DOI: https://doi.org/10.62985/j.huit_ojs.vol26.no2E.399

ENERGY-AWARE PATH PLANNING FOR A QUADCOPTER USING GENERALIZED PARTICLE SWARM OPTIMIZATION

Van Truong Hoang*

Faculty of Missile and Shipgun, Naval Academy, Khanh Hoa, Vietnam

*Email: *VanTruong.Hoang@alumni.uts.edu.au*

Received: 21 December 2025; Revised: 21 March 2026; Accepted: 2 April 2026

ABSTRACT

This paper presents an integrated framework for energy-aware path planning of a quadcopter unmanned aerial vehicle (UAV). The ultimate objective is to generate a trajectory that simultaneously satisfies path-length constraints, collision avoidance, and mission goals while accounting for vehicle energy consumption. The path-planning objectives are combined into a multi-objective cost function to discover optimal solutions. The generalized particle swarm optimization (GEPSO) is then applied to manage the overall cost, considering real-world applications. Simulation and experimental results exhibit that the proposed framework achieves smoother trajectories and significant reductions in overall energy expenditure compared to energy-ignored planning strategies.

Keywords: Path planning, Particle swarm optimization, Generalized PSO, Obstacle avoidance, Quadcopter.

1. INTRODUCTION

Unmanned aerial vehicles (UAVs), particularly quadcopters, have garnered significant attention in recent years due to their versatility and applicability in a wide range of civil and industrial missions, including surveillance, environmental monitoring, infrastructure inspection, and transportation. In many of these applications, key success involves path planning methods that create one or more optimal trajectories for UAVs to follow during their operation [1, 2]. Effective path planning for quadcopter UAV, therefore, remains a critical research challenge, especially in complex and constrained environments.

Classical path-planning approaches primarily focus on geometric feasibility, shortest-path generation, or collision avoidance. However, for battery-powered quadcopter UAVs, energy consumption is one of the most restrictive operational constraints. Ignoring energy-related factors can lead to infeasible trajectories, reduced mission endurance, or premature mission termination [3, 4]. Consequently, there is a growing need for energy-aware path-planning strategies that explicitly account for path length, power consumption, and safety while meeting the constraints of specific missions.

Metaheuristic optimization methods are extensively used in UAV path planning because they can effectively handle multi-objective and constrained optimization tasks. Various approaches, such as Genetic Algorithms (GA), Ant Colony Optimization (ACO), Teaching–Learning–Based Optimization (TLBO), and Particle Swarm Optimization (PSO), have been applied to these problems. These methods aim to ensure flyable, feasible, and efficient UAV trajectories. GA is capable of addressing mixed path-planning scenarios [5,6], while ACO performs well in reducing path length and avoiding collisions [7]. TLBO is attractive due to its simple structure, but it suffers from a slower convergence rate [8]. However, these algorithms do not always guarantee globally optimal solutions and often yield non-deterministic outcomes.

In three-dimensional (3D) path planning, each method encounters specific challenges, such as high computational cost and premature convergence in GA, slow convergence in ACO, and limited suitability of TLBO for highly complex environments [9].

Owing to its adjustable parameters and flexible framework, the PSO family has been widely explored for UAV path planning, with numerous variants proposed to enhance convergence behavior, solution quality, and coordination in both single- and multi-UAV scenarios [10–13]. These studies demonstrate the effectiveness of PSO-based methods in handling nonlinear, high-dimensional search spaces and in accommodating formation constraints and cooperative strategies. However, generating reliable and collision-free trajectories in densely obstructed environments remains a significant challenge. Particularly, the performance of PSO-based approaches is highly sensitive to parameter selection and algorithmic design, where inappropriate tuning may lead to premature convergence, loss of diversity, and degraded optimization performance [14, 15].

This paper builds upon our previous framework [16], which investigates path planning for quadcopter UAV formations using GEPSO, and extends it by incorporating energy-awareness inspired by recent advances in energy-aware and safe path planning for unmanned aircraft systems, as seen in the work of researchers in [17]. Unlike energy-ignored planning strategies, the proposed approach integrates energy consumption directly into the optimization process through a multi-objective cost function. This formulation simultaneously addresses trajectory smoothness, collision avoidance, mission feasibility, and energy efficiency, making it more suitable for the practical deployment of quadcopter UAV formations.

The main contribution of this paper is the incorporation of energy awareness into a path-planning framework for quadcopter UAVs using GEPSO. The proposed method formulates path-planning objectives into a unified multi-objective cost function and leverages GEPSO to efficiently search for optimal trajectories under real-world constraints. Our results demonstrate that the proposed framework achieves significant reductions in overall energy expenditure compared to conventional approaches that neglect energy considerations.

The remainder of this paper is organized as follows. Section II presents the problem formulation and energy-aware path-planning model. Section III describes the GEPSO-based optimization framework. Section IV discusses simulation settings and results. Finally, Section V concludes the paper and outlines future research directions.

2. PROBLEM STATEMENT

2.1. Path Planning for a Quadcopter

The motion of a quadcopter's center of gravity defines its flight trajectory. In a 3D environment, the vehicle's movement is characterized by six primary components, consisting of three translational and three rotational motions. A flight plan is generally composed of a set of sequential line segments connecting an origin point to a final destination, with waypoints established at the intersection of these segments. The Cartesian coordinate system ($Oxyz$) serves as the framework for defining both waypoints and vehicle movements. The state of the Unmanned Aerial Vehicle (UAV) is specified as $P(x, y, z, \theta, \psi)$, where (x, y, z) denotes its position, and θ and ψ represent the horizontal and vertical angles of rotation, respectively. However, because quadcopters can typically perform rotations without significant constraints, their state is commonly simplified to only the three translational coordinates.

The fundamental problem of trajectory planning for a UAV traveling between an initial state P_s and a final state P_f is to generate one or more continuous flight paths, denoted as $r(q)$, that connect these two states. This relationship can be expressed mathematically as follows:

$$P_s \xrightarrow{r(q)} P_f \tag{1}$$

where q is the path parameter.

For a UAV to fly from a location with start state $P_s(x_s, y_s, z_s)$ to a location with final state $P_f(x_f, y_f, z_f)$, Eq. (1) can be expressed as:

$$P_s(x_s, y_s, z_s) \xrightarrow{r(q)} P_f(x_f, y_f, z_f) \tag{2}$$

Nevertheless, the generated path may be infeasible, as a UAV cannot instantaneously alter its state at each waypoint. In practical applications, path planning is subject to numerous constraints, many of which are specific to UAV systems. Among these constraints, feasibility and safety are paramount. A trajectory is deemed feasible only if it complies with the vehicle's motion constraints. Safety is ensured through collision avoidance within a confined operational space. Additionally, further constraints may be imposed to satisfy a specific flight mission.

The symbol ⨅ is employed to denote constraints; consequently, the path planning problem can be formulated as follows [10]:

$$P_s(x_s, y_s, z_s) \xrightarrow{\bigsqcap r(q)} P_f(x_f, y_f, z_f) \tag{3}$$

2.2. Quadcopter Model

In this work, the quadcopter is selected because its vertical takeoff and landing and hovering capabilities make it particularly suitable for monitoring tasks in cluttered environments. Its dynamics are represented by nonlinear differential equations that describe the evolution of position and velocity, enabling trajectory tracking and control. The position vector determines the geometric path for navigation and obstacle avoidance, whereas the velocity vector and its derivatives ensure dynamic feasibility and smooth trajectory execution. In energy-aware path planning, velocity and acceleration directly influence power consumption, making them essential components of the optimization process [17].

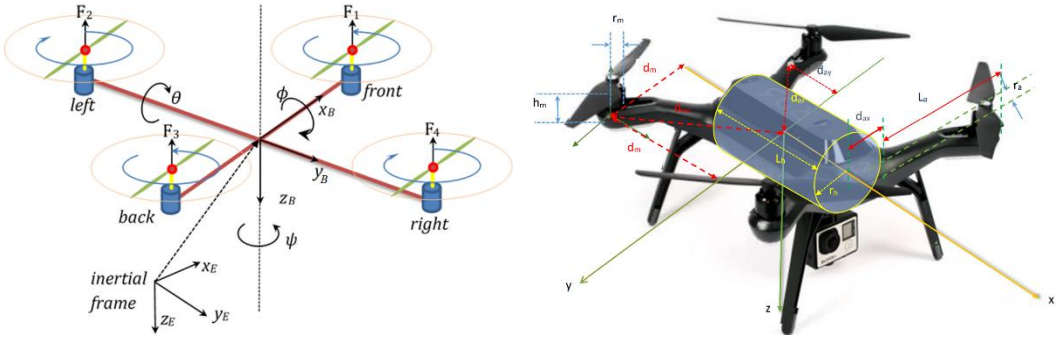

Fig. 1. Frames of reference (*left*) and a quadcopter model (*right*)

The quadcopter model is formulated with respect to two principal reference frames (Fig. 1): the inertial frame (x_E, y_E, z_E) and the body-fixed frame (x_B, y_B, z_B). The translational dynamics of the quadcopter expressed in the inertial frame are characterized by its position vector, $\xi = (x, y, z)^T$, and its velocity vector, $\dot{\xi} = (\dot{x}, \dot{y}, \dot{z})^T$. The attitude of the UAV is represented by the Euler angle vector $\Theta = (\varphi, \theta, \psi)^T$ together with the associated roll, pitch, and yaw rates $\dot{\Theta} = (\dot{\varphi}, \dot{\theta}, \dot{\psi})^T$. The angular velocity of the quadcopter in the inertial frame is denoted by $\omega = [p, q, r]^T$, that is,

$$\omega = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\varphi & c\theta s\varphi \\ 0 & -s\varphi & c\theta c\varphi \end{bmatrix} \dot{\Theta}, \tag{4}$$

where *sx* and *cx* represent sin(*x*) and cos(*x*), respectively. The transformation from the body frame to the Earth frame is subsequently defined by the following rotation matrix:

$$R = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi - s\psi s\varphi \\ s\psi c\theta & s\psi s\theta s\varphi + c\psi c\varphi & s\psi s\theta c\varphi - c\psi s\varphi \\ -s\theta & sc\theta\varphi & c\theta c\varphi \end{bmatrix}. \tag{5}$$

It is assumed that all reference trajectories and their first- and second-order derivatives are bounded. Additionally, the quadcopter's orientation is also restricted, with roll and pitch angles confined to $[-\pi/2, \pi/2]$ and yaw angle limited to $[-\pi, \pi]$.

2.3. Energy Consumption Model

For the quadcopter's energy consumption model (ECM), the model for the powertrain components and combine them as shown in Fig. 2. The components include the lithium-ion battery (LIB), the brushless direct current (BLDC) motors with attached rotors, and the electric speed controllers (ESCs).

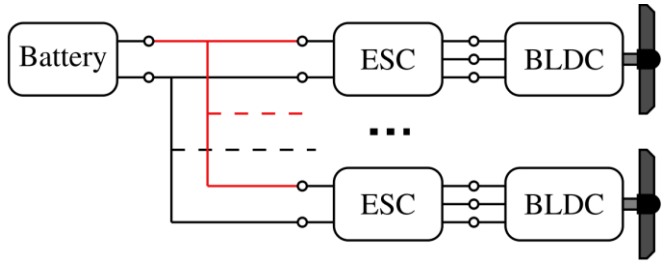

Fig. 2. Simplified power train of an electric-propelled UAV [14].

Inspired by [17], the quadcopter energy consumption is modeled by considering aerodynamic drag, thrust generation, and maneuvering effort. In this research, the entire trajectory is discretized into *M* segments, including the starting and landing points. For a discretized trajectory, the total energy consumption, E_{total}, is approximated as

$$E_{\text{total}} = \sum_{m=1}^{M} P_m \Delta t \tag{6}$$

where P_m is the instantaneous power at waypoint *m*, and Δt is the sampling interval.

The power consumption is modeled as

$$P_m = P_{\text{hover}} + \alpha \|\boldsymbol{v}_m\|^2 + \beta \|\boldsymbol{a}_m\|^2, \tag{7}$$

where P_{hover} is the power required for hovering, $\boldsymbol{v}_m$ is the velocity vector, $\boldsymbol{a}_m$ is the acceleration vector, α captures aerodynamic drag losses, β captures additional thrust and maneuvering effort. This formulation penalizes aggressive accelerations and high-speed motion, naturally encouraging smooth and energy-efficient trajectories.

The energy cost term used in the optimization is defined as

$$J_{\text{energy}} = {}^{E_{\text{total}}}/_{E_{\text{max}}}, \tag{8}$$

where E_{max} is the maximum allowable energy budget.

3. COST FUNCTION DESIGN

3.1. Path Length Cost

For a specified flight trajectory consisting of a total of N waypoints, the waypoint coordinates $W_j, j = 1..N$ can be arranged in the form as:

$$\{W_1, W_2, \dots, W_{N+1}\} = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_{N+1}, y_{N+1}, z_{N+1})\}. \quad (9)$$

The path-length cost, denoted by $\coprod_{\text{range}} r(q)$, is defined as the cumulative distance along the trajectory from the initial point to the terminal point and is given by:

$$\coprod\!\!\coprod_{\text{range}} r(q) = p_{\text{range}} \sum_{j=0}^{N+1} \left\| \overrightarrow{W_j, W_{j+1}} \right\|, \quad (10)$$

here, W_0 and W_{N+1} represent the starting and ending points of the trajectory segment, respectively, and $\|\cdot\|$ denotes the Euclidean norm. W_0 and W_{N+1} remain fixed, as all particles share identical initial and terminal position.

3.2. Collision Avoidance Cost

Let the operating environment contain K obstacles, collectively referred to as obstacles and denoted by $\{T_1, T_2, \dots, T_K\}$. Each obstacle is modeled as a circular region characterized by a center point C_k and an associated radius R_k, where the center specifies the obstacle location and the radius defines the extent of its influence.

For each path segment $\overrightarrow{W_j, W_{j+1}}$, the distance to an obstacle is evaluated using the distance between the obstacle center and the midpoint of the corresponding segment. To maintain safe operation, the UAV is permitted to traverse only regions lying outside the obstacle boundary. When a flight segment intersects the cylindrical region defined by (C_k, R_k), the obstacle avoidance cost is evaluated. The procedure for computing the corresponding hazard index is outlined as follows.

(i) For each obstacle T_k, compute the distance d_k from C_k to the projection of the flight path segment $\overrightarrow{W_j, W_{j+1}}$.

(ii) Compare d_k with the obstacle radius R_k. When $d_k \geq R_k$, the safety cost associated with the kth obstacle for the path segment ($\overrightarrow{W_j, W_{j+1}}$), is zero, that is,

$$\coprod\!\!\coprod_{\text{safe},k} (\overrightarrow{W_j, W_{j+1}}) = 0 \quad (11)$$

Otherwise, when $d_k < R_k$, proceed to the next step.

(iii) Determine the length of the segment projection covered by the kth obstacle, denoted by l_k. The safety cost of the segment with respect to the kth obstacle is then given by:

$$\coprod\!\!\coprod_{\text{safe},k} (\overrightarrow{W_j, W_{j+1}}) = \begin{cases} R_k l_k & \text{if } d_k \leq l_k \\ R_k l_k / d_k & \text{if } d_k > l_k \end{cases} \quad (12)$$

(iv) The total obstacle avoidance cost over the entire flight path is expressed as:

$$\coprod\!\!\coprod_{\text{safe}} r(q) = p_{\text{safe}} \sum_{k=1}^{K} \sum_{j=0}^{N+1} s_k \coprod\!\!\coprod_{\text{safe},k} (\overrightarrow{W_j, W_{j+1}}) \quad (13)$$

where s_k represents the hazard severity associated with the k, $p_{\text{safe}} > 0$ is the penalty coefficient for obstacle violations.

3.3. Task Cost

For UAV operations, maintaining flight within a prescribed altitude range is a key factor in enhancing mission effectiveness. When operating over terrain, a quadcopter must maintain sufficient clearance above the ground to avoid collision. At the same time, to improve the performance of onboard sensors for ground monitoring applications, the UAV must travel within a limited distance from the surface. By jointly considering these two constraints, the quadcopter must fly within a maximum and minimum allowable altitudes $z_{\max}$ and $z_{\min}$ to increase the probability of effective observation.

Denoting Q as the total number of monitoring points taken, and the terrain elevation at the point qth as T_q. The relative altitude of the UAV at this point is defined as z_q. The mission cost can be computed accordingly:

$$\coprod_{\text{task}} r(q) = p_{\text{task}} \sum_{q=1}^{Q} z_q$$

$$z_q = \begin{cases} z_q - z_{\max}, & \text{when } z_q > z_{\max} \\ 0, & \text{when } z_{\min} \le z_q \le z_{\max} \\ z_{\min} - z_q, & \text{when } 0 \le z_q \le z_{\min} \\ \infty, & \text{when } z_q \le 0 \end{cases}, \tag{14}$$

where $p_{\text{task}} > 0$ is the penalty coefficient of the task cost.

3.4. Overall Cost Function

Setting up a reasonable cost function is extremely important when applying optimization algorithms to solve practical problems. The cost function usually includes at least a length constraint and a constraint on avoiding obstacles that threaten the safety of the established flight path. In monitoring missions, other constraints that the UAVs must meet include limitations on flight altitude and maintenance of distance to monitoring objects. Altogether, we can combine the individual costs into a common function, as follows:

$$\coprod r(q) = \coprod_{\text{range}} r(q) + \coprod_{\text{safe}} r(q) + \coprod_{\text{task}} r(q) \tag{15}$$

where $\coprod_{\text{range}} r(q)$, $\coprod_{\text{safe}} r(q)$, $\coprod_{\text{alt}} r(q)$, and $\coprod_{\text{mission}} r(q)$ are the specified criteria for path length, obstacle avoidance, and distance to monitoring surface.

4. PATH PLANNING ALGORITHM

4.1. Particle swarm optimization

Particle swarm optimization (PSO) is an evolutionary computation technique introduced [15], inspired by social and collective behavior. The algorithm employs a population of particles that move through the problem's search space with associated velocities. At each iteration, particle velocities are stochastically updated based on both the particle's personal best position and the best position found in its neighborhood, guiding the swarm toward optimal or near-optimal solutions. PSO is characterized by simple implementation, high efficiency, and effective maintenance of swarm diversity [7].

In PSO, each feasible solution corresponds to a particle whose state is defined by its position vector $x_i \in R^n$ and velocity vector $v_i \in R^n$, as follows:

$$x_i^t = x_i^{t-1} + v_i^t \tag{16}$$

Each particle relies on its own experience and the collective performance of the swarm. To reflect the varying influence of these factors, random weights are applied, leading to the velocity update rule:

$$v_i^t = \omega v_i^{t-1} + \varphi_1 r_1 \left[p_i^{t-1} - x_i^{t-1}\right] + \varphi_2 r_2 \left[p_g^{t-1} - x_i^{t-1}\right] \tag{17}$$

Where p_i and p_g represent the local-best and global-best positions associated with particle i; ω represents the inertial weight; $\varphi_1, \varphi_2 > 0$ are the gain coefficients, and $r_1, r_2 \in [0, 1]$ are uniformly distributed random scalars.

4.2. Generalized Particle Swarm Optimization

Owing to its advantageous characteristics, PSO has proven to be an effective approach for multi-objective optimization problems. Numerous variants, including discrete PSO, θ-PSO, and hybrid PSO, have been developed to improve its performance. These extensions, however, retain the original position-velocity framework and primarily modify the update equations without altering the algorithmic structure.

To address these limitations, the Generalized PSO (GEPSO) was introduced [16,19]. GEPSO enhances particle interactions to accelerate swarm convergence and incorporates random velocity components to improve exploration in complex search spaces. While the position update remains unchanged, the velocity update is extended as:

$$\begin{aligned} v_i^t = \psi\{a_1 v_i^{t-1} + a_2\left[p_i^{t-1} - x_i^{t-1}\right] + a_3\left[p_g^{t-1} - x_i^{t-1}\right] + a_4\left[p_{rand}^{t-1} - x_i^{t-1}\right] \\ + a_5 v_{rand}^{t-1}\} \end{aligned} \tag{18}$$

where ψ is the constriction factor, $a_1 = \omega_1, a_2 = \omega_2\varphi_1 r_1^i, a_3 = \omega_3\alpha_1\varphi_2 r_2^i$, $a_4 = \omega_4\alpha_2\varphi_3 r_3^i$, and $a_5 = \omega_5\alpha_3\varphi_4 r_4^i$, $a_1, \cdots, a_3$ are regulated probability terms, $\omega_1, \cdots, \omega_5$ denote inertia weights, and $\varphi_1, \cdots, \varphi_4$ are acceleration coefficients.

The constriction factor is updated as:

$$\psi = \frac{2}{|2 - (\varphi_2 - \varphi_3)^2 - 5(\varphi_2 + \varphi_3)|} \tag{19}$$

The inertia weight ω_1 is dynamically adjusted at each iteration according to:

$$\begin{aligned} \omega_1 &= \min\{\omega_{\min}, \omega_k\}, \\ \omega_k &= \omega_1^{t-1} - \left\{\frac{\omega_{\max} - \omega_{\min}}{N} i\left[f\left(p_g^{t-1}\right) - f\left(p_g^{t-2}\right)\right]\right\}, \end{aligned} \tag{20}$$

where $\omega_{\max}, \omega_{\min}$ are the maximum and minimum inertia weights; i and N denote the current and total iteration counts, respectively; $f(\cdot)$ is the previous function evaluation. The initial particle positions and velocities are: $x_i^0(x_k) = x_{k,min} + \text{rand} x_{k,max}$, and $v_i^0(x_k) = x_i^0(x_k) + \gamma_k$; with $x_{k,min}$ and $x_{k,max}$ are the lower and upper bounds of the kth dimension.

4.3. Energy-Aware Path Planning Generation

```
01 foreach i < (number_of_iterations) do
02 | foreach particle do
03 | | Initialize particles;
04 | foreach particle i do
05 | | Compute cost value ∐_range r(q), ∐_safe r(q), ∐_task r(q);      /* Eq. (10, 13, 14) */
06 | | If current cost < previous costs then
07 | | | Update new value for p_i^t;
08 | | End If
```

09 | | Select a random value for p_i^t;
10 | | Calculate the value of global best p_g^t;
11 | | Select a random value for v;
12 | | Select a random particle's best value p;
13 | | Update new result for ω; /* Eq. (20) */
14 | | Compute particle's velocity; /* Eq. (18) */
15 | | Apply the velocity constriction;
16 | | Update new position; /* Eq. (16) */
17 | | Apply the position constriction;
18 | | Check the collision avoidance; /* Eq. (11) */
19 | | Evaluate flight paths based on global and safety cost functions; /* Eq. (15) */
20 | | Update new results for p_i^t;
21 | **End**
22 | Update p_g^t and $\coprod_{\text{safe}}$;
23 **End**
24 Save p_g^t and $\coprod_{\text{safe}}$ to the global memory;
25 Achieved the optimal path at the last iteration;
26 Generate trajectory for UAV P^*.

Fig. 3. GEPSO path planning algorithm in pseudo-code

The preparation phase starts once the quadcopter is assigned to a monitoring mission. This phase involves defining the operational area to identify terrain and obstacles, and specifying the allowable flight altitude range and the required monitoring distance to the target. The operational space is modeled as a rectangular region defined by four GPS latitude-longitude coordinates $\Gamma_i = \{\varphi_i, \lambda_i\}$, $i = 1 \cdots 4$, together with a maximum altitude H_{max}. Terrain and obstacle data within this region are obtained from a 3D satellite map. Each obstacle is represented as a vertical cylinder with ground-center coordinates $C_k\ (\varphi_k, \lambda_k)$, $k = 1 \cdots K$, and an avoidance radius r_p. Based on the mission, the start (P_s) and final (P_f) waypoints are then determined. In addition, the safe radius and kinematic constraints of the quadcopter are specified. Finally, all preparation data are organized, encoded, and stored in an initialization file (Init_file).

The initialization phase is executed after all input data for the path-planning problem have been obtained. This phase includes setting up the operational space, defining the number of waypoints, initializing the GEPSO algorithm parameters, and generating a random initial trajectory connecting the take-off point P_s and the landing point P_f. In this context, initialization refers to loading the Init_file into global memory.

Initial values of the mission variables are defined based on the cost functions introduced in Section 3. Penalty coefficients for the individual cost terms are predetermined before their integration into the multi-objective formulation (15), and the initial value of the overall cost function $\coprod r(q)$ is set to infinity.

GEPSO parameters may be manually specified based on the problem requirements. Key settings, such as the swarm size and the maximum number of iterations N, must be selected beforehand, along with algorithmic parameters $\omega, \alpha, \varphi,$ and ψ. At this stage, the application of the GEPSO algorithm to generate flight paths can be expressed in pseudocode, as in Fig. 3.

After creating the optimal path, it is necessary to apply the energy consumption computation with its algorithm shown in Fig. 4. The input of the algorithm is the breakdown of the generated trajectory of the quadcopter P^* into K segments, i.e., $P^* = \{p_0, p_1, \cdots, p_K\}$.

01 Initialize $E_{\text{total}} = 0$, P_0, $E_{\max}$
02 **foreach** $k, k = 1, \cdots, K$ **do**
03 | Compute velocity $\boldsymbol{v}_k = (p_k - p_{k-1})/\Delta t$;
05 | Compute acceleration $\boldsymbol{a}_k = (\boldsymbol{v}_k - \boldsymbol{v}_{k-1})/\Delta t$;
06 | Compute power P_k; /* Eq. (7) */
07 | Update new value for $E_{\text{total},k} = E_{\text{total},k-1} + P_k \Delta t$;
08 **End**
09 Normalize energy cost J_{energy}; /* Eq. (8) */
10 Return J_{energy};

Fig. 4. Energy consumption computation algorithm

5. RESULTS

5.1. Experimental Set Up

The monitoring mission aims to generate an optimal and safe UAV flight trajectory in a complex environment. The mission is conducted using a 3DR Solo UAV, a laptop-based ground control station, and the Mission Planner (MP) software. MP supports mission planning, real-time monitoring, and UAV control, while Google Satellite Maps are used to acquire environmental information and identify obstacles, which are subsequently marked on the map. All the workspace records acquisition results mentioned above are shown in Fig. 5.

Table 1. Gepso parameters selected

Symbol	*Value*	*Symbol*	*Value*	*Symbol*	*Value*	*Symbol*	*Value*
w_1, w_2	0.5	w_5	0.9	a_2, a_3	2.0	j_2, j_4	2.5
w_3, w_4	0.8	a_1	4.5	j_1, j_3	2.0	y	0.9

The task involves monitoring a farm of fruit trees within a 30 m–high rectangular operational space defined by the GPS coordinates {12.233106, 109.114506} and {12.233563, 109.115220}. The start and final points are located at {12.233194, 109.114557} and {12.233411, 109.115187}. The area contains dense obstacles (M = 29) of varying sizes, resulting in a complex environment (Fig. 3). The UAVs fly within the relative distance range to the obstacles, limited by [0,5, 4] m.

The proposed algorithm is configured with 100 individuals and 100 iterations, and the GEPSO parameters are initialized as summarized in Table 1.

Fig. 5. Working space acquisition

5.2. Experiments and Comparisons

The three-dimensional UAV trajectory presented in Figs. 6 demonstrates that the planned path reliably reaches the target under harsh operational conditions. The trajectories further verify path flyability, feasibility, and safety.

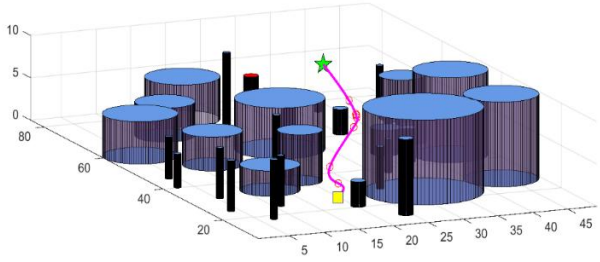

Fig. 6. 3D-path generated for the quadcopter

Fig. 7. 3D-path generated for the quadcopter

The generated trajectory is then executed in Mission Planner using the corresponding satellite map, as shown in Fig. 7. The results confirm that the planned path is smooth and flyable, and that the UAV successfully avoids densely distributed obstacles in the complex environment, thereby validating the practical feasibility of the proposed approach.

By calibrating the algorithm in Fig. 4, simulation results show that the proposed method achieves a 5–7% reduction in energy consumption compared to planning strategies that neglect energy considerations, when all constraints are satisfied. This improvement is consistent with recent findings on energy-aware UAV path planning, which emphasize the importance of explicitly incorporating energy models into the optimization process to enhance operational efficiency and safety [17].

For quantitative validation, GEPSO is compared with PSO, θ-PSO [1], and TLBO [7] based on the average results of 30 independent runs, as illustrated in Fig. 9. The convergence curves indicate that GEPSO achieves the lowest cost value and converges more rapidly and stably than the other compared algorithms.

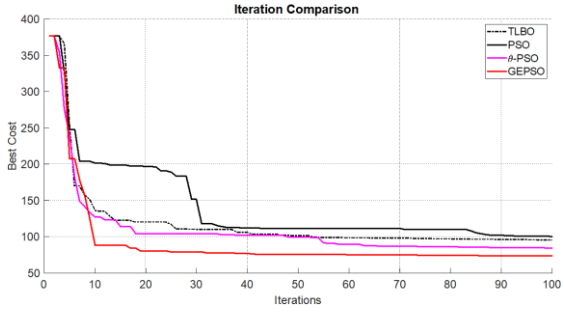

Fig. 8. Convergence comparison

These observations are further supported by the results summarized in Table 2, which lists the minimum and maximum cost values, as well as the number of iterations of convergence. Among the four compared methods, the superior performance can be attributed to the design of GEPSO, which strengthens inter-particle interactions and enhances information sharing within the swarm, thereby accelerating convergence [19]. In addition, the incorporation of random velocity components improves the exploration capability in complex search spaces, reducing the risk of premature convergence.

Table 2. Gepso performance relative to other path planning algorithms

Algorithm	*Min cost*	*Max cost*	*Iteration*
TLBO	98.47	356	57
PSO	100.09	356	92
θ-PSO	85.73	356	87
GEPSO	76.03	356	63

6. CONCLUSION

This paper has presented an energy-aware path-planning framework for a quadcopter UAV based on generalized particle swarm optimization. By incorporating energy consumption into the objective function, the proposed approach has generated safe, smooth, and energy-efficient trajectories. The results and comparisons have validated the effectiveness of the framework and demonstrate its suitability for real-world UAV applications. Future work will focus on experimental validation using real UAV platforms, dynamic obstacle avoidance, and integration with onboard power management systems.

REFERENCES

[1] V. T. Hoang, M. D. Phung, T. H. Dinh, and Q. P. Ha, "System architecture for real-time surface inspection using multiple UAVs," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2925-2936, 2019, https://doi.org/10.1109/JSYST.2019.2922290

[2] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey", *Robotics Auto. Syst.*, vol. 86, pp. 13–28, Dec. 2016, https://doi.org/10.1016/j.robot.2016.08.001

[3] J. H. Choi, D. S. Seo, S. H. Bae, Y. C. An, E. J. Kim, J. W. Pyo and T. Y. Kuc, "A Semantic Energy-Aware Ontological Framework for Adaptive Task Planning and Allocation in Intelligent Mobile Systems," *Electronics*, vol. 14, no. 18, pp. 3647, 2025, https://doi.org/10.3390/electronics14183647

[4] H. Çinar, D. Ignatye, A. Zolotas, "A comprehensive review and future challenges of energy-aware path planning for small unmanned aerial vehicles with hydrogen-powered hybrid propulsion," *The Aeronautical Journal,* vol. 129, no. 1336, pp.1468-1493, 2025, https://doi.org/10.1017/aer.2025.11.

[5] V. Roberge, M. Tarbouchi, G. Labonté, "Fast genetic algorithm path planner for fixed-wing military UAV using GPU," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2105-2117, 2018, https://doi.org/10.1109/taes.2018.2807558.

[6] J. Zheng, M.Ding, L.Sun, H. Liu, "Distributed stochastic algorithm based on enhanced genetic algorithm for path planning of multi-UAV cooperative area search," *IEEE*

Transactions on Intelligent Transportation Systems, vol. 24, no. 8, pp.8290-8303, 2023, https://doi.org/10.1109/TITS.2023.3258482.

[7] G. Li, Y. Li, "UAV path planning based on improved ant colony algorithm," in *Second International Conference on Algorithms, Microchips, and Network Applications (AMNA 2023)*, vol. 12635, pp. 59-63, May 2023, https://doi.org/10.1117/12.2678893.

[8] V. T. Hoang, M. D. Phung, "Enhanced Teaching-Learning-Based Optimization for 3D Path Planning of Multicopter UAVs," in *International Conference on Advanced Mechanical Engineering, Automation and Sustainable Development*, pp. 743-753, 2021, Cham: Springer International Publishing, https://doi.org/10.1007/978-3-030-99666-6_107.

[9] A. A. Saadi, A. Soukane, Y. Meraihi, G. A. Benmessaoud, S. Mirjalili, C A. Ramdane, "UAV path planning using optimization approaches: A survey," *Archives of Computational Methods in Engineering*, vol. 29, no. 6, pp. 4233-4284, 2022, https://doi.org/10.1007/s11831-022-09742-7.

[10] V. T. Hoang, M. D. Phung, T. H. Dinh, and Q. Ha, "Angle-encoded swarm optimization for uav formation path planning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5239-5244, Oct 2018, https://doi.org/10.1109/IROS.2018.8593930.

[11] S. Shao, Y. Peng, C. He, Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA transactions*, vol. 97, pp. 415-430, 2020, https://doi.org/10.1016/j.isatra.2019.08.018.

[12] C. Li, Q. Zhao and C. Che, "3D Flight Path Planning for UAV Based on Improved Particle Swarm Optimization Algorithm," *IEEE Access*, vol. 13, pp. 36637-36646, 2025, https://doi.org/10.1109/ACCESS.2025.3543175.

[13] W. He, X. Qi, L. Liu, "A novel hybrid particle swarm optimization for multi-UAV cooperate path planning," *Applied Intelligence*, vol. 51, pp. 7350- 7364, 2021, https://doi.org/10.1007/s10489-020-02082-8.

[14] M. Jones, S. Djahel, K. Welsh, "Path planning for unmanned aerial vehicles with environment complexity considerations: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1- 39, 2023, https://doi.org/10.1145/3570723.

[15] L. Zhai, H. Wu, L. Lai and Z. Gao, "Intelligent Optimization Algorithms for Multi-UAV Path Planning: A Comprehensive Review," *IEEE Access*, vol. 13, pp. 101106-101130, 2025, https://doi.org/10.1109/ACCESS.2025.3577598.

[16] V. T. Hoang, "Path Planning for Multi-Copter UAV Formation Employing a Generalized Particle Swarm Optimization," in *7th Vietnam International Conference and Exhibition on Control and Automation (VCCA-2024)*, pp. 35-41, 2024, https://doi.org/10.48550/arXiv.2501.05770.

[17] S. Gasche, C. Kallies, A. Himmel and, R. Findeisen, "Energy aware and safe path planning for unmanned aircraft systems," *arXiv preprint arXiv:2504.03271*, 2025, https://doi.org/10.48550/arXiv.2504.03271.

[18] J. Kennedy, R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942-1948. IEEE, 1995, https://doi.org/10.1109/ICNN.1995.488968.

[19] S. Davoud, M. Ellips, S. Mostafa, A. Hossein, "GEPSO: A new generalized particle swarm optimization algorithm," *Mathematics and Computers in Simulation*, vol. 179, pp. 194-212, 2021, https://doi.org/10.1016/j.matcom.2020.08.013.