance is higher than with classic optimization strategies. This trait makes the suggested method ideal for handling problems involving highly nonlinear, multi-variable, and multi-modal function optimization. Because of the advantages listed above, the proposed approach is applicable to a wide range of real-world challenges.

We don't know if a meta-heuristic is good at addressing a specific optimization problem since we don't know what it is. As a result, multiple meta-heuristic algorithms are applied in this paper to tackle the real-world multi-UAV navigation problem. To tackle the problem, we tried GWO, SCA, BBO, GSO, WOA, IBA, and PSO, and discovered that the proposed hybrid algorithm is the best of them all. As a result, the proposed method was chosen as the best solution to the multi-UAV navigation problem.

To tackle the UAV navigation problem, we are comparing various theoretical aspects of the proposed approach:

- As the number of iterations increases, strict social behaviour processes aid the proposed approach in storing ideal solutions. The features of the suggested approach make it simple for UAVs to go from exploration to exploitation.
- The first portion of iterations with a falling A value is dedicated to searching the search space, while the second part is dedicated to converting to the corresponding target position.
- In the proposed UAV navigation approach, there is only one parameter to alter.

6 Conclusion

The current study presents a hybrid algorithm-based solution to the navigation problem for numerous UAVs. This method determines the best path between the source and target points while maintaining the best aerial restrictions with static obstructions. Based on the social and optimised behaviour of grey wolves, the programme replicates navigation for several UAVs. When compared to previous meta-heuristic and deterministic algorithms, the Scenario ping to the algorithm provides faster convergence and avoidance of local optima, resulting in lower path computation costs and better results. The proposed solution uses fewer parameters, resulting in a faster convergence time.

References

1. J. S. Bellingham, M. Tillerson, M. Alighanbari, and J. P. How, "Cooperative navigation for Multiple UAVs in Dynamic and Uncertain Environments",in proceedings of 41st IEEE conference on Decision and Control, Las Vegas, Nevada USA, pp. 2816-2822, 2002.
2. P. Pandey, A. Shukla and R. Tiwari, "Aerial navigation using Meta-Heuristics: A Survey", 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1-7, 2017.
3. B. Moses Sathyaraj, L. C. Jain, A. Finn and S. Drake, "Multiple UAVs navigation algorithms: a comparative study", Fuzzy Optim Decis Making, Vol. 7, pp. 257-267, 2008.

4. H. Ergezer, K. Leblebicioglu, “3D navigation for Multiple UAVs for Maximum Information Collection”, Journal of Intelligent & Robotic Systems, Vol. 73, pp. 737-762, 2014.
5. S. A. Bortoff, “navigation for UAVs”, in proceedings of the American Control Conference on ACC, Chicago, pp. 364-368, 2000.
6. Liang Yang, Juntong Qi, J. Xiao and Xia Yong, ”A literature review of UAV 3D navigation,” Proceeding of the 11th World Congress on Intelligent Control and Automation, 2014, pp. 2376-2381, doi: 10.1109/WCICA.2014.7053093.
7. S. Aggarwal and N. Kumar, ”navigation techniques for unmanned aerial vehicles: A review, solutions, and challenges”, Computer Communications, Vol. 149, pp. 270-299, 2020.
8. R. Smierzchalski and Z. Michalewicz, “navigation in Dynamic Environments”, in Innovations in Robot Mobility and Control, S. Patnaik (Ed.), Springer-Verlag, Berlin Heidelberg, 2005.
9. LaValle, S., “Rapidly-Exploring Random Trees: A New Tool for navigation”, Technical Report, 1998.
10. Chen X., Li G. y. and Chen X. m., ”navigation and cooperative control for multiple uavs based on consistency theory and voronoi diagram”, 29th Chinese Control and Decision Conference (CCDC), IEEE, pp. 881-886, 2017.
11. R. Kala, A. Shukla, and R. Tiwari, “Fusion of probabilistic A* algorithm and fuzzy inference system for robotic navigation,” Artificial Intelligence Review, Vol. 33, pp. 307-327, 2010.
12. Sun Q., Li M., Wang T. and Zhao C.,”Uav navigation based on improved rapidly-exploring random tree”, 2018 Chinese Control and Decision Conference (CCDC), IEEE, pp. 6420-6424, 2018.
13. V. K. Kamboj, “A novel hybrid PSO–GWO approach for unit commitment problem”, Neural Computing & Applications, Vol. 27, pp. 1643-1655, 2016.
14. S. Mirjalili, “How effective is the Grey Wolf optimizer in training multi-layer perceptrons”, Applied Intelligence, Vol. 43, pp. 150-161, 2015.
15. S. Mirjalili, S. M. Mirjalili and A. Lewis, “Grey wolf optimizer”, Advances in Engineering Software, Vol. 69, pp. 46-61, 2014.
16. Y. Wang, F. Cai, and Y. Wang, “Dynamic navigation for mobile robot based on particle swarm optimization”, AIP Conference Proceedings 1864, pp. 20-24, 2017.
17. S. Li, X. Sun and Y. Xu, “Particle swarm optimization for route planning of unmanned aerial vehicles”, in 2006 IEEE International Conference on Information Acquisition, pp. 1213-1218, 2006.
18. P. Pandey, A. Shukla and R. Tiwari, “Three-dimensional navigation for Unmanned Aerial Vehicles using Glowworm Swarm Optimization Algorithm”, International Journal of System Assurance and Engineering Management, pp. 1-17, 2017.
19. S. Asadi, V. Azimirad, A. Eslami, and A. Ghanbari, “A novel global optimal navigation and trajectory method based on adaptive Dijkstra-immune approach for mobile robot,” in Proc. 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Budapest, Hungary, pp. 1093-1098, 2011.
20. G. Varela, P. Caamano, F. Orjales, A. Deibe, F. Lopez Pena and R. J. Duro, “Autonomous UAV based search operations using constrained sampling evolutionary algorithms”, Neurocomputing Vol. 132, pp. 54-67, 2014.
21. J.C.Latombe, Robot Motion Planning, Kluwer Academic Publishers, Boston, MA, 1991.
22. Budiyanto A., Cahyadi A., Adji T.B. and Wahyunggoro O., ”Uav obstacle avoidance using potential field under dynamic environment” in IEEE International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), pp. 187-192, 2015.
23. Dai J., Wang Y., Wang C., Ying J. and Zhai J., ”Research on hierarchical potential field method of navigation for uavs”, in 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 529-535, 2018.
24. V. Roberge, M. Tarbouchi, and G. Labonte, “Comparison of parallel genetic algorithm and particle swarm optimization for realtime uav navigation,” IEEE Transactions on Industrial Informatics, Vol. 9, no. 1, pp. 132-141, 2013.
25. Hasircioglu I., Topcuoglu H.R. and Ermis M., ”3-d navigation for the navigation of unmanned aerial vehicles by using evolutionary algorithms”, in Proceedings of the 10th

Annual Conference on Genetic and Evolutionary Computation, ACM, pp. 1499-1506, 2008.
26. The Simulated Annealing Algorithm, Available: http://katrinaeg.com/simulated-annealing.html.
27. S. Zhang, “Grey wolf optimizer for unmanned combat aerial vehicle navigation”, Advances in Engineering Software, Vol. 99, pp. 121-136, 2016.
28. S. Mirjalili, A. Lewis, ”The Whale Optimization Algorithm”, Advances in Engineering Software, Vol. 95, pp. 51-67, 2016.
29. S. Mirjalili, “SCA: A Sine Cosine Algorithm for solving optimization problems”, Knowledge-Based Systems Vol. 96, pp. 120-133, 2016.