*Computers, Materials & Continua*
DOI:10.32604/cmc.2021.018752
***Article***

Tech Science Press

# Path Planning of Quadrotors in a Dynamic Environment Using a Multicriteria Multi-Verse Optimizer

**Raja Jarray[1], Mujahed Al-Dhaifallah[2,*], Hegazy Rezk[3,4] and Soufiene Bouallègue[1,5]**

[1]Research Laboratory in Automatic Control (LARA), National Engineering School of Tunis (ENIT), University of Tunis El Manar, Tunis, 1002, Tunisia
[2]Department of Systems Engineering, King Fahd University of Petroleum & Minerals, Dhahran, 31261, Saudi Arabia
[3]College of Engineering at Wadi Addawaser, Prince Sattam Bin Abdulaziz University, Al-Kharj, 11911, Saudi Arabia
[4]Department of Electrical Engineering, Faculty of Engineering, Minia University, Minia, 61517, Egypt
[5]High Institute of Industrial Systems of Gabès (ISSIG), University of Gabès, Gabès, 6011, Tunisia
[*]Corresponding Author: Mujahed Al-Dhaifallah. Email: mujahed@kfupm.edu.sa
Received: 20 March 2021; Accepted: 23 April 2021

**Abstract:** Paths planning of Unmanned Aerial Vehicles (UAVs) in a dynamic environment is considered a challenging task in autonomous flight control design. In this work, an efficient method based on a Multi-Objective Multi-Verse Optimization (MOMVO) algorithm is proposed and successfully applied to solve the path planning problem of quadrotors with moving obstacles. Such a path planning task is formulated as a multicriteria optimization problem under operational constraints. The proposed MOMVO-based planning approach aims to lead the drone to traverse the shortest path from the starting point and the target without collision with moving obstacles. The vehicle moves to the next position from its current one such that the line joining minimizes the total path length and allows aligning its direction towards the goal. To choose the best compromise solution among all the non-dominated Pareto ones obtained for compromise objectives, the modified Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is investigated. A set of homologous metaheuristics such as Multiobjective Salp Swarm Algorithm (MSSA), Multi-Objective Grey Wolf Optimizer (MOGWO), Multi-Objective Particle Swarm Optimization (MOPSO), and Non-Dominated Genetic Algorithm II (NSGAII) is used as a basis for the performance comparison. Demonstrative results and statistical analyses show the superiority and effectiveness of the proposed MOMVO-based planning method. The obtained results are satisfactory and encouraging for future practical implementation of the path planning strategy.

**Keywords:** Quadrotors; path planning; dynamic obstacles; multi-objective optimization; global metaheuristics; TOPSIS decision-making; Friedman statistical tests

## 1 Introduction

In the last decades, unmanned aerial vehicles have acquired a great potential to complete an autonomous or semi-autonomous mission. A growing number of applications have appeared in real-world environments [1,2]. To achieve this autonomy, several challenges must be met. The trajectory planner is an essential part of the UAV autonomous control process [3]. Vehicles' trajectory planning is used to find a sequence of valid moves that allow a robot to move from an initial state to the desired end state. In general, a valid movement is a displacement that does not produce a collision and that respects the kinematic constraints of the robot [4]. In some planning problems, the location of obstacles can change over time. Thus, trajectory planning must respect the dynamic constraints that arise from the environment, i.e., mobile obstacles, and the considered specifications of the vehicles. Besides, most real path planning problems need to be solved by considering different conflicting goals such as price and quality. The trajectory planning problem is then considered as a Multi-Objective Optimization Problem (MOOP).

Many researchers have carried out various works to solve the Multi-Objective Path Planning (MOPP) problem for UAVs in a dynamic environment. The authors in [5] proposed a new methodology based on an Improved Gravitational Search Algorithm (IGSA) to solve the path planning for multi-robots in a dynamic environment. In [6], the authors investigated a new algorithm for UAV path planning problems based on Ant Colony Optimization (ACO) and artificial potential field. In [7], a Pigeon-Inspired Optimization (PIO) algorithm is used to optimize the initial path and another Fruit Fly Optimization Algorithm (FFOA) is used to solve the global path planning problem in a three-dimensional dynamic environment of oilfields. The authors in [8] proposed a novel Predator-Prey Pigeon-Inspired Optimization (PPPIO) to solve the Uninhabited Combat Aerial Vehicle (UCAV) 3D path planning problem in a dynamic environment. In [9], a novel integrated path planning approach based on an A* algorithm and local trace-back model has been proposed to solve such kind of hard problems. The authors in [10] developed an Improved Artificial Bee Colony (IABC) algorithm to solve the path planning problem in an environment of dynamic threats thanks to its fewer control parameters and faster convergence.

Although these works have been developed to solve the MOPP problem for a UAV flying in a dynamic environment, most of them converted the multi-objective problem into a single objective problem by using a weighted sum function [11]. This technique is easy to implement, but it is difficult to determine the best weights for the various contradictory objectives. The bias will be enjoined throughout the optimization process. Other methods have been developed to deal with the MOPP problem and obtain a set of optimal Pareto solutions. These methods can be classified into two types: exact methods and approximate methods [12]. An exact method such as Multi-Objective Branch & Bound (MOBB) is used to get the best solutions in the Pareto Optimal Frontier (POF) for the given path planning problem [13]. However, the computational complexity is highly elevated. The approximate multi-objective optimization methods are thus developed and used extensively in recent years to solve the MOOP. The path planning problem for UAVs in a dynamic environment is no exception. In [14], the path planning problem in an uncertain and dynamic environment is considered as a constrained multi-objective optimization problem with uncertain coefficients which is solved using a constrained Multi-Objective Particle Swarm Optimization (MOPSO) technique. In [15], a Modified Central Force Optimization (MCFO) based technique is introduced to solve the path-planning problem for a 6 DOF rotary-wing quadrotor helicopter. In such work, the theory of the Particle Swarm Optimization (PSO) and the mutation operator of the Genetic Algorithm (GA) are combined to improve the original CFO method.

In the above-mentioned studies, the idea of using multi-objective metaheuristics for path planning problem's formulation and the resolution seems a promising solution. To overcome the limits and inconveniences of the cited methods, particularly in terms of complexity and prohibitive time consuming, a systematic and efficient path planning method is proposed based on an advanced Multi-Objective Multi-Verse Optimization (MOMVO) algorithm. The main contributions of this paper are summarized as follows: 1) a novel strategy of reformulation and solving of a MOPP problem under operational constraints in a dynamic environment is proposed based on the concepts of the multi-criteria multi-verse optimization. Such a planning strategy allows guiding the quadrotor UAV to ensure destination position by calculate the next position in each step time while avoiding all moving obstacles. 2) A modified TOPSIS is employed as a higher-level decision-making approach to choose the best compromise solution among all the non-dominated solutions in the sense of Pareto. 3) A nonparametric statistical analysis method is proposed to compare all reported solvers for the hard path planning problem.

The remainder of this paper is organized as follows. In Section 2, the path planning problem for a quadrotor UAV is formulated as a multi-objective optimization problem under operational constraints. Section 3 presents the proposed multi-objective multi-verse optimizer to solve the formulated path planning problem. Section 4 describes the dynamical model of the studied quadrotor and the PID control design for the position and attitude dynamics stabilization and tracking. In Section 5, demonstrative results and comparisons are carried out and discussed. Section 6 concludes this paper.

## 2 Path Planning Problem Formulation

The quadrotor passed from the starting point A $(x_1, y_1, z_1)$ to the target one B $(x_n, y_n, z_n)$. The x-axis range $(x_1, x_n)$ is divided into $n-1$ equal segments and defined as $x_1, x_2, x_3, \ldots, x_n$. The time is incremented, the quadrotor has moved from position $\boldsymbol{w}_i = (x_i, y_i, z_i)$ to the next position $\boldsymbol{w}_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$ where the positions $\{x_i\}_{1 \le i \le n}$ are selected and the decision variables for optimization are defined as $\boldsymbol{\theta} = \{y_{i+1}, z_{i+1}\}, \forall\, i = 1, 2, \ldots, n-2$.

In the UAVs' path planning formalism, the length of the flight path is very important. In this work, the robot determines its next position from its current one and tries to align its direction towards the goal. Consider initially, the UAV placed in the location at a time $t$ in the space coordinate $(x_i, y_i, z_i)$. At the time $(t + \Delta t)$, it wants to move to the next position $(x_{i+1}, y_{i+1}, z_{i+1})$ such that the line joining between $\{(x_i, y_i, z_i),\ (x_{i+1}, y_{i+1}, z_{i+1})\}$ and $\{(x_{i+1}, y_{i+1}, z_{i+1}),\ (x_n, y_n, z_n)\}$ minimizes the total path length and allows the UAV to align its direction towards the goal. So, the first proposed objective function of the multi-objective optimization problem, denoted as $f_1$ , is defined as follows:

$$f_1(\boldsymbol{\theta}) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} + \sqrt{(x_{i+1} - x_n)^2 + (y_{i+1} - y_n)^2 + (z_{i+1} - z_n)^2} \quad (1)$$

Besides, the path planning process cannot totally ignore the dynamic characteristics of the UAV. When the UAV moves in the straighter path, the burden of the control system is reducing and the fuel cost of the flight process is decreasing [16]. The second objective function $f_2$ is thus defined as follows:

$$f_2(\boldsymbol{\theta}) = \arccos\left(\frac{\vec{\varphi}\,\vec{\psi}}{\left|\vec{\varphi}\right|\left|\vec{\psi}\right|}\right) \quad (2)$$

where $\vec{\varphi}$ is $(x_{i-1}-x_i, y_{i-1}-y_i, z_{i-1}-z_i)$ and $\vec{\psi}$ denotes $(x_{i+1}-x_i, y_{i+1}-y_i, z_{i+1}-z_i)$.

On the other hand, avoidance of obstacles in a dynamic flight environment is more complex than in a static one. To simplify the characterization of moving obstacles, they can be modeled by spheres of radius $R$ and center $c$. Thus, at the beginning of the program, the center, radius, and velocity vector of such spheres are initialized. At each time step $\Delta t$, the positions of a given moving obstacle are updated as:

$$v_{obs}(t+\Delta t) = v_{obs}(t) + a\Delta t \tag{3}$$

$$x_{obs}(t+\Delta t) = x_{obs}(t) + v_{obs}(t+\Delta t)\,\Delta t + 1/2a\Delta t^2 \tag{4}$$

$$y_{obs}(t+\Delta t) = y_{obs}(t) + v_{obs}(t+\Delta t)\,\Delta t + 1/2a\Delta t^2 \tag{5}$$

$$z_{obs}(t+\Delta t) = z_{obs}(t) + v_{obs}(t+\Delta t)\,\Delta t + 1/2a\Delta t^2 \tag{6}$$

where $v_{obs}$ is the velocity of a dynamic obstacle, and $a$ is its acceleration.

When the UAV moved from the actual position $(x_i, y_i, z_i)$ to the next position $(x_{i+1}, y_{i+1}, z_{i+1})$, it does not intersect any obstacle. So, it should be tested if any obstacle intersects the line segment connecting the two positions. The line through $(x_i, y_i, z_i)$ and $(x_{i+1}, y_{i+1}, z_{i+1})$ can be written as:

$$x = x_{i-1} + t\Delta x; \quad y = y_{i-1} + t\Delta y; \quad z = z_{i-1} + t\Delta z \tag{7}$$

where $\Delta x = x_i - x_{i-1}$, $\Delta y = y_i - y_{i-1}$, and $\Delta z = z_i - z_{i-1}$ denote the increments on the drone's positions.

The equation of a given sphere with the center's coordinates $(x_c, y_c, z_c)$ is written as follows:

$$(x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2 - R^2 = 0 \tag{8}$$

Then, substituting Eq. (7) into Eq. (8) leads to the following equation:

$$\mathcal{A}t^2 + \mathcal{B}t + \mathcal{C} = 0 \tag{9}$$

where $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ terms are defined as follows:

$$\mathcal{A} = \Delta x^2 + \Delta y^2 + \Delta z^2 \tag{10}$$

$$\mathcal{B} = 2\left(\Delta x\,(x_{i-1}-x_c) + \Delta y\,(y_{i-1}-y_c) + \Delta z\,(z_{i-1}-z_c)\right) \tag{11}$$

$$\mathcal{C} = (x_{i-1}-x_c)^2 + (y_{i-1}-y_c)^2 + (z_{i-1}-z_c)^2 - R^2 \tag{12}$$

As explained in [17], the solving of Eq. (9) can give an idea of the intersection or not of the drone path with the considered moving obstacles. Indeed, when the discriminate $\mathcal{D}_j$ of Eq. (13) is negative there are no intersections with the obstacles:

$$\mathcal{D}_j = \mathcal{B}^2 - 4\mathcal{A}\mathcal{C} \tag{13}$$

where $j = 1, 2, \ldots, m$, and $m \in \mathbb{N}$ is the number of moving obstacles.

Finally, the formulated multi-objective optimization problem for the path planning of the quadrotor UAV according to a given i$^{th}$ waypoint is defined as follows:

$$\begin{cases} \underset{\boldsymbol{\theta}\in\boldsymbol{\mathcal{F}}\subseteq\mathbb{R}^2}{\text{Minimize}}\ \varphi(\boldsymbol{\theta}) = \{f_1(\boldsymbol{\theta}),\ f_2(\boldsymbol{\theta})\} \\ \text{s.t: } g_j(\boldsymbol{\theta}) \le 0 \end{cases} \tag{14}$$

where $f_1(.)$, $f_2(.)$, and $g_j(.) = \mathcal{D}_j(.)$ are the cost and constraint functions given by Eqs. (1), (2), and (13), respectively, $\boldsymbol{\theta} = \{y_{i+1}, z_{i+1}\}$ is the decision variables and $\boldsymbol{\mathcal{F}} = \left\{\boldsymbol{\theta} \in \mathbb{R}^2 \mid \boldsymbol{\theta}^{\min} \le \boldsymbol{\theta} \le \boldsymbol{\theta}^{\max}\right\}$ is the bounded research space.

To handle the inequality constraints of the problem (14), the following external static type of penalty function is used as follows [18]:

$$\phi_k(\boldsymbol{\theta}) = f_k(\boldsymbol{\theta}) + \sum_{j=1}^{n_{con}} \mu_j \max\left\{0, g_j(\boldsymbol{\theta})\right\}^2 \tag{15}$$

where $\mu_j \in \mathbb{R}^+$ is the j$^{th}$ penalty parameter associated with the j$^{th}$ constraint, $n_{con}$ is the total number of the inequality constraints, and $k \in \{1, 2\}$.

## 3 Proposed Path Planning Algorithm

### *3.1 Multi-Objective Multi-Verse Optimizer*

Originally proposed by Mirjalili et al. [19], the Multi-Verse Optimizer (MVO) is a population-based metaheuristic inspired by the physics theory of the existence of multi-verse. In this formalism, the interaction among different universes is ensured based on the concepts of white/black holes and wormholes. The main motion equations of the MVO metaheuristic are given as follows [19]:

$$x_i^j = \begin{cases} \begin{cases} x_j + TDR + \left(ub_j - lb_j \times r_4 + lb_j\right) & r_3 < 0.5 \\ x_j + TDR - \left(ub_j - lb_j \times r_4 + lb_j\right) & r_3 < 0.5 \end{cases} & r_2 < WEP \\ x_i^j & r_2 \ge WEP \end{cases} \tag{16}$$

where $x_i^j$ denotes the jth component in the i$^{th}$ solution, $x_j$ is the jth variable of the best universe, $lb_j$ and $ub_j$ are the lower and upper bounds, respectively, $r_2$, $r_3$, and $r_4$ are random numbers in the interval [0, 1].

In Eq. (16), the terms $TDR$ and $WEP$ present the traveling distance rate and the wormholes' existence probability, respectively, and are defined as follows:

$$WEP = \rho_{\min} + iter\left(\rho_{\max} - \rho_{\min}\right) / Max\_iter \tag{17}$$

$$TDR = 1 - iter^{1/\gamma} / Max\_iter \tag{18}$$

where $\rho_{\min}$ and $\rho_{\max}$ are the wormhole existence probabilities, $iter$ is the current iteration, and $\gamma$ defines the exploitation accuracy.

To develop a multi-objective version of the MVO metaheuristic for the problem (14), a concept of the archive is investigated. The leader selection and roulette wheel approaches are used to select the fittest solutions according to the following probabilistic mechanism [20]:

$$\sigma_i = \vartheta_i / \alpha \tag{19}$$

where $\vartheta_i$ denotes the number of the vicinity solutions and $\alpha > 1$ is a constant.

Based on the above motion equations and the archive updating mechanism (19), a pseudo-code of MOMVO is given by Algorithm 1.

**Algorithm 1:** MOMVO

**Step 1**: Set the control parameters of MOMVO algorithm.
**Step 2**: Randomly initialize the population, i.e., the positions of universes.
**Step 3**: **While** ($iter < Max\_iter + 1$) **do**
Update *WEP* and *TDR* by applying Eqs. (17) and (18).
**For** each universe **do**
Boundary checking for the universes inside search space.
Calculate the inflation rate (fitness) of universes.
**End For**
Sort the fitness values
Find the non-dominated solutions.
Normalize the inflation rates of each universe.
Update the archive regarding the obtained non-dominated solutions.
**If** the archive is full **do**
Delete some solutions from the archive to hold the new.
**End if**
Update the position of universes according to Eq. (16).
**If** any new added solutions to the archive are outside boundaries **do**
Update the boundaries to cover the new solution(s).
**End if**
Increment *iter*
**Step 4**: Stop the algorithm's execution when it reaches *Max_iter*.

### *3.2 Decision-Making Model*

The selection of an optimal solution requires in particular a higher-level decision-making approach. The modified TOPSIS method is used to choose with more safety the best compromise solution among all the non-dominated ones in the sense of Pareto. Such a multiple-criteria decision-making approach is implemented as follows [21]:

**Step 1**: Obtain the decision matrix

$$D = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1w} \\ x_{21} & x_{22} & \dots & x_{2w} \\ \vdots & \vdots & \ddots & \vdots \\ x_{v1} & x_{v2} & \dots & x_{vw} \end{bmatrix}; \quad x_{ij} \ \ (i = 1, 2, \dots, v, \ j = 1, 2, \dots, w) \tag{20}$$

**Step 2**: Normalize the decision matrix

$$s_{ij} = \frac{x_{ij}}{\sum_{i=1}^{v} x_{ij}} \tag{21}$$

**Step 3**: Find the positive-and negative-ideal solutions

$$A^{+} = \left(a_1^{+}, a_2^{+}, \ldots, a_w^{+}\right); \quad A^{-} = \left(a_1^{-}, a_2^{-}, \ldots, a_w^{-}\right) \tag{22}$$

**Step 4**: Calculate the $w$-dimensional weighted Euclidean distances

$$\delta_i^{+} = \left[\sum_{j=1}^{w} \chi_j \left(a_j^{+} - a_{ij}\right)^2\right]^{1/2}; \quad \delta_i^{-} = \left[\sum_{j=1}^{w} \chi_j \left(a_j^{-} - a_{ij}\right)^2\right]^{1/2} \quad i = 1, 2, \ldots, v; \quad \sum_{j=1}^{w} \chi_j = 1 \tag{23}$$

**Step 5**: Calculate the relative closeness to the ideal solution

$$C_i = \delta_i^{-} / \left(\delta_i^{+} + \delta_i^{-}\right) \quad i = 1, 2, \ldots, v \tag{24}$$

**Step 6**: Choose an alternative with maximum $C_i$

## 4 Tracking of the Planned Paths

### *4.1 Dynamic Model*

The basic movements of the quadrotor are realized by varying the speed of each rotor as shown in Fig. 1. To evaluate the mathematical model of the quadrotor, two coordinate systems have been used, i.e., earth reference frame $\mathcal{F}_e\,(O_e, x_e, y_e, z_e)$ and body-fixed frame $\mathcal{F}_b\,(O_b, x_b, y_b, z_b)$.

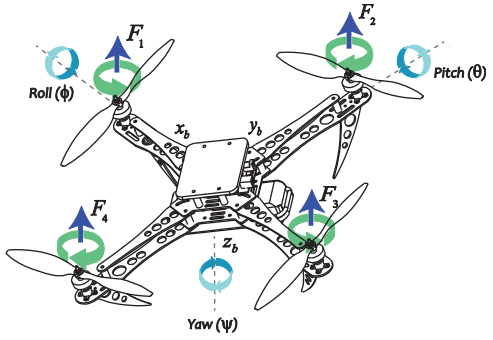



**Figure 1:** Quadrotor prototype and its related frames

The studied quadrotor is presented with its translational $\boldsymbol{\xi} = (x, y, z)^T$ and rotational $\boldsymbol{\eta} = (\phi, \theta, \psi)^T$ coordinates. Let a vector $\boldsymbol{\vartheta} = (p, q, r)^T$ denotes the angular velocity of the drone

in the body-frame. Using the Newton-Euler formalism, a dynamic nonlinear model is given as follows [22–24]:

$$\begin{cases} \ddot{x} = 1/m_Q \left(\cos\phi \cos\psi \sin\theta + \sin\phi \sin\psi\right) u_1 - \kappa_1/m_Q \dot{x} \\ \ddot{y} = 1/m_Q \left(\cos\phi \sin\psi \sin\theta - \sin\phi \cos\psi\right) u_1 - \kappa_2/m_Q \dot{y} \\ \ddot{z} = 1/m_Q \cos\phi \cos\theta u_1 - g - \kappa_3/m_Q \dot{z} \\ \dot{p} = \left(I_y - I_z\right)/I_x qr - J_r/I_x \omega_r q - \kappa_4/I_x p + 1/I_x u_2 \\ \dot{q} = \left(I_z - I_x\right)/I_y pr + J_r/I_y \omega_r p - \kappa_5/I_y q + 1/I_y u_3 \\ \dot{r} = \left(I_x - I_y\right)/I_z pq - \kappa_6/I_z r + 1/I_z u_4 \end{cases} \tag{25}$$

where $-\pi/2 \le \phi \le \pi/2$, $-\pi/2 \le \theta \le \pi/2$, and $-\pi \le \psi \le \pi$ are the roll, pitch, and yaw Euler angles, respectively, $m_Q$ denotes the mass of the quadrotor, $g$ is the gravitational acceleration, $J_r$ is the z-axis inertia of the propellers, $I_x$, $I_y$ and $I_z$ denote the inertias of the quadrotor, $\kappa_{1,2,\ldots,6}$ are the aerodynamic friction and drag coefficients, and $\omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$ is the overall residual rotor angular velocity.

The control inputs $u_1$, $u_2$, $u_3$ and $u_4$ are defined as follows:

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} \tag{26}$$

where $b$ and $d$ denote the lift body and drag propellers coefficients, respectively, $l$ denote the distance from the center of mass to each motor.

### *4.2 Tracking PID Controllers' Design*

The proposed control system of the quadrotor UAV is shown in Fig. 2. Such a control scheme is composed of an inner-loop attitude controller and an outer-loop position controller. The two controllers are designed with the classical PID structure as follows:

$$u_{follow}(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau + K_D \frac{de(t)}{dt} \tag{27}$$

where $e$ is the tracking error between the desired reference and the accessible system output, $K_P$, $K_I$, and $K_D$ are the proportional, integral, and derivative gains of the PID controller, respectively.

Two cascade loops for decoupling control of all flight dynamics are investigated. An inner loop is set to ensure the attitude and heading's tracking. And the outer loop is designed for the positions $(x, y)$ and altitude $z$ dynamics [25]. The desired trajectories for the attitude variables $\phi_d$ and $\theta_d$ are generated from Eqs. (28) and (29) shown as virtual control laws for the translational dynamics [26–28]:

$$u_x = \left(\cos\phi \cos\psi \sin\theta + \sin\phi \sin\psi\right) \tag{28}$$

$$u_y = \left(\cos\phi \sin\psi \sin\theta - \sin\phi \cos\psi\right) \tag{29}$$

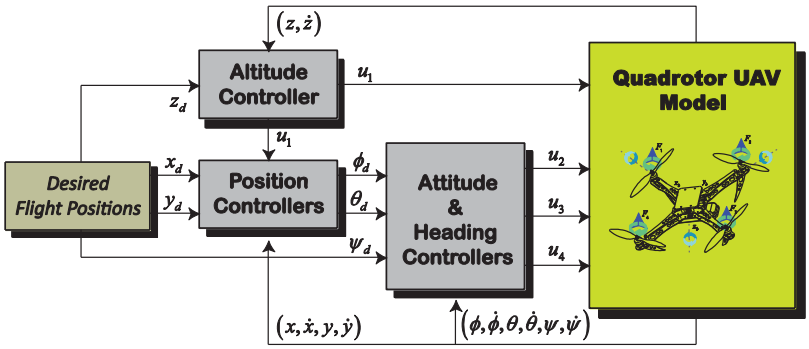



**Figure 2:** Full control scheme of the quadrotor

Solving Eqs. (28) and (29) for a given yaw angle $\psi$ leads to the desired roll and pitch angles' formula respectively given as follows:

$$\phi_d = \arctan\left(\frac{u_x \sin\psi - u_y \cos\psi}{\sqrt{\left(1 - u_x^2 \sin^2\psi + 2u_x u_y \cos\psi \sin\psi + u_y^2 \sin^2\psi - u_y^2\right)}}\right) \tag{30}$$

$$\theta_d = \arcsin\left(\frac{u_x \cos\psi + u_y \sin\psi}{\sqrt{\left(1 - u_x^2 \sin^2\psi + 2u_x u_y \cos\psi \sin\psi + u_y^2 \sin^2\psi - u_y^2\right)}}\right) \tag{31}$$

## 5 Simulation Results and Discussion

To illustrate the performance of the proposed MOMVO-based method, a 3D dynamic environment with moving obstacles is developed under the MATLAB/Simulink software. An interactive Graphical User Interface (GUI) has been implemented for the different simulations. The quadrotor's 3D trajectory can be viewed by designing an animated quadrotor that receives the simulation data and performs the dynamical responses. Some performance index values, such as path length, flight time, and response plots of the quadrotor along the X, Y, and Z-axis, are presented and discussed. In this study, the quadrotor's physical parameters are given in Appendix A. Tab. 1 gives the different flight scenarios considered in the dynamic environment.

To compare the performance of the proposed MOMVO-based planning method, others algorithms such as MSSA, MOGWO, MOPSO, and NSGAII are retained. The control parameters of such optimizers are summarized in Tab. 2.

To have a fair and reliable comparison, the common parameters such as the maximum number of iterations and the population size are set as 100 and 50, respectively. For statistical comparison purposes, all algorithms are independently executed 10 times and compared in the sense of the solutions' quality. In each step time, the quadrotor calculates the next position by solving the formulated multi-objective optimization problem (14) based on a multi-objective optimization algorithm. The execution of the reported algorithms leads to obtain a set of non-dominated solutions as shown in Fig. 3. These Pareto fronts are considered at the first time-step to have a fair comparison.

**Table 1:** Values of the simulation parameters

| Scenarios | Starting point [km] | Destination point [km] | Center of dynamic obstacles [km] | Dynamic obstacles' speed [km/s] |
|---|---|---|---|---|
| **1** | [0.0, 0.0, 0.0] | [9.0, 8.0, 0.0] | [5, 5, 2]; [3, 3, 2]; [5, 3, 1]; [2, 1, 1]; [6, 2, 2] | [4, −2, 1]; [2, −2, −2]; [4, 2, 2]; [2 2 2]; [−2, 2, −2] |
| **2** | [1.0, 2.0, 0.0] | [10.0, 10.0, 0.0] | [1, 3, 1]; [3, 5, 1]; [4, 4, 3]; [5, 5, 4]; [7, 3, 4]; [8, 2, 1]; [9, 5, 2] | [2, 1, 1]; [3, −3, 1]; [4, 1, −2]; [2, 1, 1]; [−1, −2, 2]; [0.5, 1, −1]; [1, -1, 2] |
| **3** | [1.0, 2.0, 0.0] | [15.0, 10.0, 0.0] | [2, 3, 1]; [2, 4, 1]; [4, 3, 2]; [5, 3, 3]; [5, 5, 2]; [6, 4, 1]; [7, 7, 2]; [7, 3, 4]; [8, 6, 3]; [10, 8, 2] | [−1, 3, −1]; [−1, 1, 1]; [2, 2, 1]; [1, 2, 4]; [0.2, 1, 3]; [1, −1, 1]; [2, 1, 2]; [−1, 2, 2]; [1, 2, 1]; [3 0.5 0. 2] |
| **4** | [2.0, 4.0, 0.0] | [16.0, 13.0, 0.0] | [1, 3, 1]; [2, 5, 2]; [2, 4, 3]; [2, 7, 1]; [3, 2, 1]; [3, 3, 3]; [4, 1, 2]; [4, 5, 4]; [6, 7, 1]; [7, 2, 2]; [8, 5, 2]; [10, 8, 3] | [1, 2, 1]; [−2, 1, 1]; [3, −1, 3]; [2, 1, 2]; [−1, 3, 1]; [1, 2, −2]; [2, −1, 2]; [4, 1, 2]; [3, 2, 1]; [0.5, 2, −2]; [1, −2, 1]; [1, 2, 3] |
| **5** | [0.0, 4.0, 0.0] | [16.0, 15.0, 0.0] | [1, 4, 1]; [2, 5, 1]; [2, 2, 1]; [3, 2, 4]; [3, 7, 2]; [4, 2, 1]; [4, 5, 3]; [4, 8, 1]; [5, 3, 4]; [6, 5, 2]; [7, 2, 1]; [7, 4, 5]; [8, 1, 2]; [8, 8, 1]; [9, 5, 2] | [1, 3, 1]; [3, −1, 1]; [4, 1, 3]; [2. 2. 4]; [−1, 3, 1]; [2, −2, 1]; [3, 1, 2]; [1, 3, −2]; [1, 1, 1]; [2, −0.5, 2]; [1, 3, 2]; [1, 2, 1]; [2, 2, −2]; [1, −2, 1]; [1, 1, 2] |

**Table 2:** Control parameters of the reported optimizers

| Optimizers | Parameters |
|---|---|
| MSSA [29] | Without control parameters (free-parameters algorithm) |
| MOGWO [30] | Grid inflation 0.1, grids per each dimension 10, leader selection pressure 4, and repository member selection pressure 2 |
| MOPSO [31] | Cognitive and social accelerations 2, grid inflation 0.1, leader selection pressure 2, and grids per each dimension 7 |
| NSGAII [32] | Crossover percentage 0.7, mutation percentage 0.4, and mutation rate 0.02 |
| MOMVO | Lower and upper wormhole existence probabilities $\rho_{\min} = 0.2$ and $\rho_{\max} = 1$ |

These results show the repartition topology of the set of optimal non-dominated Pareto solutions on the compromised surface. A higher-level decision-making approach, i.e., the modified TOPSIS, selects the best compromise solution. These illustrative results show the high optimization performance of the proposed MOMVO algorithm in terms of convergence dynamics and solution distribution. The optimal Pareto solutions are strongly distributed for the two considered objectives of Eqs. (1) and (2) and under operational constraints of Eq. (13), which means a good coverage of the non-dominated set of solutions of the optimization problem (14) for proposed algorithms, except the NSGAII one which it could only find some feasible solutions.

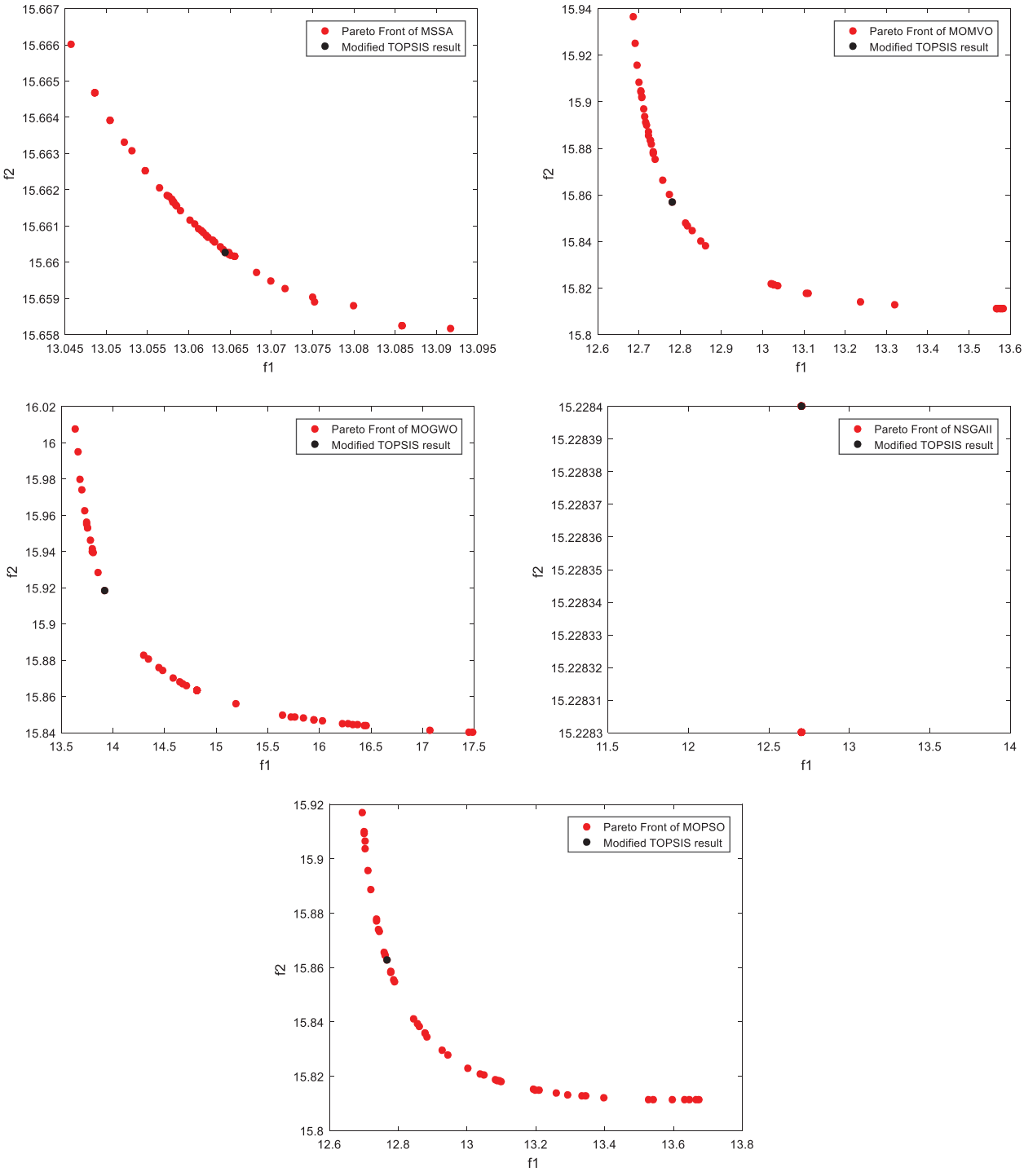



**Figure 3:** Pareto Fronts for the generation of the first position

To highlight the diversity and coverage of the obtained non-dominated solutions, various metrics such as Maximum Spread (MS) [33,34], Hyper Volume (HV) [35], and C metric [36] have been employed in this study. The statistical results for the MS metric are presented in Tab. 3. The proposed MOMVO algorithm is superior to the other reported optimizers in terms of the largest value of MS metric. It becomes the best in terms of having high coverage properties. Tab. 4 shows the optimization results of the HV index for the reported algorithms. The statistical results show that the MOGWO algorithm followed by the MOMVO presents the two best solvers compared with other proposed algorithms in terms of diversity and convergence performance. Tab. 5 shows the comparative results for the reported algorithms in terms of the C metric. The proposed MOMVO solver surpassed all the other competitive ones and it dominates more than 2% of the MSSA solutions, 99% of the MOGWO solutions, 15% of the NSGA-II solutions, and 4 % of the MOPSO solutions on average.

**Table 3:** Comparison of the MS metric for the reported algorithms

| | MSSA | MOMVO | MOGWO | NSGAII | MOPSO |
|---|---|---|---|---|---|
| Best | 18.4022 | 20.8615 | 11.8244 | 17.9264 | 20.1983 |
| Mean | 18.3603 | 20.4966 | 11.0232 | 06.5371 | 20.0943 |
| Worst | 18.3295 | 19.9721 | 10.0944 | 02.4744 | 19.9385 |
| STD | 00.0227 | 00.2899 | 00.5777 | 06.0948 | 00.0910 |

**Table 4:** Comparison of the HV metric for the reported algorithms

| | MSSA | MOMVO | MOGWO | NSGAII | MOPSO |
|---|---|---|---|---|---|
| Best | 0.000247 | 0.1074 | 0.8655 | 2.52e-09 | 0.0917 |
| Mean | 0.00020 | 0.1017 | 0.7521 | 5.18e-10 | 0.0883 |
| Worst | 0.00011 | 0.0820 | 0.5420 | 0.0000 | 0.0811 |
| STD | 4.003e-05 | 0.0079 | 0.1177 | 2.52e-09 | 0.0031 |

**Table 5:** Comparison of the C metric for the reported algorithms

| | Best | Mean | Worst | STD |
|---|---|---|---|---|
| C (MOMVO, MSSA) | 0.1000 | 0.0200 | 0.0100 | 0.0521 |
| C (MSSA, MOMVO) | 0.4221 | 0.3233 | 0.2400 | 0.0752 |
| C (MOMVO,MOGWO) | 1.0000 | 0.9900 | 0.9800 | 0.0012 |
| C (MOGWO, MOMVO) | 0.0600 | 0.0400 | 0.0000 | 0.0314 |
| C (MOMVO, NSGAII) | 0.2100 | 0.1500 | 0.0100 | 0.0546 |
| C (NSGAII, MOMVO) | 1.0000 | 0.9700 | 0.8400 | 0.0642 |
| C (MOMVO, MOPSO) | 0.0800 | 0.0400 | 0.0000 | 0.0253 |
| C (MOPSO, MOMVO) | 0.0200 | 0.0067 | 0.0000 | 0.0103 |

To analyze the statistical performance of the MOMVO-based planning method, a comparative study with MSSA, MOGWO, NSGAII, and MOPSO algorithms is performed on three performance criteria, such as path length, elapsed time, and capacity to avoid the moving obstacles as

shown in Tab. 6. While considering two performance criteria, i.e., path length and path travel times, a statistical comparison according to its mean value based on the nonparametric Friedman test is implemented and discussed to indicate the significant differences among the performances of the reported algorithms. The Iman-Davenport extension of the classical Friedman test [37] provides the computed value $F_{F_1} = 27.25$ for the elapsed time criterion and $F_{F_2} = 79.33$ for the flight time criterion. For the five reported algorithms ($\zeta = 5$) and five scenarios ($\lambda = 5$) at a 95% level of significance, the critical value of the $F$ distribution with $\zeta - 1$ and $(\zeta - 1)(\lambda - 1)$ degrees of freedom is equal to $F_{4,16,0.05} = 3.01 < F_{F_1} < F_{F_2}$. So the null hypothesis is declined and there are significant differences among the performance.

**Table 6:** Optimization result of the path length and the flight time

| Scenarios | | MSSA | | MOMVO | | MOGWO | | NSGAII | | MOPSO | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PL[1] | ET[2] | PL | ET | PL | ET | PL | ET | PL | ET |
| 1 | Best | 13.054 | 482.27 | 12.301 | 480.22 | 14.243 | 789.4 | 12.192 | 1110.2 | 17.191 | 757.1 |
| | Mean | 13.066 | 496.08 | 12.427 | 493.66 | 14.861 | 854.5 | 12.326 | 1265.4 | 19.857 | 821.5 |
| | Worst | 13.306 | 516.63 | 13.251 | 506.63 | 15.251 | 958.8 | 13.241 | 1421.5 | 22.502 | 886.7 |
| | STD | 0.1083 | 5.5645 | 0.2544 | 4.0314 | 0.3514 | 5.142 | 0.641 | 5.874 | 0.554 | 4.544 |
| 2 | Best | 12.621 | 520.12 | 12.451 | 501.24 | 14.892 | 834.5 | 12.351 | 1187.2 | 18.524 | 798.4 |
| | Mean | 12.462 | 556.23 | 12.384 | 524.24 | 15.214 | 863.2 | 12.841 | 1354.2 | 20.241 | 854.2 |
| | Worst | 13.762 | 620.63 | 13.484 | 589.92 | 15.458 | 987.1 | 13.541 | 1465.5 | 22.741 | 932.1 |
| | STD | 0.2145 | 5.741 | 0.2014 | 4.1345 | 0.3741 | 5.142 | 0.667 | 5.984 | 0.574 | 4.651 |
| 3 | Best | 16.201 | 825.12 | 16.121 | 817.21 | 17.451 | 1115.7 | 15.942 | 7545.3 | 25.874 | 2624.8 |
| | Mean | 17.149 | 855.41 | 16.443 | 848.05 | 18.899 | 1570.6 | 16.512 | 7646.5 | 27.317 | 2909.3 |
| | Worst | 17.354 | 945.12 | 17.207 | 895.45 | 19.542 | 1618.2 | 16.774 | 8068.5 | 28.651 | 3187.4 |
| | STD | 0.2451 | 5.781 | 0.2214 | 4.245 | 0.392 | 5.413 | 0.6754 | 6.2654 | 0.6224 | 4.988 |
| 4 | Best | 16.673 | 923.5 | 16.654 | 891.2 | 19.214 | 1704.7 | 16.741 | 7998.5 | 27.874 | 3478.1 |
| | Mean | 16.715 | 1064.5 | 16.682 | 917.64 | 20.415 | 1845.8 | 16.784 | 8142.2 | 28.145 | 3584.2 |
| | Worst | 20.854 | 1123.5 | 19.177 | 1013.2 | 21.214 | 1991.4 | 17.514 | 8534.6 | 29.941 | 3782.1 |
| | STD | 0.3641 | 5.804 | 0.3146 | 5.5243 | 0.4201 | 5.6103 | 0.6774 | 6.4541 | 0.6412 | 5.2341 |
| 5 | Best | 20.121 | 1105.4 | 19.471 | 1089.6 | 21.754 | 2154.2 | 19.104 | 9120.1 | 30.987 | 4212.2 |
| | Mean | 21.286 | 1188.19 | 20.275 | 1129.07 | 22.730 | 2263.7 | 19.115 | 9293.3 | 33.281 | 4422.4 |
| | Worst | 21.764 | 1272.3 | 20.941 | 1262.4 | 23.147 | 2549.7 | 20.321 | 9752.2 | 35.102 | 4949.7 |
| | STD | 0.4587 | 5.8715 | 0.4031 | 5.6441 | 0.5342 | 5.4924 | 0.6871 | 6.6934 | 0.6733 | 5.441 |

*Notes*: [1]Path Length (Km), [2]Elapsed Time (s).

To find out which algorithms differ from the others, Fisher's LSD post-hoc test is applied [37]. The ranks' sums for all the proposed methods in the different scenarios for the two performance indices are summarized in Tabs. 7 and 8. The paired comparisons are given in Tabs. 9 and 10. The bold and underlined values indicate that the absolute difference of the rank's sum $|R_i - R_j|$ is greater than the critical values 4.2398 and 2.5963 for the path length and flight time criteria, respectively [37]. From the statistical results based on the two performance criteria, i.e., path length and path travel time, it is obvious that the MOMVO algorithm outperforms the other reported algorithms for the path planning problem of the quadrotor in the considered dynamic environment.

**Table 7:** Ranks' sum of mean performances: path length criterion

| | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | | Scenario 5 | | Ranks' sum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Score | Rank | Score | Rank | Score | Rank | Score | Rank | Score | Rank | |
| MSSA | 13.066 | 3 | 12.462 | 2 | 17.149 | 3 | 16.715 | 2 | 21.286 | 3 | 13 |
| MOMVO | 12.427 | 2 | 12.384 | 1 | 16.443 | 1 | 16.682 | 1 | 20.275 | 2 | 7 |
| MOGWO | 14.861 | 4 | 15.214 | 4 | 18.899 | 4 | 20.415 | 4 | 22.730 | 4 | 20 |
| NSGA-II | 12.326 | 1 | 12.841 | 3 | 16.512 | 2 | 16.784 | 3 | 19.115 | 1 | 10 |
| MOPSO | 19.857 | 5 | 20.241 | 5 | 27.317 | 5 | 28.145 | 5 | 33.281 | 5 | 25 |

**Table 8:** Ranks' sum of mean performances: path travel time criterion

| | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | | Scenario 5 | | Ranks' sum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Score | Rank | Score | Rank | Score | Rank | Score | Rank | Score | Rank | |
| MSSA | 496.08 | 2 | 556.23 | 2 | 855.41 | 2 | 1064.5 | 2 | 1188.1 | 2 | 10 |
| MOMVO | 493.66 | 1 | 524.24 | 1 | 848.05 | 1 | 917.64 | 1 | 1129.0 | 1 | 5 |
| MOGWO | 854.5 | 4 | 863.2 | 4 | 1570.6 | 3 | 1845.8 | 3 | 2263.7 | 3 | 17 |
| NSGA-II | 1265.4 | 5 | 1354.2 | 5 | 7646.5 | 5 | 8142.2 | 5 | 9293.3 | 5 | 25 |
| MOPSO | 821.5 | 3 | 854.2 | 3 | 2909.3 | 4 | 3584.2 | 4 | 4422.4 | 4 | 18 |

**Table 9:** Paired comparison of the proposed metaheuristics: path length criterion

| $\lvert R_i - R_j \rvert$ | MOMVO | MOGWO | NSGA-II | MOPSO |
|---|---|---|---|---|
| MSSA | **6** | **7** | 3 | **12** |
| MOMVO | – | **13** | 3 | **18** |
| MOGWO | – | 0 | **10** | **5** |
| NSGA-II | – | – | – | **15** |

**Table 10:** Paired comparison of the proposed metaheuristics: path travel time criterion

| $\lvert R_i - R_j \rvert$ | MOMVO | MOGWO | NSGA-II | MOPSO |
|---|---|---|---|---|
| MSSA | **5** | **7** | **15** | **8** |
| MOMVO | – | **12** | **20** | **13** |
| MOGWO | – | – | **8** | 1 |
| NSGA-II | – | – | – | **7** |

By visualizing the simulations of the 3D trajectory of the quadrotor, all proposed algorithms succeed in completing the flight mission still avoiding all the moving obstacles. The simulation results of the proposed MOMVO-based method are shown in Fig. 4. Several periods of time are given where T is the total time of the UAV path planning. These results show that the quadrotor avoids all dynamic obstacles in all four periods of time and guarantee the planning performance

of the proposed MOMVO-based method. The time-domain responses of the controlled position dynamics are shown in Figs. 5–9 corresponding to the mean case of the multicriteria optimization. The proposed flight PID controllers allow the quadrotor to reach the desired trajectories. The PID controller gains' selection is achieved by an iterative trials-errors-based method. Even though there were minor tracking errors in the time-domain responses of the closed-loop, these results remain encouraging. The tracking errors can be due to using PID gains that were not defined from the control design approach but by trials-errors based tuning.

By visualizing these figures, we can notice that the proposed algorithm MOMVO gives the most direct path, which guarantees high efficiency in flight missions. The MOPSO algorithm generates a trajectory with many fluctuations along the Z-axis. From these results, the quadrotor has started the mission after a time delay which is due to the calculation time of the next point. A minimum execution time for an algorithm ensures the high efficiency of collision avoidance with the dynamic obstacles and causes a minimum flight time.

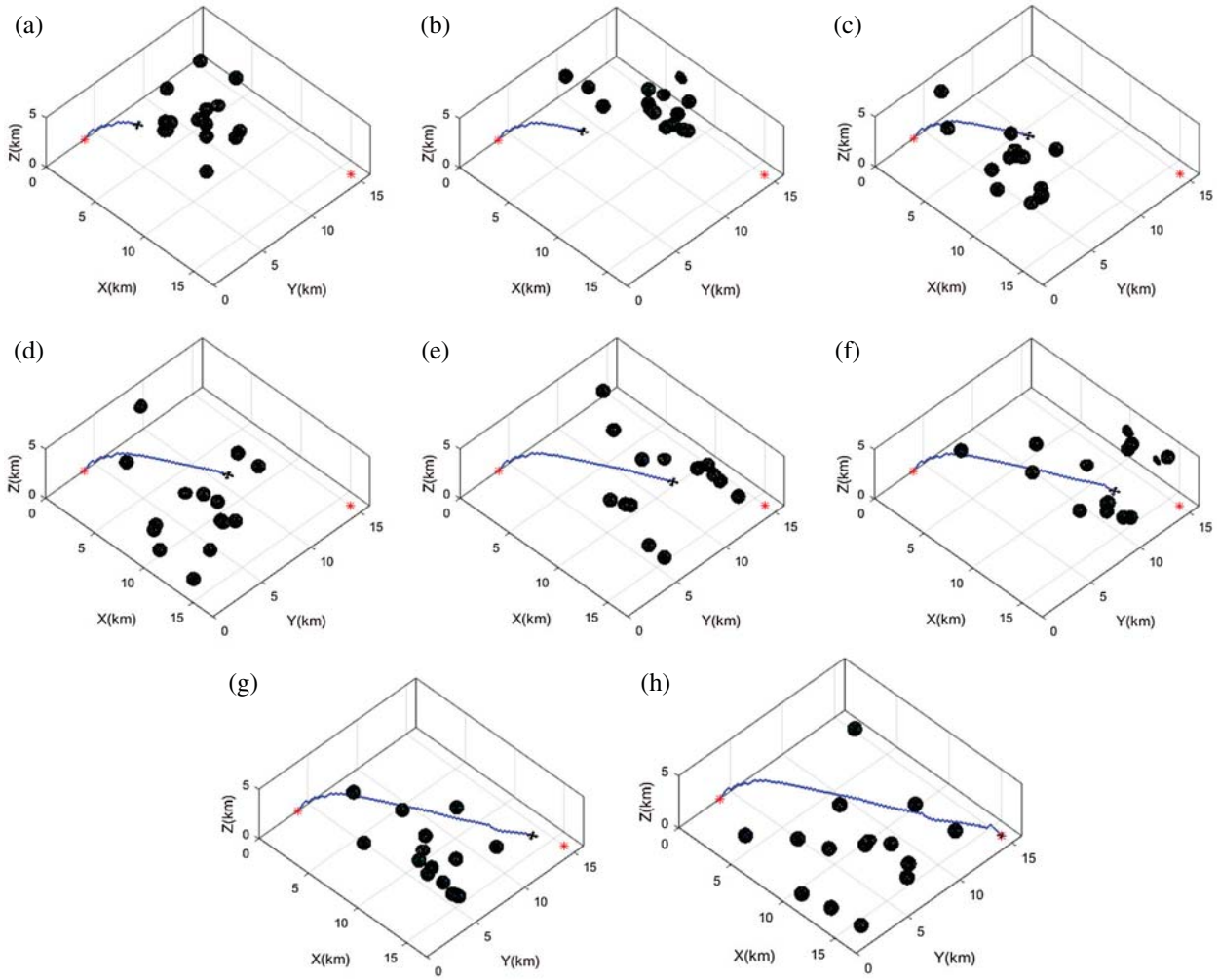



**Figure 4:** Planned path tracking in scenario 5 for the proposed MOMVO-based approach: (a) T/8; (b) 2T/8; (c) 3T/8; (d) 4T/8; (e) 5T/8; (f) 6T/8; (g) 7T/8; (h) total mission time (T)

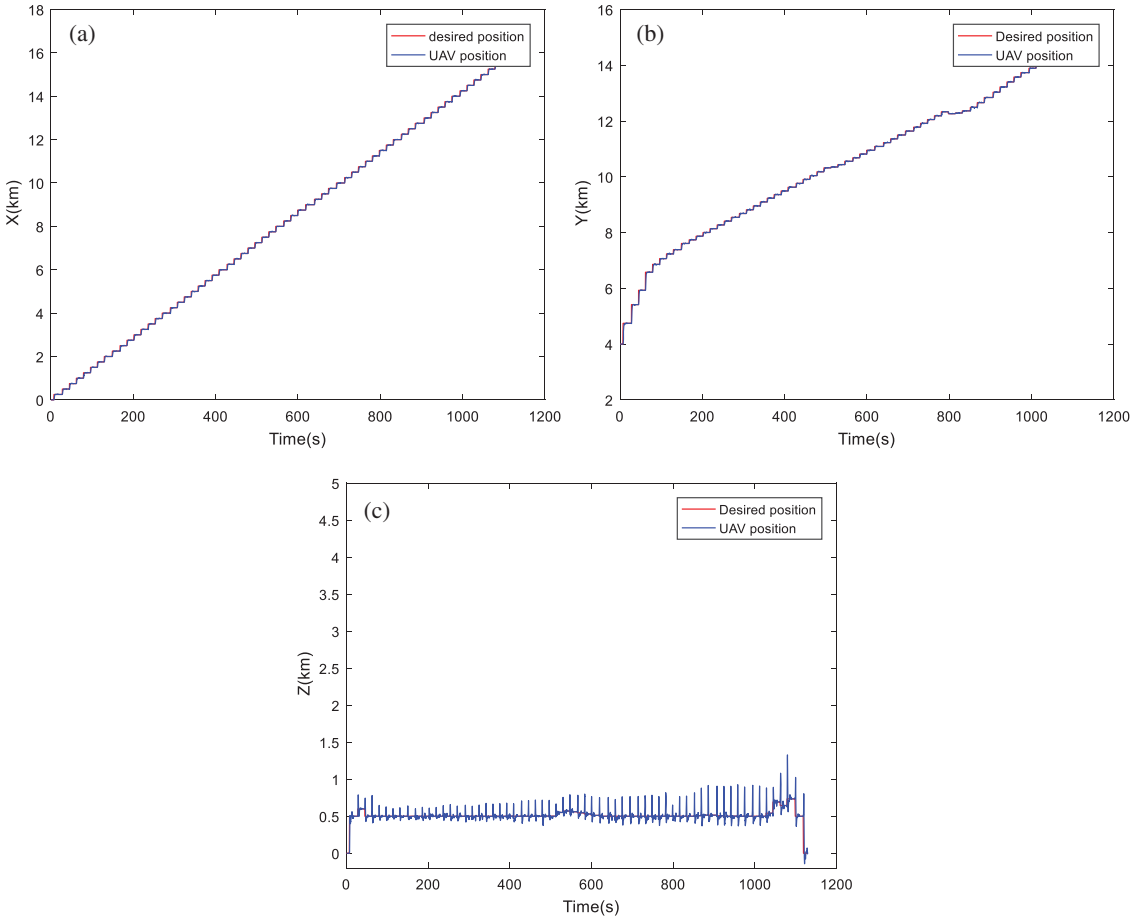
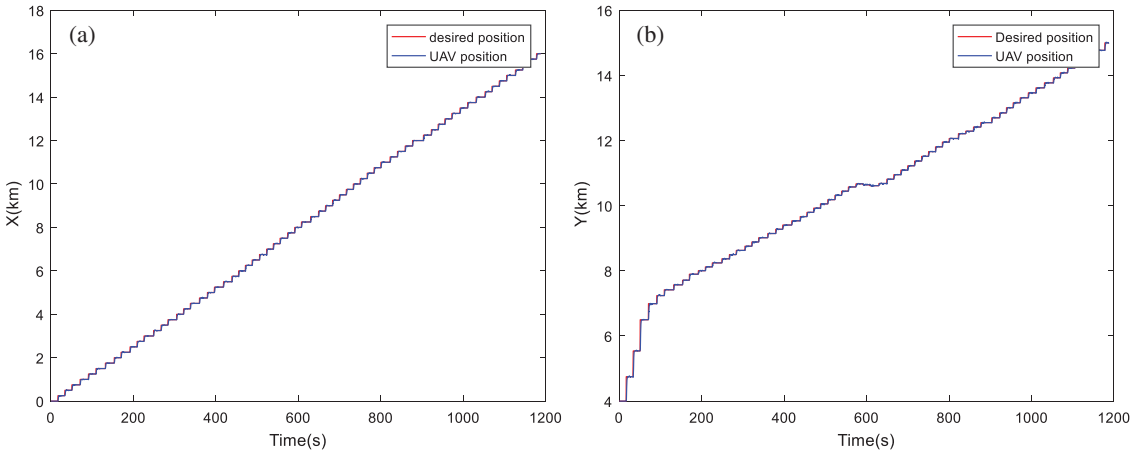



**Figure 5:** Position tracking based on MOMVO algorithm: (a) Tracking dynamics on X-axis; (b) Tracking dynamics on Y-axis; (c) Tracking dynamics on Z-axis



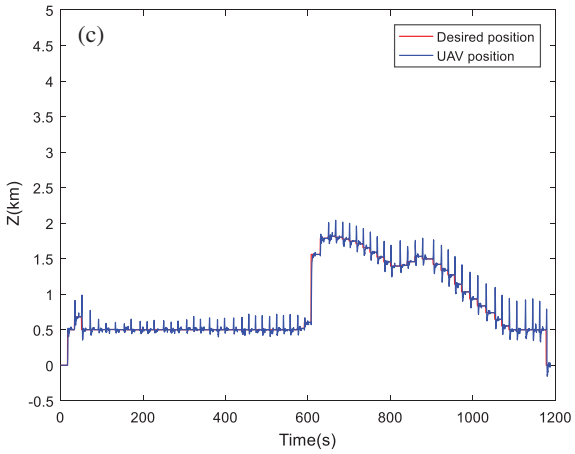



**Figure 6:** Position tracking based on MSSA algorithm: (a) Tracking dynamics on X-axis; (b) Tracking dynamics on Y-axis; (c) Tracking dynamics on Z-axis

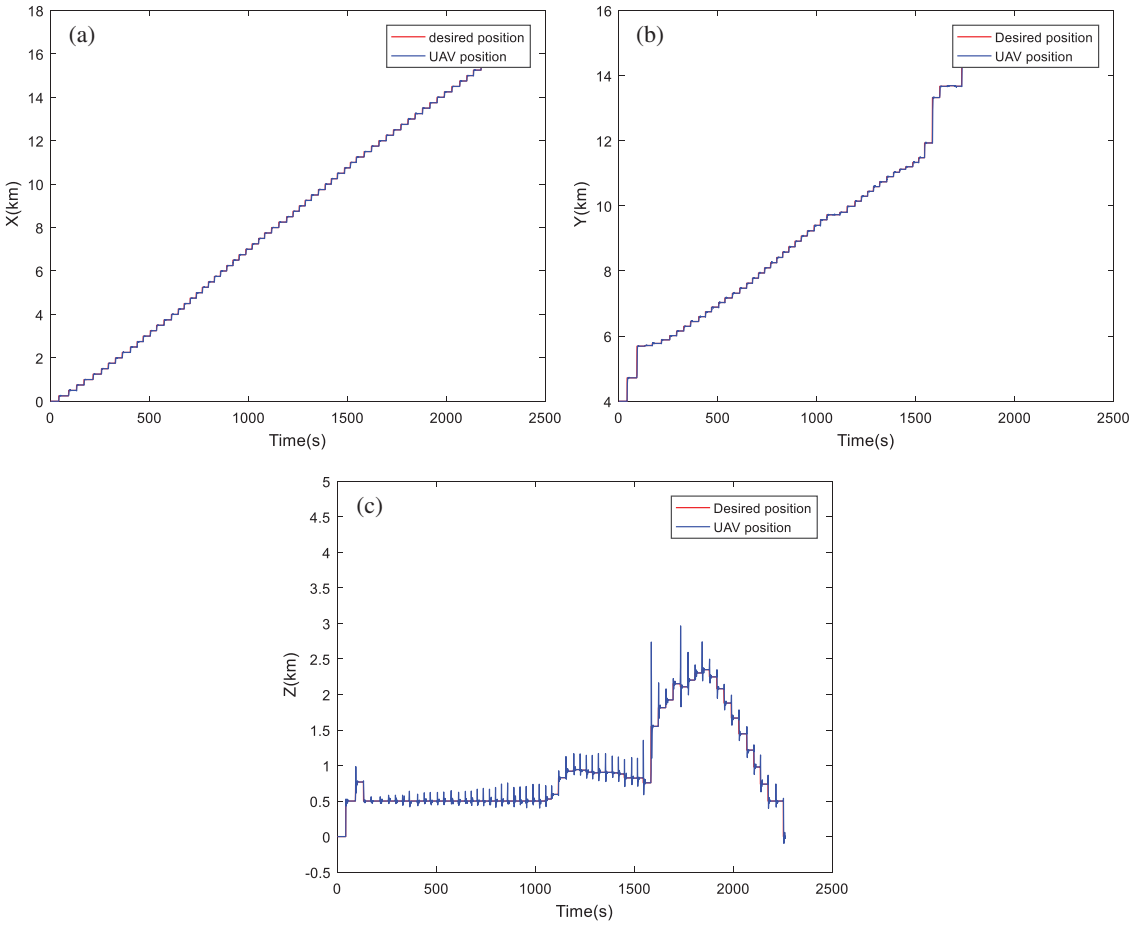



**Figure 7:** Position tracking based on MOGWO algorithm: (a) Tracking dynamics on X-axis; (b) Tracking dynamics on Y-axis; (c) Tracking dynamics on Z-axis

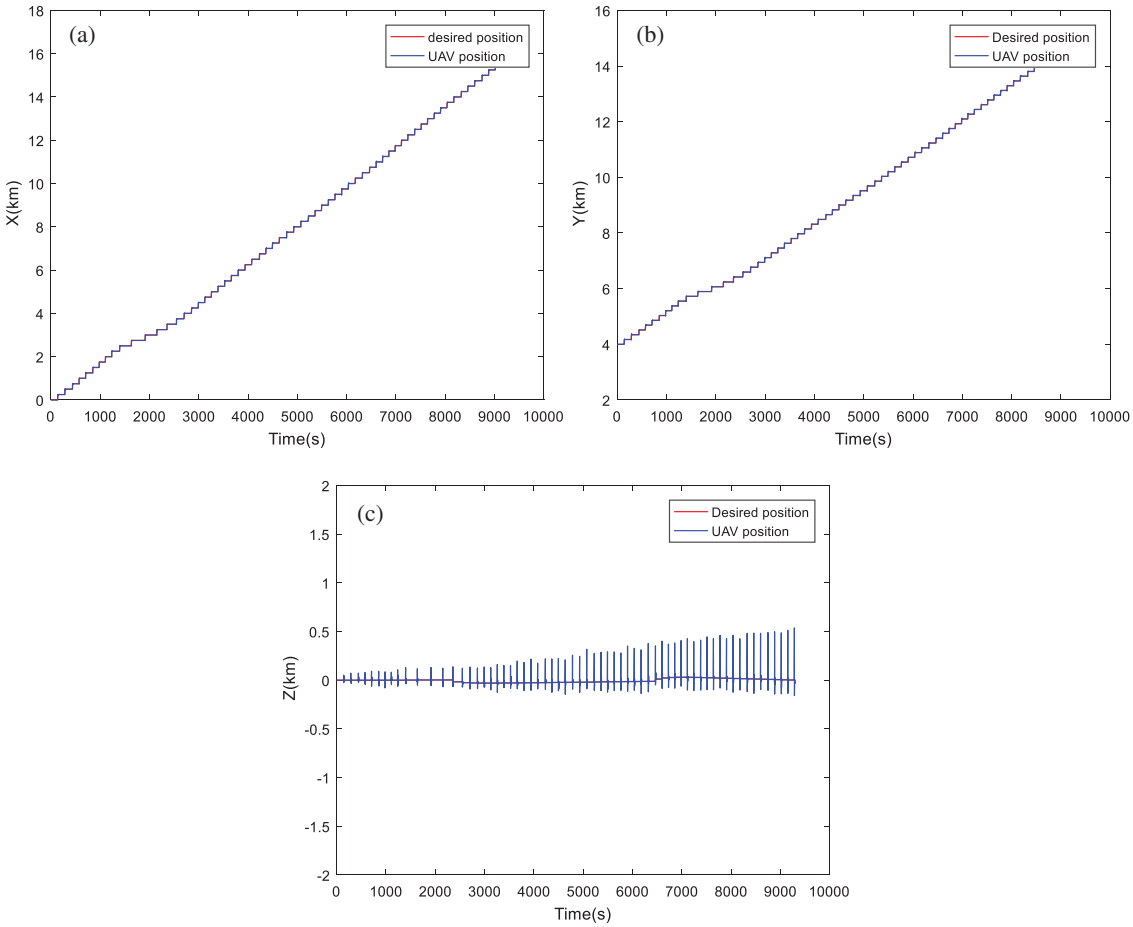
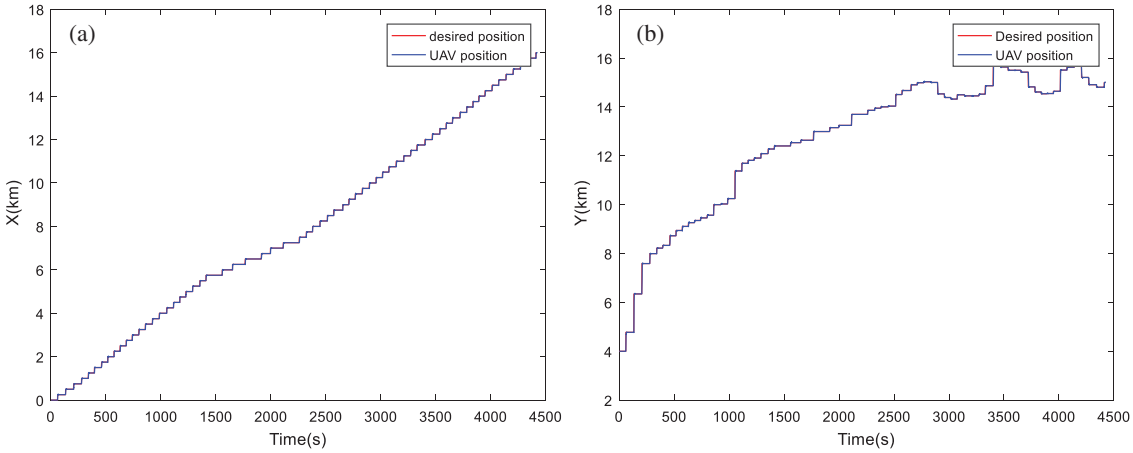



**Figure 8:** Position tracking based on NSGA-II algorithm: (a) Tracking dynamics on X-axis; (b) Tracking dynamics on Y-axis; (c) Tracking dynamics on Z-axis



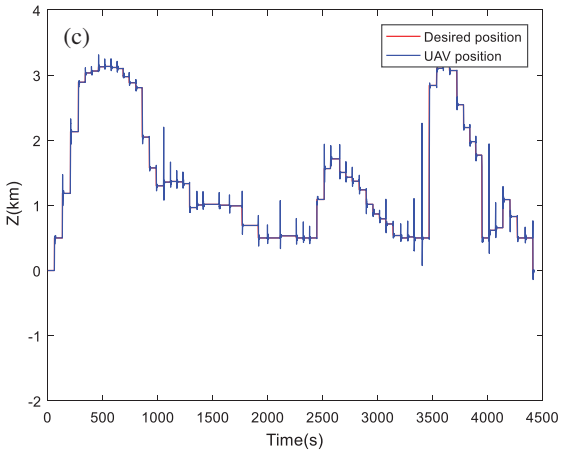



**Figure 9:** Position tracking based on MOPSO algorithm: (a) Tracking dynamics on X-axis; (b) Tracking dynamics on Y-axis; (c) Tracking dynamics on Z-axis

## 6 Conclusions

In this paper, a multi-objective multi-verse optimizer-based method has been proposed and successfully applied to solve the path planning problem of quadrotors UAV in a 3D dynamic environment. The path planning problem was formulated as a multi-objective optimization problem under operational constraints. The proposed planning approach aims to lead the drone to traverse a short and fast path in a dynamic environment without collision with the moving obstacles. An interactive graphical interface was developed under MATLAB/Simulink software environment to implement the proposed MOMVO-based path planning strategy. The demonstrative results and nonparametric statistical analyses in the sense of Friedman and the post-hoc tests show that the proposed MOMVO-based method is efficient and powerful compared to other reported algorithms. In comparison with MSSA, MOGWO, MOPSO, and NSGA-II optimizers, the main advantages of the proposed multi-verse algorithm are the remarkable simplicity of software implementation as well as the reduced number of its control parameters. The exploration/exploitation capabilities are superior to those of the other reported algorithms. Besides, the paired comparisons for two different optimization criteria showed that the MOMVO algorithm outperforms all the reported optimizers.

Future works deal with the real-world prototyping and experimentation of the proposed MOMVO-based planning approach using a real model of quadrotor available in our laboratory. The Parrot AR. Drone 2.0 kit associated with MATLAB/Simulink software will be used for the experimentations.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] Y. Fu, M. Ding, C. Zhou and H. Hu, "Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, pp. 1451–1465, 2013.

[2] R. R. Pitre, X. R. Li and R. Delbalzo, "UAV route planning for joint search and track missions: An information-value approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2551–2565, 2012.

[3] H. Chen, X. M. Wang and Y. Li, "A survey of autonomous control for UAV," in *Proc. of the 2009 Int. Conf. on Artificial Intelligence and Computational Intelligence*, Shanghai, China, pp. 267–271, 2009.

[4] M. Radmanesh, M. Kumar and A. Nemati, "Dynamic optimal UAV trajectory planning in the national airspace system via mixed integer linear programming, Proceedings of the Institution of Mechanical Engineers," *Part G: Journal of Aerospace Engineering*, vol. 230, no. 9, pp. 1668–1682, 2016.

[5] P. K. Das, H. S. Behera, P. K. Jena and B. K. Panigrahi, "Multi-robot path planning in a dynamic environment using improved gravitational search algorithm," *Journal of Electrical Systems and Information Technology*, vol. 3, no. 2, pp. 295–313, 2016.

[6] C. Huang, Y. Lan, Y. Liu, W. Zhou, H. Pei *et al.,* "A new dynamic path planning approach for unmanned aerial vehicles," *Complexity*, vol. 2018, no. 8420294, pp. 1–17, 2018.

[7] F. Ge, K. Li, Y. Han, W. Xu and Y. Wang, "Path planning of UAV for oilfield inspections in a three-dimensional dynamic environment with moving obstacles based on an improved pigeon-inspired optimization algorithm," *Applied Intelligence*, vol. 50, no. 1, pp. 2800–2817, 2020.

[8] B. Zhang and H. Duan, "Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 97–107, 2017.

[9] X. Zhao, Y. Zhang and B. Zhao, "Robust path planning for avoiding obstacles using time-environment dynamic map," *Measurement and Control*, vol. 53, no. 1–2, pp. 214–221, 2020.

[10] G. Tian, L. Zhang, X. Bai and B. Wang, "Real-time dynamic track planning of multi-UAV formation based on improved artificial bee colony algorithm," in *Proc. of the 37th Chinese Control Conf.*, Wuhan, China, pp. 10055–10060, 2018.

[11] S. Zajac and S. Huber, "Objectives and methods in multi-objective routing problems: A survey and classification scheme," *European Journal of Operational Research*, vol. 290, no. 1, pp. 1–25, 2020.

[12] R. Olaechea, D. Rayside, J. Guo and K. Czarnecki, "Comparison of exact and approximate multi-objective optimization for software product lines," in *Proc. of the 18th Int. Software Product Line Conf.*, Firenze, Italy, pp. 92–101, 2014.

[13] V. Rodriguez-Fernandez, C. Ramirez-Atencia and D. Camacho, "A multi-UAV mission planning videogame-based framework for player analysis," in *Proc. of the IEEE Congress on Evolutionary Computation*, Sendai, Japan, pp. 1490–1497, 2015.

[14] Y. Zhang, D.-W. Gong and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, no. 1, pp. 172–185, 2013.

[15] 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, no. 1, pp. 878–888, 2016.

[16] V. Skala, "A new approach to line-sphere and line-quadrics intersection detection and computation," in *Proc. of the 12th Int. Conf. on Numerical Analysis and Applied Mathematics*, Rhodes, Greece, pp. 1–4, 2014.

[17] D. A. G. Vieira, R. Adriano and J. A. de Vasconcelos, "Handling constraints as objectives in a multiobjective genetic based algorithm," *Journal of Microwaves and Optoelectronics*, vol. 2, no. 6, pp. 50–58, 2002.

[18] S. Mirjalili, S. M. Mirjalili and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 1, pp. 495–513, 2016.

[19] S. Mirjalili, P. Jangir, S. Z. Mirjalili, S. Saremi and I. N. Trivedi, "Optimization of problems with multiple objectives using the multi-verse optimization algorithm," *Knowledge-Based Systems*, vol. 134, no. 1, pp. 50–71, 2017.

[20] H. Deng, C.-H. Yeh and R. J. Willis, "Inter-company comparison using modified TOPSIS with objective weights," *Computers & Operations Research*, vol. 27, no. 10, pp. 963–973, 2000.

[21] A. Nagaty, S. Saeedi, C. Thibault, M. Seto and H. Li, "Control and navigation framework for quadrotor helicopters," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1–4, pp. 1–12, 2013.

[22] H. Elkholy and M. K. Habib, "Dynamic modeling and control techniques for a quadrotor," in *Unmanned Aerial Vehicles: Breakthroughs in Research and Practice*, Information Resources Management Association (Eds.), 1st ed., USA: IGI Global, pp. 20–66, 2019.

[23] S. Badr, O. Mehrez and A. E. Kabeel, "A design modification for a quadrotor UAV: Modeling, control and implementation," *Advanced Robotics*, vol. 33, no. 1, pp. 13–32, 2019.

[24] S. Abdelhay and A. Zakriti, "Modeling of a quadcopter trajectory tracking system using PID controller," *Procedia Manufacturing*, vol. 32, no. 1, pp. 564–571, 2019.

[25] A. Nagaty, S. Saeedi, C. Thibault, M. Seto and H. Li, "Control and navigation framework for quadrotor helicopters," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1, pp. 1–12, 2013.

[26] S. Bouallègue and K. BenKhoud, "Integral backstepping control prototyping for a quad tilt wing unmanned aerial vehicle," *International Review of Aerospace Engineering*, vol. 9, no. 5, pp. 152–161, 2016.

[27] N. BenAmmar, H. Rezk and S. Bouallègue, "Control strategy for a quadrotor based on a memetic shuffled frog leaping algorithm," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 4081–4100, 2021.

[28] R. Fessi and S. Bouallègue, "LQG controller design for a quadrotor UAV based on particle swarm optimization," *International Journal of Automation and Control*, vol. 13, no. 5, pp. 569–594, 2019.

[29] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris *et al.,* "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, no. 1, pp. 163–191, 2017.

[30] S. Mirjalili, S. Saremi, S. M. Mirjalili and L. S. Coelho, "Multiobjective grey wolf optimizer: A novel algorithm for multi-criterion optimization," *Expert Systems With Applications*, vol. 47, no. 1, pp. 106–119, 2016.

[31] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[32] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[33] E. Zitzler, K. Deb and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[34] S. Khalilpourazari, B. Naderi and S. Khalilpourazary, "Multi-objective stochastic fractal search: A powerful algorithm for solving complex multi-objective optimization problems," *Soft Computing*, vol. 24, no. 4, pp. 3037–3066, 2020.

[35] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca and V. G. Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[36] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[37] D. G. Pereira, A. Afonso and F. M. Medeiros, "Overview of Friedman's test and post-hoc analysis," *Communications in Statistics-Simulation and Computation*, vol. 44, no. 10, pp. 2636–2653, 2014.

**Appendix A: Quadrotor's model parameters**

| Symbol | Description | Value/unit |
|---|---|---|
| $b$ | Lift coefficient | $2.984 \times 10^{-5}\ N.s^2/rad^2$ |
| $d$ | Drag coefficient | $3.30 \times 10^{-7}\ N.s^2/rad^2$ |
| $m_Q$ | Mass | $0.5\ Kg$ |
| $l$ | Arm length | $0.50\ m$ |
| $J_r$ | Motor inertia | $2.8385 \times 10^{-5}\ Kg\,m^2$ |
| $I_x$, $I_y$, $I_z$ | Quadrotor inertia | 0.005, 0.005, 0.010 |
| $\kappa_{1,2,3}$ | Aerodynamic friction coefficients | 0.3729 |
| $\kappa_{4,5,6}$ | Translational drag coefficients | $5.56 \times 10^{-4}$ |
| $g$ | Acceleration of the gravity | $9.81\ m.s^{-2}$ |