MVO-Based 2-D Path Planning Scheme for Providing Quality of Service in UAV Environment

Puneet Kumar, *Student Member, IEEE*, Sahil Garg, *Student Member, IEEE*, Amritpal Singh, *Student Member, IEEE*, Shalini Batra, *Member, IEEE*, Neeraj Kumar, *Senior Member, IEEE*, and Ilsun You, *Senior Member, IEEE*

***Abstract*—The need to develop smart unmanned aerial vehicles (UAVs) which are capable of deciding their trajectories is increasing at a rapid pace. Due to their usage in wide range of applications, such as-military, security, communications, survey mapping, disaster management, etc., the provisioning of end-to-end quality of service (QoS) is a challenging task in UAV environment. Moreover, with limited power, the efficiency of the UAVs can be enhanced if adaptive decisions with respect to their itineraries is considered dynamically. However, most of the solutions reported in the literature are not efficient with respect to QoS preservations for various applications. Motivated by this, several recently proposed meta-heuristic optimization schemes for reactive path planning of UAVs have been explored while designing a UAV path planning problem using multiverse optimizer (MVO). By carrying out the simulations over 1000 iterations, it has been demonstrated that MVO algorithm performs better in majority of the cases with average fitness function value of 0.152 and average execution time of 33.686 s.**

***Index Terms*—Meta-heuristics, multiverse optimizer (MVO), optimization, path planning, quality of service (QoS), unmanned aerial vehicles (UAVs).**

I. Introduction

UNMANNED aerial vehicles (UAVs), software controlled air-crafts, without a human pilot on board are used in various multidisciplinary areas, such as-agriculture, mission dependent activities, emergency rescue operations, surveillance, etc. [1]. All these applications require UAVs to autonomously follow a predefined path. Thus, autonomous path-planning techniques are required to provide feasible solutions, such as-fetching or delivery of data, processing real-time information, etc. within a stipulated span of time. However, challenges like physical collision, path congestion, etc. make the task of path planning more complicated [2], [3]. One of the major requirements is to represent the path by a sequence of waypoints in such a manner that UAVs can dynamically adjust themselves to avoid collisions with other UAVs [4], [5]. Traditional path planning approaches, such as-*visibility graphs*, *potential fields*, and *cell decomposition* are not efficient as they are proactive in nature and suffer from problems which include high time complexity and local minima trapping [6], [7]. A viable solution for the development of path planning models is possible through the indulgence of reactive approaches as these are more productive in comparison to proactive approaches [8].

In recent years, several optimization schemes, such as-A* search algorithm, linear programming, evolutionary algorithm, random trees, genetic algorithm (GA), particle swarm optimization (PSO), etc., have been used for UAV path planning and trajectory generation [9]. However, most of these are gradient-based approaches, which tend to compute the exact solution for path planning. Although, these algorithms are capable of extracting the information about the shape and behavior of a function used for path planning of UAVs, they may get trapped in the local optima. To solve these problems, gradient free (stochastic) meta-heuristic techniques are used widely. A meta-heuristic is a high-level generic algorithmic scheme which serves as a blueprint for heuristic optimization algorithms. Most of the meta-heuristics do not require the computation of gradients for search spaces. Since they solve the problems stochastically, there are less chances of getting trapped in the local optima and hence can be used to obtain the global best solution much faster.

In this context, YongBo *et al.* [8] proposed modified wolf pack search (WPS) algorithm for UAVs path planning. In a similar work, Zhang *et al.* [10] used grey wolf optimizer (GWO) for path planning in combat field. Phung *et al.* [11] formulated the path planning problem for UAV vision-based surface inspection by enhancing the discrete PSO technique. Here, the proposed approach was used to solve the traveling salesman problem by taking both the coverage and obstacle avoidance into account. Similarly, Chen *et al.* [12]

Manuscript received September 1, 2017; revised November 26, 2017; accepted December 9, 2017. Date of publication January 23, 2018; date of current version June 8, 2018. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea Government (MSIT) (No.2017-0-00664, Rule Specification-based Misbehavior Detection for IoT-Embedded Cyber Physical Systems) as well as the Soonchunhyang University Research Fund. *(Corresponding author: Ilsun You.)*

P. Kumar, S. Garg, A. Singh, S. Batra, and N. Kumar are with the Computer Science and Engineering Department, Thapar Institute of Engineering and Technology (Deemed University), Patiala 147004, India (e-mail: puneet9983@gmail.com; garg.sahil1990@gmail.com; amritpal.singh203@gmail.com; sbatra@thapar.edu; neeraj.kumar@thapar.edu).

I. You is with the Department of Information Security Engineering, Soonchunhyang University, Asan-si 31538, South Korea (e-mail: ilsunu@gmail.com).

Digital Object Identifier 10.1109/JIOT.2018.2796243

TABLE I
COMPARISON OF RECENTLY PROPOSED STOCHASTIC-BASED META-HEURISTIC ALGORITHMS

Algorithm	Inspiration	Parameters	Solution	Optimization Steps		
				Exploration	Exploitation	Other (if any)
ALO [13]	Hunting behaviour of antlions.	d: Dimensions n, m: No. of ants and antlions $\vec{A}_{i,j}$, $AL_{i,j}$: Position of ant and antlion	Bringing the insect to the location of the pit.	Through random selection of antlions and random walks of ants around them.	By continuous shrinking boundaries of the traps around the ants.	Gradual decrease in ant movements with iterations is equivalent to convergence.
DA [14]	Swarming behavior of dragonflies.	N: Number of neighbouring individuals X: Position of current individual X_j: Position of j^{th} neighbour A_i, C_i: Alignment, Cohesion	Finding Prey's location.	Hunting behavior of the dragonflies.	Navigation movements of the flies.	'Separation', 'Alignment'and 'Cohesion'- modeled for collision avoidance, velocity and central tandency.
GWO [15]	Living hierarchy and hunting strategy of grey wolves.	$\vec{X}$, $\vec{X_p}$: Position vector of wolf and prey $\alpha, \beta, \gamma, \omega$: Hierarchy of wolves $\vec{A}$, $\vec{C}$: Coefficient vectors	Final shrunken circle indicating prey's position.	The wolves diverge from each other to search for prey. And converge back on finding it.	Wolves encircle the prey, and attack after skrinking the circle.	—
MFO [16]	Navigation methods of moths.	n, m: No. of moths and flames D: Distance between moth and flame d: No. of variables	Reaching final destination by moving in the transverse orientation.	Moth finds the next available position in its orientation space with the flame.	After finding the next position, moth navigates towards it.	—
WOA [17]	Prey hunting nature of whales.	$\vec{X}$: Position vector $\vec{r}$: Random vector $\vec{A}$, $\vec{C}$: Coefficient vectors	Final shrunken circle indicating prey's position.	A random vector is initialized and, directed towards prey's location iteration by iteration.	Whales form a bubble net around prey's location and attack it.	Before exploration, whales recognizes the location of the prey and, encircle it.
MVO [18]	Multi verse theory of Physics.	n, d: No. of universes and dimensions NI: Normalized inflation rate WEP: Wormhole existence probability TDR: traveling distance rate	The path with highest inflation rate.	By sorting the universes based on inflation rate through black holes and white holes.	By the exchange of objects of the universes through wormholes.	—

ALO: Ant Lion Optimizer; DA: Draganfly Algorithm; GWO: Grey Wolf Optimizer; MFO: Moth Flame Algorithm; WOA: Whale Optimization Algorithm; MVO: Multi Verse Optimizer

introduced a modified central force optimization, which is based on the metaphor of gravitational kinematics, to solve 3-D UAV path planning problem. In this process, PSO algorithm and GA were used to improve the original central force optimization method. The convergence analysis by linear difference equation method showed that their optimization technique gives the effective path planning result when applied on quadrotor helicopter. In an another work, YongBo *et al.* [8] proposed the modified WPS algorithm to solve the path planning problem of UAVs. In their approach, crossover and mutation operators of GA are applied to realize the WPS algorithm. Moreover, in order to plan the suitable path, the authors deployed the path smoothing process of the cubic B-spline curve.

Although several meta-heuristics exist in literature, but their usage for UAVs path planning has not been explored to their full potentials. This was an inspiration to choose several stochastic-based meta-heuristics for UAVs path planning. A comparison of some recently proposed meta-heuristics is given in Table I.

A. Contributions

This paper aims to explore and compare the existing meta-heuristic schemes for UAV path planning. On the basis of this performance comparison, the multiverse optimizer (MVO) algorithm is selected to deduce UAV's path with high quality of service (QoS) assurance. The main contributions are described as follows.

1) First, existing meta-heuristic techniques for UAV path planning are statistically compared on several benchmark functions, such as-Rastrigin function, Griewank function, Weierstrass function, and Ackley function to identify the best one.
2) Second, path planning problem for deciding the trajectory of UAV with minimum collision and least deviation has been formulated mathematically by using the near optimal technique identified in the first step, i.e., MVO.
3) Finally, a comparison with the other existing proposals is presented to validate the proposed path planning scheme.

B. Organization

The outline of this paper is as follows. The problem formulation is discussed in Section II. Then the goodness value of considered meta-heuristics is computed in Section III. Section IV presents the proposed MVO-based path planning scheme for UAVs. In Section V, the performance of the proposed model using simulations is evaluated. Finally, the conclusions are drawn in Section VI.

II. PROBLEM FORMULATION

Consider a geographical area, g_i, with k UAVs ($S_{\text{UAV}} = \{u_1, u_2, \ldots, u_k\}$), where each UAV performs the assigned task (which may include serving user demands, arriving at the given target on-time, etc.) over the defined area and then moves to the destination d_j, such that, $d_j \in g_i$. UAVs are supposed to choose the itinerary under the following constraints.

1) Number of collisions (col) between UAVs, operating within the defined network range should be minimum.
2) Shortest path (sp) should be used to accomplish the task such that total travel time required is minimized.
3) Deviation (δ) of UAVs from final destination should be minimum, i.e., it should reach the destination or should be within the vicinity of the defined destination.
4) The performance (pm) of assigned task, i.e., urban delivery, serving user demands, etc. should be maximized.

Thus, the mathematical model for communication among different entities in the proposed solution can be defined as

$$\underset{\max(\text{pm})}{\min(\text{col,sp},\delta)} \{g_i \rightarrow [S_{\text{UAV}}]\} \rightarrow \text{UAV}_i^{\text{pm}} \mapsto \begin{cases} \overset{S}{\mapsto} \text{Destination}^{\text{sp},\delta} \\ \overset{C}{\rightleftharpoons} \text{UAV}_j^{\text{col}} \end{cases}$$

$$\begin{cases} \text{where:} \\ ((S = \text{Services}) \text{ and } (C = \text{communication})) \\ \delta = (\text{Position Reached} - \text{destination}) \\ \text{sp} = (\text{Collision free path}) \times (\text{Minimum time}) \\ \text{col} = (C_{\text{UAV}_i} - C_{\text{UAV}_j}) < 2R \\ (\text{where } R \text{ is radius of UAV and } C_{\text{UAV}_i} \text{ is center of UAV}). \end{cases} \tag{1}$$

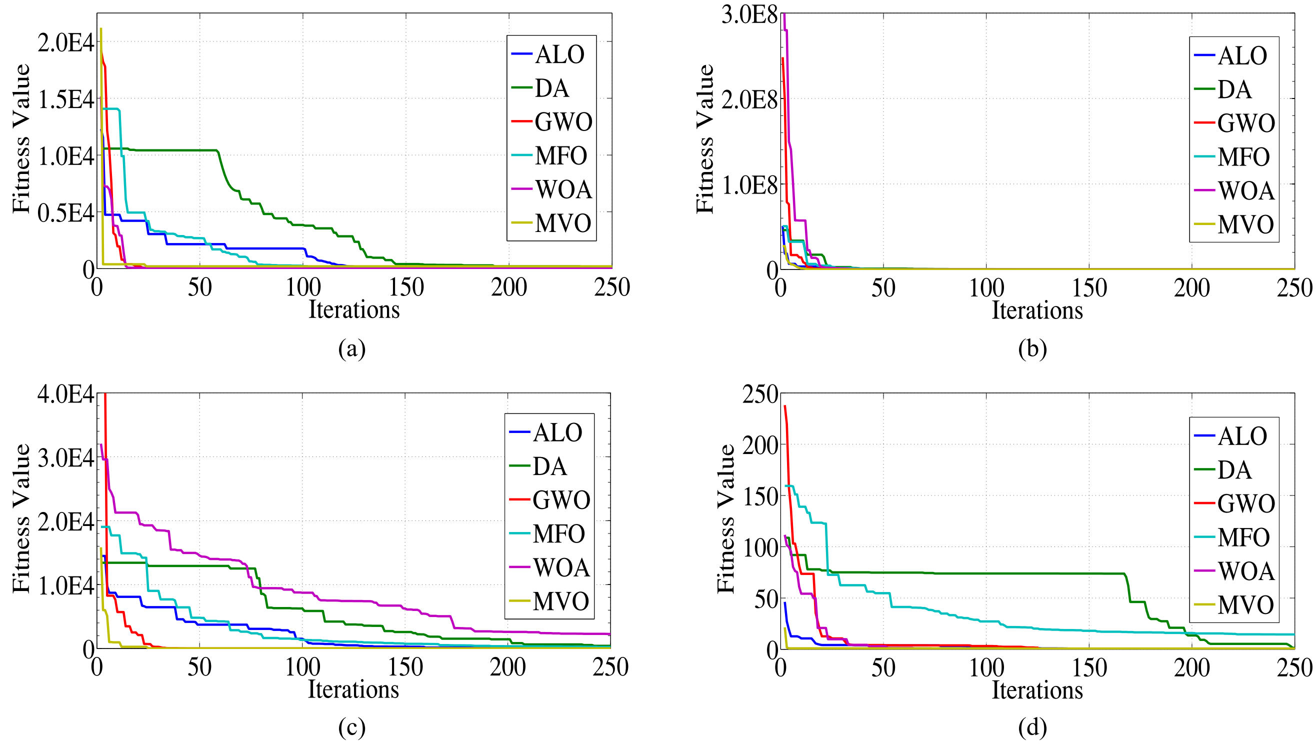


Fig. 1. Convergence plots wit respect to different algorithms. (a) Rastrigin function. (b) Griewank function. (c) Weierstrass function. (d) Ackley function.

III. Algorithms Selection

Meta-heuristic techniques aim to find an approximate solution to the problem. According to no free lunch theorem [19], single meta-heuristic method will not work for all optimization problems. Hence, this paper considers six meta-heuristic optimization techniques namely ant lion optimizer (ALO), dragonfly algorithm (DA), GWO, moth flame optimization (MFO), whale optimization algorithm (WOA), and MVO to find the most suitable one. Here, two evaluation metrics, i.e., convergence rate and number of function evaluations (NFEs) are employed to gauge the goodness of the optimization methods under consideration.

1) Convergence Rate: In [20], convergence rate has been considered as an indicator to estimate the efficiency of an optimization scheme. Since this paper tries to solve the path planning problem using optimization, it has used convergence rate as a parameter to select the most suitable technique for this problem.

Definition 1 (Convergence Rate): Convergence rate in an optimization process is the speed at which the fitness function (objective function) approaches its optimal value.

The considered optimization algorithms have been evaluated to monitor their rate of convergence toward the optimum value of the fitness function. For this purpose, four benchmark fitness functions of different mathematical standards have been used, i.e., Rastrigin function, Griewank function, Weierstrass function, and Ackley function [21]. Each of the aforementioned algorithms are compared with respect to the convergence rate by varying the number of dimensions along with number of search agents. The results obtained are illustrated in Fig. 1. It can be seen that MVO converges faster as compared to its counterparts (ALO, DA, GWO, MFO, and WOA) in most of the cases. This was the motivation to use MVO for UAVs path planning.

2) Number of Function Evaluations: NFEs has been used as a non deterministic method of complexity analysis of the optimization method with sizeable number of stochastic parameters by Kiran *et al.* [22] and Sadollaha *et al.* [23].

Definition 2 (Number of Function Evaluations): NFE for an algorithm denotes the number of times the fitness function gets called until the optimal solution is found. Smaller number of NFE denotes lesser computational effort and hence more efficiency of the optimization method.

Table II represents the comparative analysis of MVO with the other optimization algorithms for various benchmark functions. A total of 1000 iterations have been performed for each run and number of fitness function executions have been recorded. Here, *acceleration measure* is defined as the ratio of NFE of MVO to NFE of other algorithms. Larger value of this measure denotes more relative speed. Values achieved in Table II clearly indicate that MVO makes lesser number of calls to the fitness function as compared to other algorithms in reaching its optimal value leading to the best acceleration measure.

IV. Mathematical Model for Proposed Solution

This section proposes a solution for path planning using MVO. According to MVO, search-space of UAVs can be considered as a set of universes, where the concepts of white hole and black hole are utilized to explore and select the optimal paths for UAVs. On the contrary, wormholes are used to map the non-UAV objects (birds, air-crafts, etc.) present in search

TABLE II
NUMBER OF FUNCTION EVALUATION

Fun	Dim	Number of Function Evaluations					
		ALO	**DA**	**GWO**	**MFO**	**WOA**	**MVO**
F1	30	9557760	12131320	6731920	5268560	4356840	3306560
F2	30	13605648	11074686	8473668	7398132	4839702	3676848
F3	30	10272350	4063910	8164946	6082508	4901620	3560524
F4	30	8730070	7786691	9891208	6290994	3942183	4117529
F5	30	9463230	7215985	8433320	6708380	4007360	3782765
F6	30	9801220	6467760	6854640	6856850	3932075	3358843
Total		61430278	48740352	48549702	38605424	25979782	21803068
R		0.3549	0.4473	0.4490	0.5647	0.8392	1

R: Acceleration Measure; F_1: Sphere Function; F_2: Rosenbrock Function; F_3: Step Function; F_4: Rastrigin Function; F_5: Ackley Function; F_6: Griewank Function.

space. To apply MVO on UAV-based network, objectives are modeled in terms of MVO as follows:

$$(\text{Universe} \mapsto S_{\text{UAV}}) \begin{cases} \text{Explore}\begin{cases}\text{White hole}\begin{cases}\text{Explore all paths}\\ \text{Collision identification}\end{cases}\end{cases} \\ \text{Exploit}\begin{cases}\text{Black hole}\begin{cases}\text{Deviation in final postion}(\delta)\\ \text{Shortestpath(sp)}\\ \text{Collision Avoidance}\end{cases}\\ \text{Wormhole}\begin{cases}\text{Non} - \text{UAV obstacles.}\end{cases}\end{cases}\end{cases} \tag{2}$$

The potential path for the UAV network is analogous to a universe in the MVO. The inflation rate of the universe is equivalent to the corresponding value of the fitness function. The implementation uses original rules of MVO with the following assumptions.

1) Each UAV has a “moving rate” similar to the “expansion rate” of a universe and the network of UAVs has an overall expansion rate similar to the network of universes.
2) A UAV path is one of the possible combinations of the subpaths. Different UAVs may exchange the subpaths among them just as the universes may exchange the objects among them.
3) Solutions with high moving rate are likely to have longer traversal paths whereas, the ones with low rate are likely to terminate their search near to the destination point.
4) Number of threats in all the solutions may be arbitrary and independent of the moving rate of the UAV.

A. Adopted MVO Algorithm for UAV Path Planning

The algorithm starts by exploring a set of all the random paths for the UAVs in the defined domain and tries to reach the optimal path by combining, moving or evolving the potential solutions according to the defined constraints. Following steps are used in this process.

1) Explore the Universe: All associated paths with source–destination pairs for each UAV are explored and the possibilities of collisions with other UAVs are considered.

1) Initially, all paths from source to destination (s_i, d_i), i.e., UP_s^d are created for the ith UAV, wherein s_i is its current position and (d_i) indicates its destination.
2) All UAVs share their path information to select the best universe, i.e., the shortest collision-free path for all UAVs.
3) It is assumed that initial and destination positions of the UAV_i are (x_i, y_i) and (x_{d_i}, y_{d_i}), respectively, where obstacles are located at (t_{x_i}, t_{y_i}) such that $i = \{1, 2, \ldots, N^t\}$. Here, optimal path for the UAV is subject to a set of constraints, $g(x^j, y^j) = 0$. This involves the cost components related to path length, obstacle cost, and end point’s deviation from the destination.
   a) An important factor for generating an optimal path between source and destination is the speed with which UAV moves. It is defined as UAV moving rate (UMR) and is defined as follows:
   $$\text{UMR}_{\text{UAV}_i} = \overset{k}{\min}\left(\text{UP}_{s_i}^{d_i}\right) = \left(\forall j \in \text{UP}_{s_i}^{d_i}\right)(\text{Dist}(j, d_i))$$
   $$\{(\text{where Dist() is distance between two points}). \tag{3}$$
   b) A threat could be another UAV or some other non-UAV obstacle which includes flying objects like birds, air-crafts, etc. For a given path (p_i), probability of collision by a non-UAV object in each path is defined as the wormhole existence probability (WEP). If UCP is the collision probability of any UAV, then overall threat probability (OTP) for the UAV is given as follows:
   $$\text{OTP}_{p_i} = (\text{UCP})x\left(\text{WEP}_{p_i}\right)$$
   $$\begin{cases}\text{where:}\\ \text{WEP} = Pr[\text{Non} - \text{UAV obstacles}]\\ \text{UCP} = F\left(\left(\frac{\text{Number of UAVs on } p_i}{\text{Maximum limit}}\right), t\right)\\ (\text{here UCP is function of time}).\end{cases} \tag{4}$$
4) According to the above stated metrics for path selection, the function for white hole is defined as
$$F_{\text{WH}}^{\text{UAV}_i} = (\forall j \in S_{\text{UAV}}) \frac{\#(\text{CPS}(i, j))\left(\text{TC}_{p_i}(p_i \in \text{CPS})\right)}{\text{Dist}_{\text{UAV}_i}^{\text{UAV}_j}(t)}$$
$$\begin{cases}\text{where:}\\ \text{CPS}(i, j) = \text{UMR}_{\text{UAV}_i} \cap \text{UMR}_{\text{UAV}_j}\\ \text{Dist}_{\text{UAV}_i}^{\text{UAV}_j}(t) = \left(C_{\text{UAV}_i} - C_{\text{UAV}_j}\right)^t < 2R\\ (\text{here CPS refers to Common Path Set and}\\ \text{UMR is UAV Moving Rate}).\end{cases} \tag{5}$$

The UAVs that either have a overlapping path at a particular time or, the UAVs whose separation is less than the distance between their centers, are likely to collide. White hole function will list down all such paths here. In further steps, the potential collisions are detected and resolved.

5) *Collision Detection:* Using white hole function F_{WH}, for each UAV, at a given time t, a set of collision path (PC) is maintained for each UAV as follows:

$$(\forall i | i \in S_{\mathrm{UAV}})\mathrm{PC}_i \leftarrow \left[(\forall t_k | t_k \in T_{\mathrm{UAV}})\mathrm{CPS}((i,j),t_k)\right]$$
$$\begin{cases}\text{where:}\\ \mathrm{CPS}((i,j),t_k) = \left(\mathrm{UMR}^{t_k}_{\mathrm{UAV}_i} \cap \left[(\forall j | j \in S_{\mathrm{UAV}})\mathrm{UMR}^{t_k}_{\mathrm{UAV}_j}\right)\right]\\ \text{here } (i \neq j),\ (T_{\mathrm{UAV}} = \text{Total Time}).\end{cases} \tag{6}$$

6) The priority (ImP) is set on the basis of flight time and number of collision paths. The UAV having lesser flight time and more collision paths has higher priority. The path modification process for collision avoidance is carried on first for the UAV with the highest priority

$$(\forall i | i \in S_{\mathrm{UAV}})\mathrm{ImP}_i \leftarrow \frac{\#(\mathrm{PC}_i)}{\left(t^i_c\right)(T_{\mathrm{UAV}})}$$
$$\begin{cases}\text{where:}\\ (t^i_c = \text{Time taken by UAV}_i \text{ to reach}\\ \quad \text{collision point})\\ (T_{\mathrm{UAV}} = \text{Total flight time of UAV}_i).\end{cases} \tag{7}$$

2) Exploit the Universe: Optimal collision free path is decided by following the constraints that the path identified should be the shortest possible and the deviation between final position of the UAV to its destination point should be minimum.

1) Based on the priority of UAVs ($\mathrm{ImP}_{\mathrm{UAV}}$) assigned by (7) in white hole function, paths of the UAVs are readjusted to avoid collisions.
2) The UAV_i with $\mathrm{Max}(\mathrm{ImP}_i)$ is first scanned for all the alternative routes seeking the shortest path. Accordingly, the new path is communicated to all the UAVs using (3)–(6).
3) According to the above stated metrics for path reselection, the function for black hole is defined below. It exploits the best path after filtering all the listed paths on the basis of the length of the path and the deviation between the actual destination and the end point of the path

$$\begin{cases}F^{\mathrm{UAV}_i}_{\mathrm{BH}} = F^{\mathrm{UAV}_i}_{\mathrm{WH}}(\mathrm{sp})(\delta)\\ F^{\mathrm{UAV}_i}_{\mathrm{BH}} = (\forall j \in S_{\mathrm{UAV}})\dfrac{\#(\mathrm{CPS}(i,j))\left(\mathrm{TC}_{p_i}(p_i \in \mathrm{CPS})\right)(\mathrm{sp})(\delta)}{\left(\mathrm{Dist}^{\mathrm{UAV}_j}_{\mathrm{UAV}_i}(t)\right)}\end{cases}$$
$$\begin{cases}\text{where:}\\ \mathrm{sp} = \min\left(\mathrm{UP}^{d_i}_{s_i}\right) \ \&\ \mathrm{UP}^{d_i}_{s_i} = \left(\mathrm{UP}^{d_i}_{s_i} - \mathrm{sp}\right)\\ \delta = \min\left(d^{\mathrm{UAV}_i}_i - fp^{\mathrm{UAV}_i}\right)\\ (\text{here } fp = \text{final position}).\end{cases} \tag{8}$$

4) Based on the determined alternative paths, a collision free stable network of the UAVs is established such that the stated performance metrics can be achieved optimally.

3) Non-UAV Collision Avoidance: After selecting the optimal paths for all the UAVs using (8), non-UAV objects are randomly placed in the universe of the UAVs. If the paths which have been calculated earlier overlap with the position of the non-UAV objects, wormhole function is called which recalculates the paths for all the UAVs

$$\begin{cases}\mathrm{NU}_r \leftarrow \mathrm{Random}(x,y)^{g_i}\\ \forall i | \mathrm{UAV}_i(\mathrm{CPS}(\mathrm{UAV}_i, \mathrm{NU}_r))\\ \mathrm{if}(\mathrm{CPS}(\mathrm{UAV}_i, \mathrm{NU}_r) == \phi)\begin{cases}\text{TRUE} & \text{No Exchange}\\ \text{FALSE} & F^{\mathrm{UAV}_i}_{\mathrm{WO}}\end{cases}\\ F^{\mathrm{UAV}_i}_{\mathrm{WO}} = \begin{cases}\text{Calculate } F^{\mathrm{UAV}_i}_{\mathrm{WH}}\ (3)\\ \text{Calculate } \mathrm{PC}_i\ (4)\\ \text{Calculate } \forall(i,j \in \mathrm{UAV}_s)\mathrm{CPS}(i,j)\ (4)\\ \text{Update } F^{\mathrm{UAV}_i}_{\mathrm{BH}}\ (6)\end{cases}\end{cases}$$
$$\begin{cases}\text{where:}\\ \mathrm{NU}_r = \text{Random Non} - \text{UAVobject}\\ F^{\mathrm{UAV}_i}_{\mathrm{WO}} = \text{Function for Wormhole}\end{cases} \tag{9}$$

Path recalculation is done dynamically and the same is communicated to all the UAVs present in the environment. It improves the moving rate of the UAVs, hence the possibility of their collisions reduces and their stability improves. The proposed solution ultimately achieves a collision free existence of all the UAVs in the presence of other UAVs and non-UAV objects.

The adopted MVO algorithm for UAV path planning is outlined in Algorithm 1. First, the algorithm parameters are defined, i.e., universe of UAVs, moving rate, UAV paths, colliding paths, priority, threat cost, etc. In lines 2–7, white hole functions is called for each UAV, which lists all possible paths for that UAV. In lines 10–13, collision is detected for every path listed above. And then, a collision free path is picked for each UAV by calling the black hole function. In lines 18–21, a random non-UAV object is placed in the universe of the UAVs. The algorithm from lines 22–26 checks whether the non-UAV object placed above collides with the UAV paths already determined in earlier steps. It is checked by analyzing the collision path set of the non-UAV object with each UAV. If the collision paths do exist then wormhole function is called, which in turn calls the white hole function to list UAV paths again and then calls the black hole function to select a collision free path for each UAV in the presence of non-UAV objects.

B. Fitness Function

In this section, the fitness function (J) is defined to minimize the cost of a given UAV path. The cost depends on the number

Algorithm 1 MVO Algorithm for UAV Path Planning

Define Universe of UAVs (S_{UAV}) from (0,0) to (X,Y)
Define Normalized Moving Rate of UAVs (*NMR*)
Define Moving Rate of the UAVs (*UMR*)
Define Priority of UAV (*ImP*)
Define Overall Threat Probability (*OTP*)
Define Possible paths for a UAV (*PS*)
Define Set of Colliding Path Set (*CPS*)
Function: CalcPath()

```
Step 1: Exploration Phase
for i in S_UAV do
    Update UMR_i
    Calculate OTP_i using Eq. (4)
    Calculate ImP_i using Eq. (7)
    PS_i ← F_WH^{UAV_i} using Eq. (5)
    /* List all possible paths for each UAV */
end for
Step 2: Exploitation Phase
for j in PS_i do
    Calculate CPS_j using Eq. (6)
    if (CPS_j == ϕ) then
        Calculate F_BH^{UAV_i} using Eq. (8)
        /* Decide a collision free path */
    end if
end for
Step 3: Non UAV Collision Avoidance
for each Non UAV object NU_k do
    x_k=random([0,X]);
    y_k=random([0,Y]);
    place NU_k at location (x,y)
    for l in S_UAV do
        Calculate CPS(UAV_l, NU_j)
        /* Check collision path set with each UAV */
        if (CPS(UAV_l, NU_j) ≠ ϕ) then
            Calculate F_WO^{UAV_i} using Eq. (9)
            /* Call the wormhole function if paths collide. */
        end if
    end for
end for
```

of UAVs, intersection among their paths, length of the path and the deviation between actual destination and the end point of the path

$$J = \frac{w_1(\#\text{UAV})^2 \times w_2(\text{sp}) \times w_3(\delta)}{w_4\left(\text{Dist}_{\text{UAV}_i}^{\text{UAV}_j}(t)\right)}. \tag{10}$$

Here, #UAV*s* denotes number of other UAVs and non-UAV obstacles, sp is the shortest possible path identified, CPS and TC are assumed to be proportional to the number of UAVs and $w_i \in \mathcal{R}^N$ is a weight vector relating to each cost component of the function to be determined experimentally.

V. Performance Evaluation

This section demonstrates the performance of MVO in comparison to other meta-heuristic optimization algorithms (ALO, DA, GWO, MFO, and WOA) on the basis of statistical distribution values of the fitness function and the computation time of the optimization methods. All the experiments have been performed in MATLAB R2014a installed on Intel Core i5-7200U, 2.70 GHz, 8 GB RAM machine with 64 bit Windows 10 OS.

A. Simulation Setup

Here, six different use-cases have been considered to evaluate the fitness function given in (10). To implement this scenario, experimental setup is given in Fig. 2. It has been assumed that all weight vectors have equal influence. In case of potential collision, the UAV turns by one grid and continues traversing parallel to its initial path.

TABLE III
Experimental Setup

c_1	c_2	c_3	c_4	c_5	c_6
1	4.0	2	0.50	10	2.0178
	4.0	2	0.50	15	2.1237
	4.0	2	0.50	20	1.9371
	4.0	2	0.50	30	1.9023
2	3.0	3	0.50	10	8.8745
	3.0	3	0.50	15	8.7241
	3.0	3	0.50	20	8.2412
	3.0	3	0.50	30	8.0918
3	2.83	3	1.00	10	11.3137
	2.83	3	1.00	15	11.9209
	2.83	3	1.00	20	11.2324
	2.83	3	1.00	30	11.1265
4	4.65	2	1.5	10	3.3715
	4.65	2	1.5	15	3.0115
	4.65	2	1.5	20	2.9213
	4.65	2	1.5	30	2.8016
5	2.15	3	0.67	10	12.4009
	2.15	3	0.67	15	12.4116
	2.15	3	0.67	20	12.2020
	2.15	3	0.67	30	12.1012
6	4.0	3	1.0	10	9.3567
	4.0	3	1.0	15	9.1216
	4.0	3	1.0	20	8.9817
	4.0	3	1.0	30	8.1815

c_1: Case; c_2: Dist(t_0); c_3: #UAVs; c_4: δ; c_5: Dim; c_6: J.

Table III represents the simulation set up and fitness value calculation for the use cases shown in Fig. 2. Here Dist (t_0) is the initial distance between a particular UAV and the UAV with adjoining priority, *Dim* denotes the population size for the optimization method, *J* represents an average of the fitness function values for all the UAVs. As shown in Fig. 2, ImP_i, d_i, and f_i denote priority, destination point and end point of UAV's path, respectively. Moreover, non-UAV objects are denoted by "•." The fitness function values are calculated as per (10) and the trajectories for UAVs are determined according to the rules defined in Section IV-A.

In this set-up, it has been assumed that all the UAVs would start at the same time and travel at the same speed. In case of potential collision, the UAV with highest priority will turn by one grid. Thus, it would not arrive at the point of potential collision at the same time as other UAVs, and the collision will be avoided.

Following is the description of how the necessary parameters to calculate the fitness value are determined and final UAV paths are planned for each simulation case. The variable $\text{Dist}_{\text{UAV}_i}^{\text{UAV}_j}(t)$ denotes the Euclidean distance between UAV_i and UAV_j at time t. This distance is calculated at start time t_0. Here, sp is the shortest collision free path for a particular UAV, which is calculated after making all the adjustments in the planned paths. Finally, the fitness values are calculated by plugging these parameters in (9).

1) *Case I:* There are two UAVs and no non-UAV objects. Both the UAVs have equal priorities. Hence, any one of the UAVs would modify its path to avoid the potential collision. The path of UAV_1 has been changed by one grid and then it continued moving parallel to its original path. Hence, it arrived one grid away from its destination

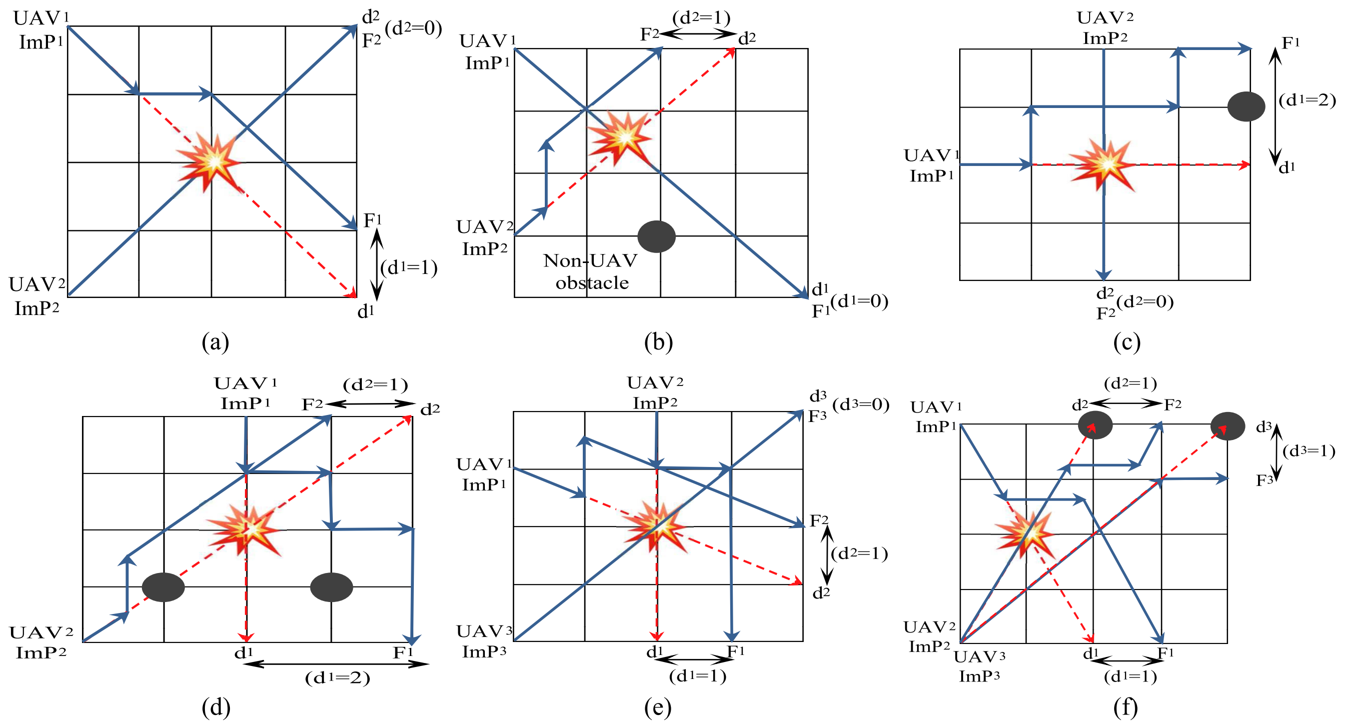


Fig. 2. Different cases for path simulations. (a) Case I: $\mathrm{ImP}_1 = \mathrm{ImP}_2$. (b) Case II: $\mathrm{ImP}_1 < \mathrm{ImP}_2$. (c) Case III: $\mathrm{ImP}_1 = \mathrm{ImP}_2$. (d) Case IV: $\mathrm{ImP}_1 > \mathrm{ImP}_2$. (e) Case V: $\mathrm{ImP}_1 > \mathrm{ImP}_2 > \mathrm{ImP}_3$. (f) Case VI: $\mathrm{ImP}_1 = \mathrm{ImP}_2 > \mathrm{ImP}_3$.

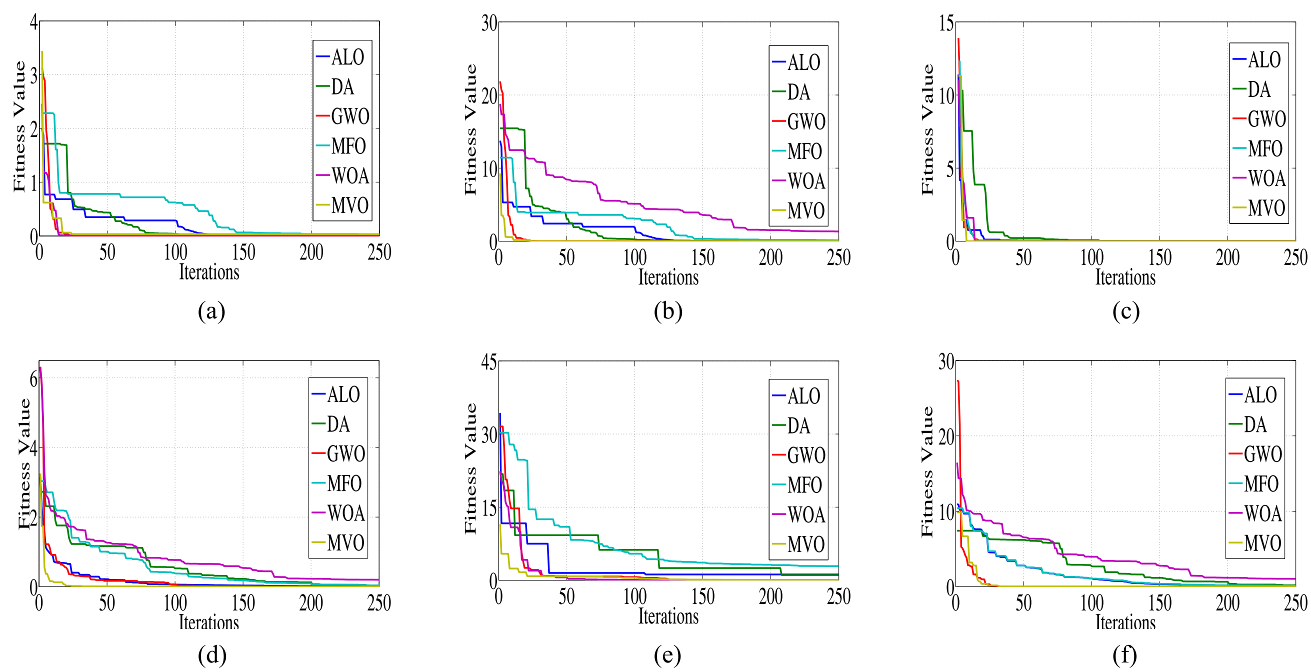


Fig. 3. Convergence plots for the fitness function. (a) Case I. (b) Case II. (c) Case III. (d) Case IV. (e) Case V. (f) Case VI.

position. While the final position UAV_2 was same as its destination.

2) *Case II:* There are two UAVs and one non-UAV object. UAV_2 has shorter flight time and higher priority according to (7), hence it adjusts its path to avoid collision with UAV_1. Placing non-UAV object at random coordinates did not collide with the calculated paths of any of the UAVs. So, path recalculation is not initiated as per the rules described in Section IV-A. Deviations are calculated accordingly for both the UAVs.

3) *Case III:* Both the UAVs present in the UAV space have equal flight time and collision path set, hence they have equal priority. One of the UAVs, UAV_1 adjusted its path to avoid the collision with the other UAV. Then a non-UAV object was placed randomly, which would collide with the expected path of UAV_1. Hence, the path of

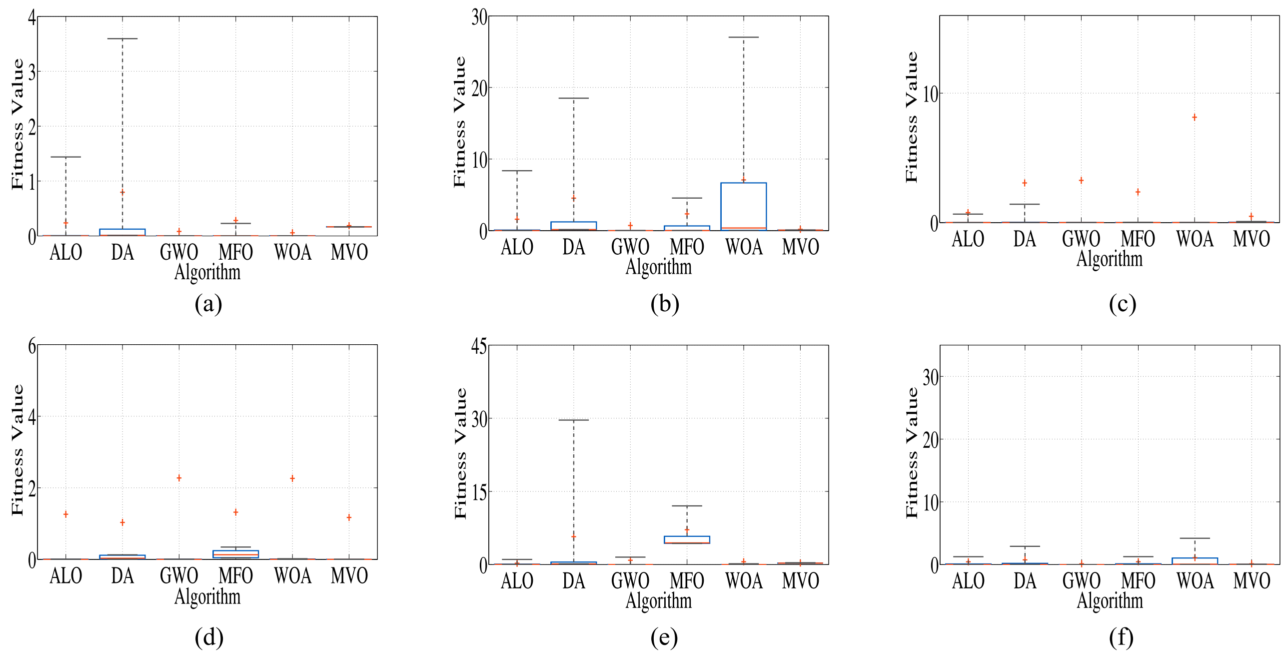


Fig. 4. Box plots for the fitness values of simulation cases. (a) Case I. (b) Case II. (c) Case III. (d) Case IV. (e) Case V. (f) Case VI.

UAV_1 is further adjusted and it reaches to its destination position with a deviation of two grids.

4) *Case IV:* A total of four objects are present in the considered UAV area—two UAVs and two non-UAV objects. UAV_1 has higher priority based on the flight time. It adjusts its path by turning one grid before the expected collision. Then two non-UAV objects are placed which hinder the paths of both the UAVs. Hence, path calculation is initiated for both the UAVs by calling wormhole function described in (9). Final paths and respective deviations for the UAVs are calculated as shown in Fig. 2.

5) *Case V:* This scenario contains three non-UAVs with different planned path lengths, originally planned to cross the center of the UAV space at the same time. Because of the shortest planned path, UAV_1 has the highest priority, followed by UAV_2 and then UAV_3. First UAV_1 adjusts its path to avoid collision and then UAV_2 does so. They reach their final positions with one grid deviation for each. UAV_3 does not require any path adjustment, so its final position is same as the planned destination.

6) *Case VI:* There are three UAVs and two non-UAV objects. UAV_1 and UAV_2 has same flight time and hence, same priority. UAV_3 has more flight time, hence lesser priority. UAV_1 makes the necessary adjustments to its path first. And, after placing the non-UAV objects, UAV_2 and UAV_3 have to replan their paths. Thus, collision free universe of the UAVs is finally achieved.

B. Comparative Evaluation

This section discusses the relative evaluation results of the proposed path planning scheme in comparison with the different optimization techniques under consideration.

1) Convergence Plots: Convergence rate denotes the speed at which the value of an optimization function approaches its optimal value. In this analysis, the convergence rate is analyzed using the proposed fitness function. The results are shown in Fig. 3 for six simulation cases. In cases II–IV, MVO has the fastest rate to approach its optimal fitness value through out the iterations. In case I, GWO and MOA outperform MVO for initial 25 iterations to attain their optimal value. But soon after that, MVO converges at a much faster rate and reaches its optimal fitness. In case V, WOA outperforms MVO between iteration 50 and iteration 125 whereas in case VI, GWO approaches to its optimal value at a faster rate as compared to initial value of MVO. However, in cases V and VI, MVO attains its best fitness function value sooner than other algorithms. It can be seen that MVO achieves the best value of the fitness function in lesser number of iterations indicating a better performance in comparison to the existing proposals.

2) Box Plots: Box and whisker diagrams in Fig. 4 represent the significance of the fitness function values obtained in all six use-cases. Plots for MVO is narrower and lower than the plots of other algorithms in most of the cases. The median fitness value for MVO is equal to or below the minima value of ALO, DA and MFO in all the cases. WOA shows competitive fitness values as MVO in cases I and III–V but in cases II and VI, MVO outperforms it significantly. Similarly, GWO has equivalent fitness values to MVO in 4 out of the 6 cases but in rest of the cases, it performed much worse. MVO depicted consistency in converging toward fitness value in all the cases. It has the least average fitness function value (0.152). Thus, the statistical results suggest the superiority of MVO compared to the other algorithms.

3) Computation Time: In this analysis, average execution time for aforementioned simulation cases is analyzed. Fig. 5 represents this analysis for the six simulation cases.

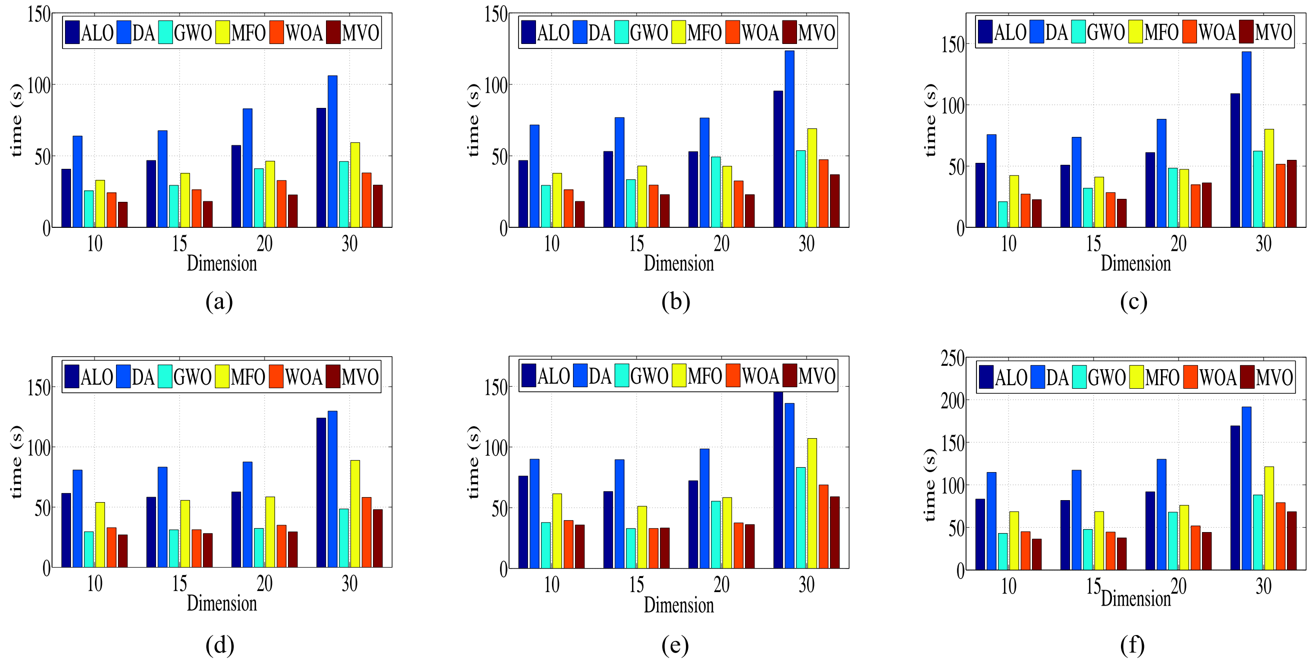


Fig. 5. Average execution time for the considered simulation cases. (a) Case I. (b) Case II. (c) Case III. (d) Case IV. (e) Case V. (f) Case VI.

In cases I, II, IV, and VI, MVO executed faster than all the algorithms for all the dimensions considered. In case III, for dimensions = 20 and 30, WOA outperformed MOV, but for all the other subcases with different dimensions, MVO came out to perform better (34.111 s) compared to WOA (35.428 s). Similarly, in case V, for dimensions = 15, WOA showed competitive results to MOV but average time for all the subcases was better for MVO. Overall, MVO had the least average computation time of 33.686 s in comparison to other algorithms (76.935 s for ALO, 99.929 s for DA, 44.518 s for GWO, 60.375 s for MFO, and 39.794 s for WOA), and hence it is computationally efficient for UAVs path selection.

VI. Conclusion

In this paper, MVO algorithm has been employed for UAV path planning while provisioning end-to-end QoS. Further, the performance evaluation of MVO with the recently proposed gradient free meta-heuristics algorithms namely ALO, DA, GWO, MFO, and WOA has been carried out to ascertain an optimized traversal path for UAVs in 2-D space. Effectiveness of these algorithms is compared based on the convergence rate, distribution of the fitness function values and computation efficiency. MVO showed superior performance in comparison to other existing algorithms. It can be seen that the exploration and exploitation processes of the MVO in path planning is superior than the other state-of-the-art meta-heuristic algorithms.

Future research plan includes, determining the subset of UAVs with which path recalculation information of a specific UAV should be exchanged. This will save a lot of time to exchange the information with all the UAVs of the environment; even with the ones whose path wound not be altered by the modification in the path of the particular UAV. Further, handling the noise during communication among the UAVs using Bayesian methods, such as-Kalman filter, particle filter, etc. and experimentally determining the exact influence of weight vectors used in the fitness function can also be considered as future enhancements in this paper.

References

[1] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.

[2] J. Li *et al.*, "A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9585–9596, Dec. 2016.

[3] N. Uchida, K. Ito, T. Ishida, and Y. Shibata, "Adaptive array antenna control methods with delay tolerant networking for the winter road surveillance system," *J. Internet Services Inf. Security*, vol. 7, no. 1, pp. 2–13, 2017.

[4] N. Uchida, S. Takeuchi, T. Ishida, and Y. Shibata, "Mobile traffic accident prevention system based on chronological changes of wireless signals and sensors," *J. Wireless Mobile Netw. Ubiquitous Comput. Depend. Appl.*, vol. 8, no. 3, pp. 57–66, 2017, doi: 10.22667/JOWUA.2017.09.30.057.

[5] Q. He, Q. Dong, B. Zhao, Y. Wang, and B. Qiang, "P2P traffic optimization based on congestion distance and DHT," *J. Internet Services Inf. Security*, vol. 6, no. 2, pp. 53–69, 2016.

[6] T. Kimura and C. Premachandra, "Optimal relay node selection in two-hop routing for intermittently connected MANETs," *J. Wireless Mobile Netw. Ubiquitous Comput. Depend. Appl.*, vol. 7, no. 1, pp. 23–38, 2016.

[7] J. Zeng *et al.*, "Multi-attribute aware path selection approach for efficient MPTCP-based data delivery," *J. Internet Services Inf. Security*, vol. 7, no. 1, pp. 28–39, 2017.

[8] C. YongBo, M. YueSong, Y. JianQiao, S. XiaoLong, and X. Nuo, "Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm," *Neurocomputing*, vol. 226, pp. 445–457, Nov. 2017, doi: 10.1016/j.neucom.2017.05.059.

[9] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013.

[10] S. Zhang, Y. Zhou, Z. Li, and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Adv. Eng. Softw.*, vol. 99, pp. 121–136, Sep. 2016.
[11] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Autom. Construct.*, vol. 81, pp. 25–33, Sep. 2017.
[12] Y. Chen, J. Yu, Y. Mei, Y. Wang, and X. Su, "Modified central force optimization (MCFO) algorithm for 3D UAV path planning," *Neurocomputing*, vol. 171, pp. 878–888, Jan. 2016.
[13] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.
[14] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2016.
[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[16] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl. Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
[17] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[18] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, 2016.
[19] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[20] M. R. Bonyadi and Z. Michalewicz, "Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 370–385, Jun. 2016.
[21] G. Li *et al.*, "Artificial bee colony algorithm with gene recombination for numerical function optimization," *Appl. Soft Comput.*, vol. 52, pp. 146–159, Mar. 2017.
[22] M. S. Kiran, H. Hakli, M. Gunduz, and H. Uguz, "Artificial bee colony algorithm with variable search strategy for continuous optimization," *J. Inf. Sci.*, vol. 300, pp. 140–157, Apr. 2015.
[23] A. Sadollaha, H. Eskandarb, A. Bahreininejadc, and J. H. Kima, "Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems," *J. Appl. Soft Comput.*, vol. 30, pp. 58–71, May 2015.

Puneet Kumar (S'17) received the B.E. degree in computer science and engineering from the Manipal Institute of Technology, Manipal, India, in 2014. He is currently pursuing the M.E. degree at the Thapar Institute of Engineering and Technology (Deemed University), Patiala, India.

He was with Oracle India Development Center, Madhapur, Hyderabad, as an Associate Applications Engineer from 2014 to 2016. His current research interests include cognitive science, machine learning, mathematical optimization, and Bayesian statistics.

Sahil Garg (S'16) received the B.Tech. degree in computer science and engineering from Maharishi Markandeshwar University Mullana, Ambala, India, in 2012, and the M.Tech. degree in computer science and engineering from Punjab Technical University, Jalandhar, India, in 2014. He is currently pursuing the Ph.D. degree in computer science and engineering at the Thapar Institute of Engineering and Technology (Deemed University), Patiala, India.

His current research interests include machine learning, big data analytics, knowledge discovery, game theory, and vehicular ad-hoc networks.

Amritpal Singh (S'17) received the M.E. degree minor in big data and advanced data structures from the Thapar Institute of Engineering and Technology (Deemed University), Patiala, India, in 2013.

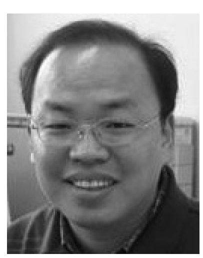
He has been a Research Scholar with the Computer Science Department, Thapar Institute of Engineering and Technology (Deemed University), since 2015. He served both industry and academia. His current research interests include probabilistic data structures, machine learning, and big data.

Shalini Batra (M'17) received the Ph.D. degree in computer science and engineering from the Thapar Institute of Engineering and Technology (Deemed University), Patiala, India, in 2012.

She is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed University). She has guided many research scholars leading to Ph.D. and M.E./M.Tech. She has authored over 60 research papers published in various conferences and journals. Her current research interests include machine learning, Web semantics, big data analytics, and vehicular ad-hoc networks.

Neeraj Kumar (M'16–SM'17) received the Ph.D. degree in CSE from Shri Mata Vaishno Devi University, Katra, India.

He was a Post-Doctoral Research Fellow with Coventry University, Coventry, U.K. He is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed University), Patiala, India. He has authored or coauthored over 150 technical research papers in leading journals and conferences of IEEE, Elsevier, Springer, and Wiley. Some of his research findings have been published in top cited journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, *IEEE Network*, *IEEE Communications Magazine*, IEEE WIRELESS COMMUNICATIONS, the IEEE INTERNET OF THINGS JOURNAL, IEEE SYSTEMS JOURNAL, *Future Generation Computer Systems*, the *Journal of Network and Computer Applications*, and *Computer Communications*. He has guided many research scholars leading to Ph.D. and M.E./M.Tech. degrees. His research is supported by funding from the Tata Consultancy Service and Department of Science and Technology.



Ilsun You (SM'13) received the M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, South Korea, in 1997 and 2002, respectively, and the Ph.D. degree from Kyushu University, Fukuoka, Japan, in 2012.

From 1997 to 2004, he was with THIN Multimedia Inc., Mountain View, CA, USA, the Internet Security Company, Ltd., and the Hanjo Engineering Company Ltd., Ulsan, South Korea, as a Research Engineer. He is currently an Associate Professor with the Department of Information Security Engineering, Soonchunhyang University, Asan, South Korea. His current research interests include Internet security, authentication, access control, and formal security analysis.

Dr. You has served or is currently serving as a main organizer of international conferences and workshops, such as MIST, MobiSec, and MobiWorld. He is the Editor-in-Chief of the *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*. He is an Editorial Board member of *Information Sciences*, the *Journal of Network and Computer Applications*, IEEE ACCESS, *Intelligent Automation and Soft Computing*, the *International Journal of Ad Hoc and Ubiquitous Computing*, *Computing and Informatics*, and the *Journal of High Speed Networks*. He is a Fellow of the IET.