IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 1, 2020, accepted September 13, 2020, date of publication September 25, 2020,
date of current version October 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3026666

Optimized Path-Planning in Continuous Spaces for Unmanned Aerial Vehicles Using Meta-Heuristics

GEOVANNI FLORES-CABALLERO[1],
ALEJANDRO RODRÍGUEZ-MOLINA[2], **(Member, IEEE),**
MARIO ALDAPE-PÉREZ[1], **AND**
MIGUEL GABRIEL VILLARREAL-CERVANTES[1], **(Member, IEEE)**
[1]Postgraduate Department, Instituto Politécnico Nacional, CIDETEC, Mexico City 07700, Mexico
[2]Tecnológico Nacional de México/IT de Tlalnepantla, Research and Postgraduate Division, Estado de México 54070, Mexico

Corresponding author: Alejandro Rodríguez-Molina (alejandro.rm@tlalnepantla.tecnm.mx)

This work was supported in part by the Secretaría de Investigación y Posgrado (SIP), and in part by the Comisión de Operación y Fomento de Actividades Académicas (COFAA) of the Instituto Politécnico Nacional (IPN).

ABSTRACT This work presents a novel path-planning approach for Unmanned Aerial Vehicles (UAVs) in continuous 3D environments. This proposal aims to minimize the path length while avoiding collisions through the suitable adjusting of control points (the points that take the UAV from a start position to a target location). The above is stated as a constrained global optimization problem. This problem considers the overall length of the path as the single objective function. Regarding the problem constraints, they are related to the collision of the obstacles with the 3D shape of a path. The assignment of the path shape is also proposed in this work to streamline the planning process. Due to the optimization problem features (high nonlinearity, multimodality, non-differentiability, and the lack of an initial guess solution), a constraints-handling mechanism is used in meta-heuristics to find suitable optimized paths. Also, an enhanced path-search mechanism is included in these algorithms to deal with complex planning scenarios. The enhanced mechanism incorporates a path computed by a variant of the A-Star method (the Pruned A-Star) in the first set of candidate solutions of the meta-heuristics. The proposed approach is tested through six complex scenarios. Moreover, the performance of three well-known meta-heuristics, Differential Evolution (DE), Particle Swarm Optimization (PSO), and the Genetic Algorithm (GA), is studied to find a potential candidate to solve the path-planning problem. In this way, the paths found by DE show outstanding performance. The paths obtained by the Pruned A-Star technique are adopted as a point of comparison to determine the advantages and drawbacks of the proposal.

INDEX TERMS Path-planning, optimization problem, aerial vehicles, meta-heuristics, continuous spaces.

I. INTRODUCTION

Autonomy is a highly desirable feature of robots because it endows them with skills to perform complex tasks with minimal human intervention [1].

In the context of mobile robotics, motion planning is one of the most challenging problems due to the large number of issues that need to be addressed. The issues related to motion planning of mobile robots include environment perception, localization, map building, recognition, path-planning, and motion control [2].

The motion planning is used to determine how a robot must move to perform a task successfully. In perception, the mobile robot processes environmental sensor data [2], so threats, obstacles, or agents can be identified. In the case of localization and map building, they determine the current position of the mobile robot regarding its environment [2]. According to a specific observed agent's actions or motion effects, the purpose of recognition is to deduce the agent's behavior [3]. The path-planning involves searching for an

The associate editor coordinating the review of this manuscript and approving it for publication was Bijoy Chand Chand Chatterjee.

optimized collision-free path from a current location to a target goal [2]. Finally, motion control solves various control problems to govern the robot movements (given by the actuators and stabilizers) in such a way that it successfully follows the optimized path by compensating the robot dynamics, uncertainties, and disturbances [4].

By themselves, the above issues conform to full research areas. Among them, the path-planning is one of the most attractive due to the notable evolution of mobile robots, which can navigate through the most diverse spaces [5]; the extent of their use for a wide variety of tasks beyond the industrial field [6]; and the increasing complexity of the operation environments [7], e.g., regarding a more significant number of agents and obstacles, and their different arrangements, geometries, and behaviors. Therefore, this work is focused on this particular problem.

In a first analogy, named as the *the piano mover's problem* [8], the path-planning issue is compared to the problem of moving furniture (with complex geometries such as a big piano) from one place to another inside a furnished house, where the information about the objects (locations and geometries) is wholly known.

A more accurate definition is found in [9], which indicates that the path-planning problem is to determine the suitable intermediate configurations (e.g., positions and orientations) that a robot should reach to move effectively (e.g., traveling the shortest possible distance) between two locations inside a workspace while avoiding threats or obstacles [10].

When a particular kind of mobile robot is pointed out, the features of the associated path-planning problem are bounded, and its relevance is highlighted. The above is the case of Unmanned Aerial Vehicles (UAVs).

The UAVs are mobile robots designed to operate in the air without human actors aboard. They have a great set of features and advantages, including a relatively low cost and outstanding maneuverability [11]. The UAVs are especially useful to perform tasks such as object manipulation, surveillance, transportation, search and rescue, mapping, and monitoring [10]. For this reason, the application fields of the UAVs are increasing daily. Nowadays, there is a wide variety of areas that take advantage of these robots. Many application examples are found in the military [12], construction [13], research [14], health [15], and service [16] contexts.

Unfortunately, the most critical drawback of the UAVs is related to their energetic limitations concerning the onboard battery capacity. In this way, the effectiveness of the path-planing turns critical. Moreover, the path-planning problems with UAVs are related to 3D spaces in the vast majority of the cases, which make them harder when compared to the path-planning in the plane (e.g., for wheeled mobile robots). In many cases, the 3D spaces include a significant number of threats or obstacles with different geometries and configurations (e.g., the office furniture in an indoor environment), which further complicates the search for a suitable route.

Throughout the decades, several methods have been proposed to solve the path-planning problem in general for 3D environments, which are also valid for UAV applications. Those methods can be classified into five categories according to [17]: the sampling-based, node-based, mathematical-model-based, bio-inspired, and multi-fusion-based algorithms. The latter is referred to the use of two or more of the algorithms mentioned above to obtain an improved path but taking into consideration the combined advantages and drawbacks of them (these are discussed next).

The sampling-based algorithms require a mathematical representation of the workspace. Then, these methods sample nodes or cells, typically in a stochastic way, until a feasible path is generated [18]. An example of these methods is observed in [19], where the Random Tree (RT) algorithm and three proposed variations are used to successfully sample feasible paths from a graph in such a way they avoid collisions between the UAV and moving obstacles. The graph is generated with nodes representing positions in the discretized workspace and edges as collision-free connections of two nodes. Another work based on RTs for the path-planning with UAVs is found in [20]. In that research, a Biased Sampling Potentially Guided Intelligent Bidirectional Rapidly-exploring Random Tree Star (BPIB-RRT-Star) method is proposed to adjust the sampling space flexibly and, in consequence, improves the convergence rate.

On the other hand, the work in [21] proposes two sampling methods to solve the path-planning problem for a fixed-wing UAV stated as a variation of the traveling salesman problem for finite discrete graphs. A different sampling-based approach is found in [22]. That work presents a path-planning framework with Smart Exploration and Exploitation (Sampling-SEE). The exploration in the Sampling-SEE samples path alternatives uniformly in the search space, while the exploitation enhances those paths by a biased sampling of their surrounding regions. The application of the framework can be extended to the UAVs.

In the node-based algorithms, the workspace is conceptualized as a graph or a grid, and the feasible path is obtained using approaches similar to the well-known Dijkstra algorithm [23]. Within this category, the work in [24] describes an improved variant of the A-Star algorithm that allows the online path-planning for UAVs. This improvement is based on the partition of the problem into a series of independent grids to decrease the computational burden. Each grid includes a reduced number of cells that consider the threats and other valuable information of the 3D environment. Then, the A-Star algorithm performs local searches for feasible paths. The optimization of the traditional A-Star performance is also addressed in [25] by the use of a Field Programmable Gate Array (FPGA) system. That proposal accelerates the convergence of the A-Star method to achieve real-time path-planning performance. The research in [26] also uses the A-Star algorithm in the path-planning for UAVs that flight around restricted areas. In this case, the graph required by A-Star is built based on the information of the Global Position System (GPS).

The node-based approach is also addressed in [27], where a variant of the Dijkstra algorithm is used in the path-planning of a fixed-wing UAV. A visibility graph defines the workspace, and its nodes are close to the vertexes of polygons that represent threats or obstacles. Hence, the Dijkstra algorithm can find the shortest route, considering the terrain elevation.

In all the above methods, a discretization of the workspace is required. The above implies a trade-off between the computational complexity and path-planning accuracy. The larger the size, the less precision and computational resources needed for planning, and vice versa. Moreover, an actual optimal path cannot be obtained with a discretization (an infinite graph or grid would be necessary).

In the case of the mathematical-model-based algorithms, a dynamic mathematical programming problem is stated considering constraints regarding the environment and robot dynamics. The feasible path is obtained as the solution to such a problem by typically using classical optimization approaches. In [28] for example, the dynamics of a quad-rotor and its associated Artificial Potential Field (APF) (this models the attraction of the robot to the goal and its repulsion to threat regions) are used to state an unconstrained optimization problem, which is in turn transformed into an optimal control problem. Then, the solution to this problem includes the control actions that the UAV must take to track the optimized path. The path-planning problem for UAVs is also addressed as an optimal control one in [29]. The non-convexities of the problem are approximated by a series of sequential convex programming problems to obtain more stable solutions in a faster way.

Other optimal control problems related to path-planning for UAVs can be found in [30], [31]. In [30], the problem is stated and solved to generate different optimal paths that satisfy diverse payloads and oscillation angles in a quadrotor with a suspended cable load. On the other side, the work in [31] establishes an optimal control problem based on the Dubin Traveling Salesman Problem that considers the reaching and exploration of particular regions of the flight space for surveillance missions.

The mathematical-model-based methods do not take into account the entire environment or mobile features. The behavior of these elements is dynamically modeled as time-variant systems subject to kinematic and dynamic constraints. A crucial drawback of these methods is the requirement of an accurate model, which is critically affected by the environmental conditions for UAVs, including uncertainties (e.g., discrepancies in model parameters), disturbances (e.g., gusts of wind), and unmodeled dynamics. Also, the formulation of the associated mathematical programming problems is complex, and the computational burden required to find a solution increases proportionally. On the other hand, the classical optimization approaches tend to stagnate at local minima when the above problems are very complex.

On the other hand, bio-inspired algorithms model the behavior of natural phenomena to solve very complex problems for which the methods mentioned above are not successful enough. In this way, the path-planning problem is stated as an optimization problem considering environmental information. The solution to this problem includes the optimized feasible path.

Among bio-inspired techniques, meta-heuristics have gained popularity due to their attractive and valuable features, including high performance, universality, and simplicity [32], [33]. These techniques are typically inspired by the operation of natural systems and encompass full research fields like evolutionary computation [34] and swarm intelligence [35].

There are several path-planning approaches for UAVs that adopt these kinds of methods. In [36], the Wolf Pack Search (WPS) algorithm is modified with the concepts of Genetic Algorithms (GAs) to optimize the points of the B-splines that conform to the path. The workspace is discretized by uniformly dispersed waypoints marked as feasible or unfeasible. WPS solves the established unconstrained single-objective problem that considers both the total path length, obstacle avoidance, the height, and the smoothness, to generate optimized paths. The research in [37] compares the performance of the GA and the Particle Swarm Optimization (PSO) in the path-planning for UAVs. The related unconstrained optimization problem considers the path length, the threat evasion, the obstacle collisions, the altitude, the fuel, and the power in a single objective. Then, a fixed number of points are set by the GA and PSO in a discretized workspace to generate the optimized path. A similar approach is found in [38]. In that work, the Ant Colony Optimization (ACO) algorithm distributes the path points in a workspace represented by a finite 3D grid whose cells contain information about obstacle occupancy.

More examples of the bio-inspired path-planning approach are observed in [39], [40]. In [39], an improved PSO that incorporates a chaos-based Logistic map, linear-varying parameters, and a mutation strategy is adopted to find UAV paths that minimize a cost function, which considers the length, environmental constraints, and collision avoidance. The work in [40] improves the PSO by incorporating an APF that enhances the evaluation of the paths cost in highly obstacle-dense environments. Then, the control points of the path can be adjusted by minimizing a cost function that considers the attraction to the goal and the repulsion to the threat regions. In [41], path planning is design considering the length and the altitude of the flight. The constraints-handling with α level comparison-based technique is incorporated in the Differential Evolution (DE) algorithm to avoid inappropriate turning angle, climbing/gliding slope, and attitude, to prevent the UAV flying into specific areas and limit the map range. This technique promotes the search into the constrained space to find search directions where feasible solutions are located.

Despite most of the recent meta-heuristics handle continuous optimization problems, the discretization of the workspace is still the preferred approach in the above works. It is also observed that the majority of the works address

unconstrained optimization problems by incorporating the constraints as penalization values into a single objective function. The above could be dangerous in many cases since there is not a guarantee of obtaining a feasible path or even the most suitable feasible path [41]. On the other hand, an additional advantage of the use of bio-inspired methods is the possibility of incorporating more requirements to the path-planning beyond the minimization of the path length (e.g., the reduction of the energy consumption). Finally, like the three previous methods, the bio-inspired ones do not take into account the full geometry of the obstacles or the UAV. So, valuable workspace regions are not considered in the search.

The present work aims to overcome the difficulties found in the studied path-planning methods. Therefore, this paper presents a new approach to the path-planning for UAVs in continuous 3D environments with static obstacles. In this approach, the path-planning problem is established as a constrained global optimization problem. For this, the control points between the start and target points must be optimized such that they minimize the overall path length. The constraints are related to the obstacle avoidance of both the control points and the edges that are generated between them. All obstacles and edges are modeled as Oriented Bounding Boxes (OBBs), while spherical shapes are adopted by control points to ease the collision detection. An enhanced path-search mechanism, based on a proposed variant of the traditional A-Star method (the Pruned A-Star), is included in the constrained version of the Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Differential Evolution (DE) to deal with complex planning scenarios. The enhanced mechanism incorporates a feasible path in the first set of candidate solutions of these meta-heuristics. The proposed path-planning approach is tested in six complex scenarios with the three different meta-heuristic algorithms. Because those meta-heuristic algorithms are the three most representative in the path planning for UAVs, we mainly consider them in the proposal to solve the problem and compare the proposed path planning approach with the obtained results given by the Pruned A-Star algorithm.

Hence, the main contributions of this work are listed next:

- A novel path-planning approach for continuous search spaces through a proposed multi-fusion based algorithm that adjusts the control points in such a way that the path length is minimized and the collisions are avoided in complex scenarios. The multi-fusion based algorithm enhanced the path-search mechanism by incorporating in the explorative search of the most used meta-heuristic algorithms, the feasible sub-optimal path of the Pruned A-Star algorithm.
- The proposed path-planning approach is stated assigning a 3D geometry to the path that provides a safe navigation space for the UAV and allows the selection of a suitable trade-off between the obstacle collision detection accuracy and the computational cost required in this procedure.
- The tackling of the collision avoidance problem through the use of feasibility rules for constraints-handling in the meta-heuristic algorithms to ensure feasible paths.

The remaining sections of this paper are organized as follows. Section I introduces the elements of the proposed path-planning approach, including the workspace representation, the optimization problem formulation, and the development of the meta-heuristic optimizers. The experimental conditions and the discussion of the results are presented in Section II. Finally, the conclusions and future work are addressed in Section IV.

II. OPTIMIZED PATH-PLANNING APPROACH FOR UAVs IN CONTINUOUS SPACES

The proposed approach consists of three steps. The first step is related to the representation of the workspace and all the elements in it. These elements include the UAV, the obstacles, and the start and target points. Given the workspace information, it is possible to formulate a constrained global optimization problem in the second step, whose solution contains the shortest possible path composed by distinct control points between the predefined start and target ones. This path must ensure the avoidance of threats or obstacles. In the last step, the above problem is solved through a meta-heuristic optimizer, and the obtained solution can be implanted in the UAV application.

A. REPRESENTATION OF THE WORKSPACE

The workspace of the UAV is its flight space, i.e., the volume reachable by this vehicle. On the other hand, the obstacles are defined as a group of objects within the workspace that cannot share the space of the UAV at the same time.

In virtual reality, the workspace and its elements can be described by a computational 3D representation or simulation [42]. One of the simplest and useful ways to perform the above representation is using bounding volumes with simple geometries such as spheres or boxes [43]. These geometries allow the execution of complex tasks, such as collision detection, at a low computational cost.

Therefore, the workspace is represented by an Axis-Aligned Bounding Box (AABB) [44] with dimension $d^w = [x^w, y^w, z^w]^T$ and origin $p_0^w = [0, 0, \frac{1}{2}z^w]^T$, i.e., as a fixed-dimension box placed on the floor in the center of the room or scenario.

Analogously, an Oriented Bounding Box (OBB) [44] is used to simplify the representation of the UAV. It has dimension $d^u = [x^u, y^u, z^u]^T$ and its origin and orientation continuously change during a flight mission. The dimension of the OBB should be chosen in such a way that the bounding box conforms to the geometry of the UAV and allows short-range maneuvers.

Regarding the obstacles, the OBB is also used for their representation. Depending on the actual obstacle geometry, different OBB configurations can be adopted, as observed in Fig. 1. Nevertheless, there is a trade-off between the computational complexity of the path-planning problem and the

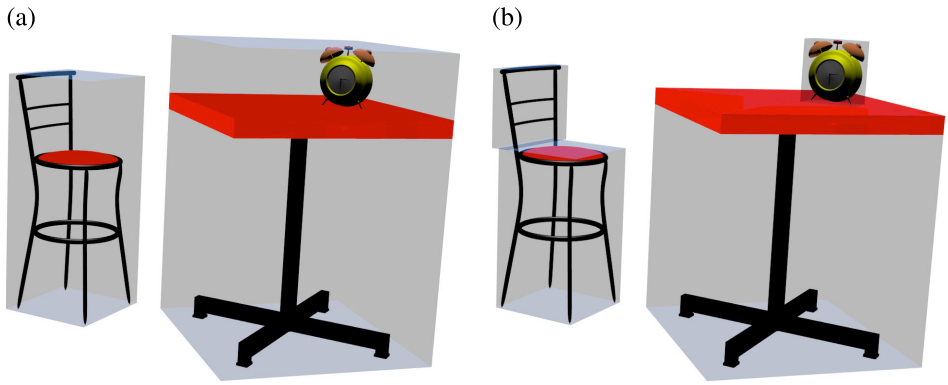

FIGURE 1. Representation of obstacles through oriented bounding boxes. (a) Two oriented bounding boxes wrap three objects in a more straightforward obstacle representation. (b) Four oriented bounding boxes wrap three objects in a more detailed obstacle representation.

detail level of the obstacle representation (a higher detail level implies a better path-planning accuracy). The simpler obstacle representation in Fig. 1-(a) requires less computational burden, while the more detailed one in Fig. 1-(b) needs additional processing. In any case, the selection of suitable trade-off depends on the application necessities.

Given the above, a set of n_o obstacles is distributed within the workspace, and their corresponding OBBs are referred as $B_k, \forall\ k = 1, \ldots, n_o$. For each B_k, the origin of the OBB is placed in the geometric center of the obstacle, while its orientation and dimension are chosen to accurately fit the geometry of the original object.

Finally, the start point p_s and target point p_t are considered as two 3D points that can be placed arbitrarily inside the workspace.

B. OPTIMIZATION PROBLEM STATEMENT

A constrained global optimization problem can be stated as in (1). The aim is to find a design variable vector x that minimizes a cost function or objective function $J(x)$. The possible values of x can be bounded by inequality constraints $g_i(x)$, equality constraints $h_j(x)$, and box constraints given by the lower and upper bounds of x, denoted by x^{lb} and x^{ub} respectively.

$$\begin{aligned} &\textbf{min}_{\ x\in\mathbb{R}^n}\ J(x) \\ &\text{subject to: } g_i(x) \leq 0, \quad i = 1, \ldots, n_1 \\ &\qquad\qquad h_j(x) = 0, \quad j = 1, \ldots, n_2 \\ &\qquad\qquad x_k^{lb} \leq x_k \leq x_k^{ub}, \quad k = 1, \ldots, n \end{aligned} \tag{1}$$

The path-planning problem for UAVs is proposed as a constrained global optimization problem, and its elements are described next.

1) DESIGN VARIABLE VECTOR

The design variable vector is observed in (2) and includes the coordinates of each 3D control point $p_l = [p_l^x, p_l^y, p_l^z]^T, \forall\, l = 1, \ldots, n$ along the path. The order of the points inside x is important, in the sense that each control point in x is connected with its predecessor and successor points to define the path depicted in Fig. 2. In the case of the first control point p_1 its predecessor is p_s, while the successor of the last point p_n is p_t. It is essential to mention that the number of control points n, used to describe a path from p_s and p_t, is a tunable parameter of the path-planner.

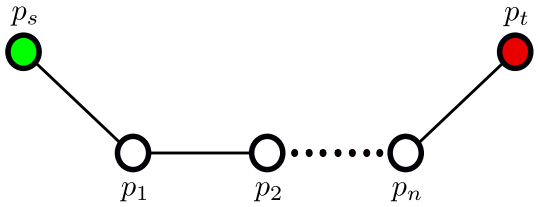

FIGURE 2. An arbitrary path which includes the start point, three control points, and the destination point.

$$x = [p_1, p_2, \ldots, p_n]^T \tag{2}$$

2) OBJECTIVE FUNCTION

The cost or objective function in (3) determines the distance of the complete path from p_s to p_t, i.e., the sum of the Euclidean distances (obtained by the Euclidean norm) between all the pairs of consecutive points in the path (edges), including p_s and p_t.

$$J(x) = \|p_s - p_1\| + \sum_{l=1}^{n-1}(\|p_l - p_{l+1}\|) + \|p_n - p_t\| \tag{3}$$

3) EQUALITY CONSTRAINTS

The proposed path-planning approach does not consider inequality constraints. However, a single equality constraint $h(x) = 0$ is included as a mechanism to generate collision-free paths. The function $h(x)$ indicates the times that the path collides with the obstacles within the workspace.

It is necessary to assign a 3D geometry to the path to compute $h(x)$. Since the path is composed by the start and target points, a set of control points, and the edges between all the pairs of consecutive points, two different 3D geometries are considered to represent the path:

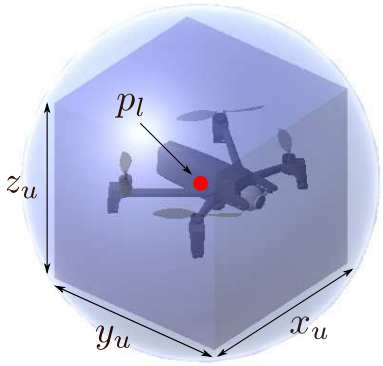

FIGURE 3. The oriented bounding box of the UAV is wrapped by a sphere that represents a control point.

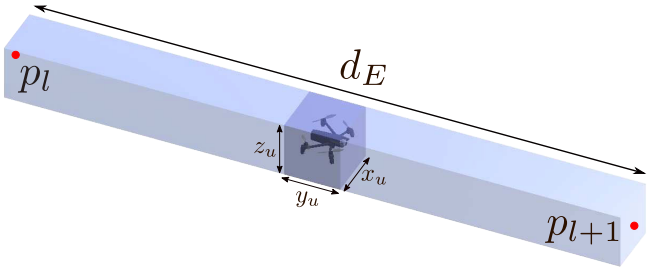

FIGURE 4. The oriented bounding box of the UAV is wrapped by the oriented bounding box that corresponds to the edge between two control points.

1) Spherical geometries $S_l, \forall\ l = 1, \ldots, n$ are selected to represent all the control points. The diameter of each sphere depends on the UAV dimension and is given by $d_s = \|d^u\|$, while its center is located at the control point p_l. This kind of geometry allows the UAV maneuverability, in the sense that it can change its forward direction towards the next point in the path. Fig. 3 shows the spherical geometry associated with an arbitrary control point p_l and its relationship with the OBB of the UAV.
2) The OBBs $E_l, \forall\ l = 1, \ldots, n+1$ are adopted to represent the edges in the path. For this, the width is x^u, the height is z^u, and the length is the Euclidean distance d_E, between the extreme points of the edge, i.e., a pair of consecutive control points or the pair of the start/target point with the first/last control point. The OBB center is located at the middle of the extreme points, and it is oriented considering a forward direction vector given by the difference of the extreme points, and an upward direction vector pointing up. Fig. 4 shows the OBB associated to the edge between two consecutive control points p_l and p_{l+1} and its relationship with the OBB of the UAV.

Fig. 5 exemplifies the use of all the considered geometries in a simple workspace with one obstacle. This figure shows an arbitrary path with three interconnected control points p_1, p_2, and p_3, that take the UAV from the start point p_s to the target one p_t. The edge geometries allow collision-free

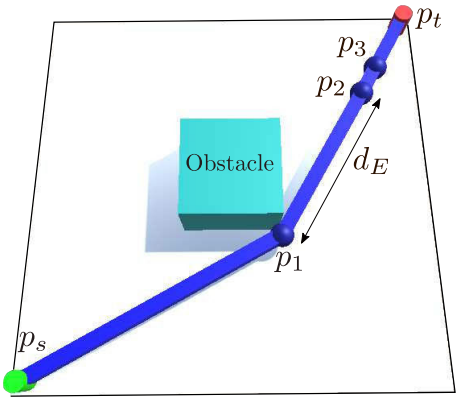

FIGURE 5. An example path with three control points and one obstacle in the center of the scenario.

linear displacements of the UAV, while the spherical shapes assigned to the control points permit orientation changes.

Given the 3D representation of the path, the single equality constraint is defined in (4), where $c(G_1, G_2)$ in (5) is the function that determines a colission between the geometries G_1 and G_2 based on the 3D Separating Axis Theorem (SAT) [45].

$$h(x) = \sum_{k=1}^{n_o}\sum_{l=1}^{n} c(B_k, S_l) + \sum_{k=1}^{n_o}\sum_{l=1}^{n+1} c(B_k, E_l) \tag{4}$$

$$c(G_1, G_2) = \begin{cases} 1, & \textbf{if } G_1 \text{ intersects } G_2 \\ 0, & \textbf{otherwise} \end{cases} \tag{5}$$

4) BOX CONSTRAINTS

The location of the control points in the design variable vector (2) is limited to the space within the workspace geometry. Then, for an arbitrary control point $x_k = p_l$, the lower bounds are $x_k^{lb} = [-\frac{1}{2}x^w, -\frac{1}{2}y^w, 0]^T$, and the upper bounds are $x_k^{ub} = [\frac{1}{2}x^w, \frac{1}{2}y^w, z^w]^T$.

C. OPTIMIZERS

The path-planning problem stated before is constrained, highly nonlinear, and has a multimodal nature (regarding the possible control point combinations aligned in a single edge). Moreover, for complex workspaces, i.e., scenarios with a reasonably large number of obstacles, an initial guess solution is hard to propose. Therefore, intelligent techniques such as meta-heuristics [46] are preferred over classic optimization approaches [47] in the search for optimized paths.

1) META-HEURISTICS

Meta-heuristics are stochastic computational techniques designed to find proper solutions to hard optimization problems, such as the path-planning one, at a reasonable cost [48], [49]. These techniques have increased their popularity due to their high performance, universality, and simplicity [32], [33]. Besides, most of them are inspired by natural phenomena.

Although many meta-heuristics have been developed over time, none of them can solve all classes of problems, as stated

in the No Free Lunch (NFL) theorem [50]. So, different alternatives must be tested to find the most suitable choice for a particular class of problems.

Among meta-heuristic optimizers, the Differential Evolution (DE) [51], the Genetic Algorithm (GA) [52], and the Particle Swarm Optimization (PSO) [53] are recurrently used in the specialized literature for the solution of a vast variety of optimization problems showing an outstanding performance [54]–[56].

For the above reason, the optimizers mentioned above are adopted for the solution of the path-planning problem, and they are described next.

1) **Differential Evolution (DE) [51]:** The natural evolution phenomenon inspires this technique. During several generations, the individuals (solution vectors) of a population (randomly initialized) mutate and recombine to generate offsprings. The mutation uses the scaled differences of the solutions in the population to obtain mutants. At given rates, the mutants interchange variables with the original individuals to generate offsprings. If these offsprings are fitter, they can replace the original solutions. By the end generation, the best solution is in the population.
2) **Genetic Algorithm (GA) [52]:** The natural evolution also inspires this optimizer. For each generation of the GA, the individuals in the population (randomly initialized) compete to reproduce. Fitter individuals have better chances to recombine and generate offsprings. The obtained offsprings can mutate at given rates to maintain diversity. When a generation finishes, only the fittest individuals from both the original and offspring populations persist. In the last generation, the best individual is found in the population.
3) **Particle Swarm Optimization (PSO) [53]:** This alternative is inspired by the collaborative behavior of species in the search for resources to survive. For each iteration of PSO, the positions (solution vectors) of the particles in a swarm (randomly initialized) are updated based on a velocity factor. This factor is calculated based on the pondered individual particle memory (the best position known by the particle) and swarm knowledge (the best position known by the swarm or by a group of neighbors). When the last iteration is reached, the swarm knows the best position in the search space.

Due to the presence of hard constraints in the path-planning problem, a constraints-handling mechanism is necessary to choose the best alternatives by taking their feasibility into account. Moreover, for very complex scenarios, the probability of getting a random feasible path in the early generations/iterations of the meta-heuristics is too low, and an enhanced path-search mechanism is necessary. Next, the above two mechanisms are detailed.

2) CONSTRAINTS-HANDLING MECHANISM

For the selected meta-heuristic algorithms, a fitness criterion is needed to decide if a solution is better than another. That criterion is used either to choose the mating partners, to determine the survivors, or to update the solution knowledge [57].

The proposed path-planning approach adopts a variation of the rules described in [58]. Those rules determine the solution fitness based on its feasibility and convergence separately, instead of using these features together in a single criterion that considers the penalization of the objective function. The above due to penalization methods do not allow the discrimination of unfeasible solutions during the optimization procedure [59]. Then, the obtained paths using penalization may be dangerous in the context of an actual flight mission with the UAV.

The rules mentioned above are established in a tournament selection operator, and the proposal extends it to the context of path-planning as follows:

- Any feasible path is preferred to an unfeasible one.
- Among two feasible paths, the one with the shortest length is preferred.
- Among two unfeasible paths, the one with the fewest collisions is preferred.

In addition to the above rules, if two unfeasible paths have the same number of collisions, the fittest solution is selected randomly.

The above criteria is adopted in the selection stages of DE and GA, and is used to determine the best positions known by the particles and the swarm in PSO.

3) ENHANCED PATH-SEARCH MECHANISM

The presence of threats and obstacles restricts the safe flying space of the UAV. The smaller the safe space, the greater the difficulty in calculating a feasible path stochastically.

As described before, meta-heuristic techniques such as DE, GA, and PSO start the search from a set of random candidate solutions. If the workspace is very complex, finding the first feasible path can take a long time and consume a lot of computational resources.

Because of the above, the selected meta-heuristic algorithms are endowed with an enhanced path-search mechanism. For each meta-heuristic, this mechanism feeds the first set of solutions with a single feasible alternative obtained from a deterministic path generator, i.e., an individual/particle of the population/swarm.

Among the path generators in the specialized literature, the A-Star algorithm is a well-known efficient alternative that deterministically finds sub-optimal feasible paths in complex scenarios [60]. A-Star operates over a grid discretization of the workspace that considers the cells containing obstacles or threats as prohibited regions. Moreover, A-Star starts the search from an initial location in the grid and perform movements to adjacent feasible cells. For each step, A-Star evaluates a function to determine the cost of moving from a current cell to another towards the target.

Hence, A-Star is adopted to feed meta-heuristics with a single feasible solution. In this work, the A-Star uses the cost function $\hat{f}(q) = \hat{h}(q) + \hat{g}(q)$ presented in [61], where q is

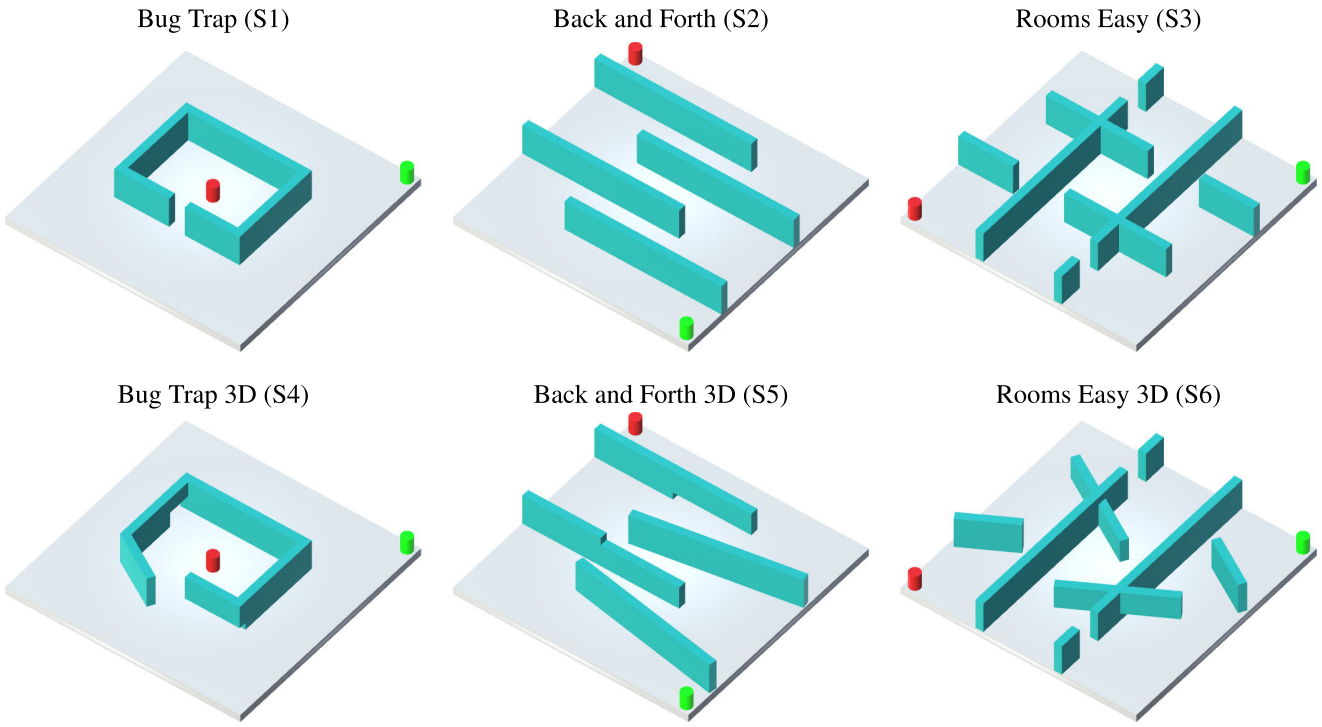

FIGURE 6. Scenarios used to test efectiveness of the proposal.

the next cell, $\hat{h}(q)$ is the Manhattan heuristic function that measures the cost of the path from q to the target, and $\hat{g}(q)$ is the cost from the start cell to q. Also, movements from a given cell to another are restricted to six directions, including up, down, right, left, forward, and backward, i.e., diagonal movements are not allowed to avoid possible collisions with the 3D corners of the obstacle OBBs.

The path achieved by A-Star is a list of consecutive cells from the start location to the target one. Based on the above list, a set of control points is extracted from the center of each cell in the path, and then is included in the initial solution set of the meta-heuristics. Then, the number of control points is also determined by A-Star.

Due to the paths found by A-Star can contain several consecutive cells along the same edge, a pruning strategy is proposed to remove all the successive cells with the same movement direction. This A-Star variant is referred from now on as Pruned A-Star.

Using the Pruned A-Star, an equivalent path is obtained with a lower number of cells, and in consequence, the meta-heuristic optimizers require to adjust a lower number of control points.

III. RESULTS AND DISCUSSION

A. EXPERIMENT DETAILS

The Parrot ANAFI quadrotor is considered as the UAV. For this UAV, its container OBB dimension is set as $d^u = [0.175, 0.24, 0.065]^T$ (m) to allow short-range maneuvers.

Six different scenarios were developed in the Unity game engine [62] to test the effectiveness of the proposed path-planning approach, and they are shown in Fig. 6. The first three scenarios, i.e., the Bug Trap (S1), the Back and Forth (S2), and the Rooms Easy (S3), are the 3D versions of three 2D complex scenarios from the motion planning map repository in [63]. The original 2D obstacles were extruded to the ceiling of the workspace to generate these 3D versions.

The remaining three scenarios, i.e., the Bug Trap 3D (S4), the Back and Forth 3D (S5), and the Rooms Easy 3D (S6), are variations of the first three that consider different obstacle orientations and shapes to provide additional safe regions.

All the alternatives depicted in Fig. 6 share the same dimension $d^w = [30, 30, 4]^T$ (m), and the location of the start and target points is highlighted in green and red, respectively.

Table 1 shows additional details about the adopted scenarios, including the number of obstacles n_o, the location of the start and target points (p_s and p_t, respectively), and the complexity ρ. The complexity ρ is the rate of feasible (collision-free) paths in a set of a million alternatives generated randomly (between the established p_s and p_t, and using the number of control points provided by the Pruned A-Star for each scenario). So, a smaller value of ρ implies a more complex path-planning problem. According to the ρ values in Table 1, the adopted scenarios entail very complex path-planning problems.

Cubic cells of dimension 1 (m) are considered to discretize the workspace for the Pruned A-Star. The number of control points achieved by the Pruned A-Star for all the scenarios is observed in Table 1. It is important to remark that the above value denotes the number of points that need

TABLE 1. Details of the developed scenarios.

Scenario	Name	Obstacles (n_o)	Start point (p_s)	Target point (p_t)	Control points (n)	Complexity (ρ)
S1	Bug Trap	5	$[14, 14, 1]^T$	$[0, 0, 1]^T$	4	1.0640E-3 (0.1064%)
S2	Back and Forth	4	$[14, -14, 1]^T$	$[-14, 14, 1]^T$	10	0.0000 (0.0000%)
S3	Rooms Easy	8	$[14, 14, 1]^T$	$[-14, -14, 1]^T$	18	0.0000 (0.0000%)
S4	Bug Trap 3D	6	$[14, 14, 1]^T$	$[0, 0, 1]^T$	8	9.5000E-5 (0.0095%)
S5	Back and Forth 3D	6	$[14, -14, 1]^T$	$[-14, 14, 1]^T$	20	0.0000 (0.0000%)
S6	Rooms Easy 3D	8	$[14, 14, 1]^T$	$[-14, -14, 1]^T$	22	0.0000 (0.0000%)

TABLE 2. Results of the descriptive statistical analysis.

Scenario	Optimizer	$Mean(J)$ (m)	$Std(J)$ (m)	J_{min} (m)	J_{max} (m)
S1	DE	**34.3372**	0.0526	**34.3111**	34.5980
(ρ=1.0640E-3)	GA	34.3410	**0.0181**	34.3285	**34.4252**
	PSO	34.6707	0.1428	34.4155	35.0166
S2	DE	78.4561	0.6221	**77.8471**	79.7051
(ρ=0.0000)	GA	**78.4533**	**0.3813**	77.8759	**79.1146**
	PSO	88.0715	1.8122	84.0151	91.9291
S3	DE	**88.1850**	0.9248	**87.3806**	91.8931
(ρ=0.0000)	GA	88.4501	**0.5769**	87.6572	**89.8263**
	PSO	110.7170	2.7546	103.5064	115.2676
S4	DE	29.7986	0.0815	**29.7570**	30.0919
(ρ=9.5000E-5)	GA	**29.7972**	**0.0305**	29.7756	**29.9210**
	PSO	31.3494	0.8631	30.4645	33.8432
S5	DE	48.4103	0.7278	**47.9288**	51.3850
(ρ=0.0000)	GA	**48.1354**	**0.0962**	48.0198	**48.4523**
	PSO	55.4473	1.8411	52.6068	59.5182
S6	DE	74.4667	0.6136	**73.6970**	76.0007
(ρ=0.0000)	GA	**74.0667**	**0.2484**	73.7730	**74.9037**
	PSO	85.7374	2.2488	81.8335	90.4314

to be adjusted by the meta-heuristics in the continuous search space.

Concerning the optimizers, the DE, GA, and PSO meta-heuristics, with the constraints-handling mechanism and the enhanced path-search mechanism described in Section II.C, are selected to solve the path-planning problem.

The remaining details and parameters of the used meta-heuristics are described next:

- **DE**: The variant DE/best/1/bin [64] is selected. The scaling factor and the crossover rate of DE/best/1/bin are set respectively as $F = 0.7$ and $CR = 0.8$.
- **GA**: The version of the GA proposed in [65] is adopted in this work. This GA variant uses the polynomial mutation (PM), the simulated binary crossover (SBX), and the binary tournament selection. The used parameters are the crossover probability $p_c = 1.0$, the mutation probability $p_m = 0.1$, the SBX distribution index $\eta_c = 100$ and the PM distribution index $\eta_m = 100$.
- **PSO**: The chosen variant uses a full-connected topology with variable inertia factor [66]. For this PSO version the parameters are the local factor $C_1 = 2.5$, the global factor $C_2 = 1.5$, and the maximum and minimum values of the inertia $\omega_{max} = 0.1$ and $\omega_{min} = 0$.

The above algorithm configurations were obtained by trial and error, starting from the parameter values reported in their corresponding researches and then systematically updating them to minimize the total path length for the test scenario S6 (the most complex alternative regarding the complexity ρ, the number of obstacles n_o and the number of control points n).

All the above optimizers are assigned with the same population/swarm size $NP = 20$ and the same number of generations/iterations $G_{max} = 2000$ to perform fair comparisons. The algorithms are implemented using the *C#* programming language in a PC with a core $i7 - 8th$ processor at 2.2 GHz and 8 GB RAM.

B. EXPERIMENT RESULTS

Thirty independent runs of the DE, GA, and PSO optimizers are performed for each scenario to show the effectiveness of the proposal. It is important to highlight that there are no unfeasible solutions by the end generation/iteration of these algorithms, i.e., all alternatives generate collision-free paths.

The results of the descriptive statistical analysis are shown in Table 2. The information in this table is grouped by scenario, which is indicated in the first column. Each group describes the results achieved by each optimizer. They include the mean path length $Mean(J)$ and the corresponding standard deviation $Std(J)$, the length of the shortest path J_{min}, and the length of the longest path J_{max}. The best results are highlighted in boldface.

The $Mean(J)$ value in Table 2 indicates that DE can find the best paths for S1 and S3. On the other hand, the GA finds better paths for the rest of the scenarios and is closely followed by DE in performance. In the case of the PSO, it does not outperform the other optimizers. Regarding the $Std(J)$ values, they describe the repeatability and confidence of the results. In this sense, the GA obtains similar paths in every run for all the scenarios and is followed by DE and the

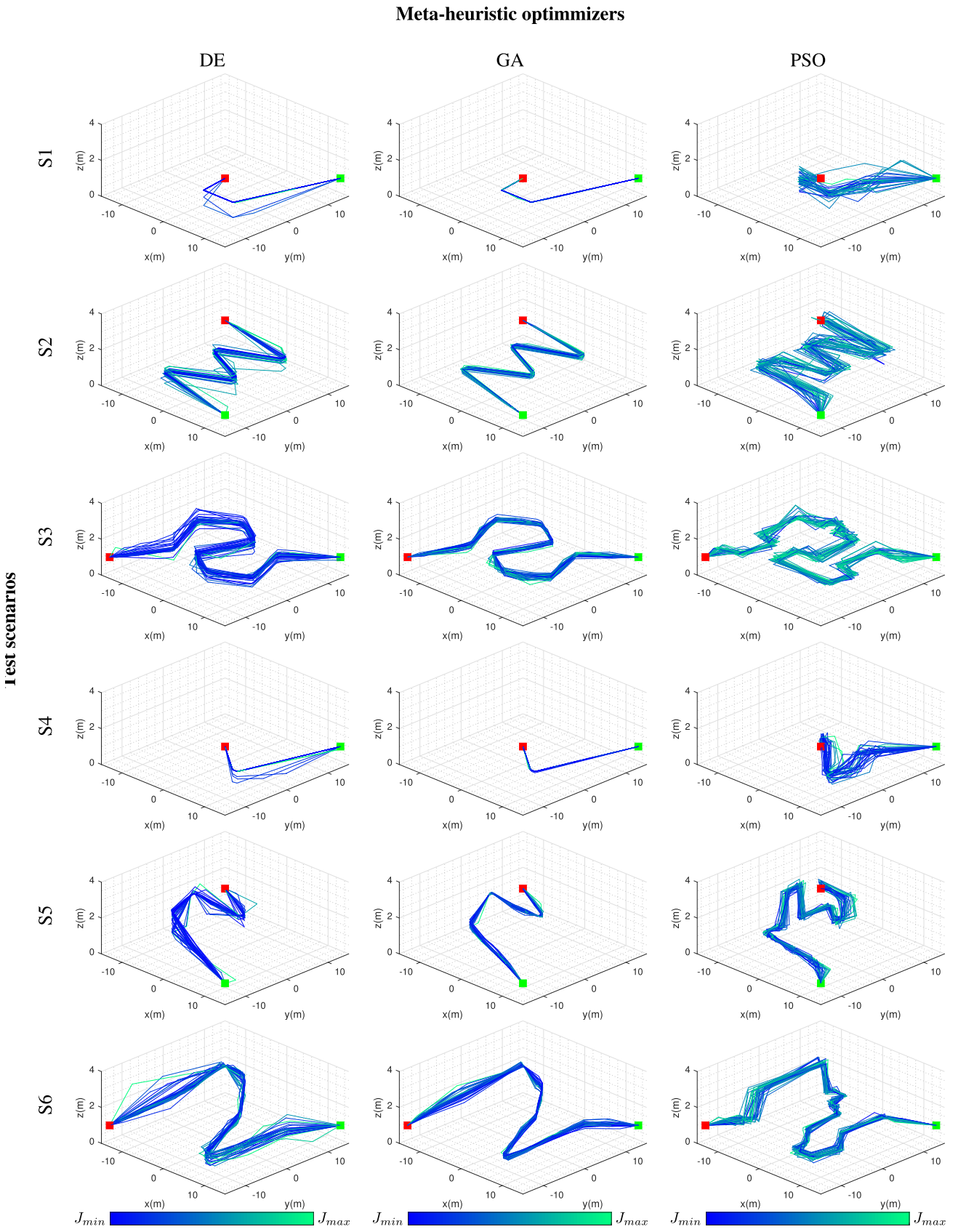

FIGURE 7. Paths obtained by each meta-heuristic optimizer for all scenarios.

PSO in this order. Considering J_{min}, DE gets the best overall short paths in all cases, and according to J_{max}, the GA reaches the best overall long path for all the scenarios.

Despite the observed performance differences, the $Std(J)$ value exhibits small deviations (in the interval of 0.0526(m) and 0.9248(m)) for the best optimizers DE and GA. It highlights the convergence level of the proposal towards the optimal solution.

It is important to mention that while DE and the GA show different advantages in the path-planning, only PSO is far from the best performance in all comparisons. The above is attributed to the lack of a selection mechanism in the PSO operation, which is included in both DE and the GA.

Fig. 7 shows the thirty paths obtained by each optimizer for all scenarios. In this figure, the green and red squares represent the start and target points, respectively. A path of darker color indicates that the length is closer to the J_{min} value in Table 2. Analogously, a path of lighter color points out a length similar to J_{max} in the same table.

The paths depicted in Fig. 7 can be contrasted with the values in Table 2. In this figure, the paths of the GA appear to be similar and evenly separated around a mean shape, as suggested before by the $Std(J)$ value. On the other hand, the paths obtained by DE appear to converge to a reduced set of shapes. Although these shapes are visibly different, their mean length is outstanding. Then, DE could be finding different local best solutions, i.e., different paths with congruent lengths (regarding the low value of $Std(J)$). In the case of the PSO, the achieved paths are dispersed, and a mean shape cannot be easily distinguished. The above leads to the poor performance discussed before.

Since all the optimizers have a non-deterministic operation and the result distribution is different than the normal, additional non-parametric statistical analyzes are necessary to confirm the best alternative [67]. In this way, the Wilcoxon pairwise test is adopted in this work to draw robust conclusions. This test allows finding significant median differences of the length distribution obtained by two optimizers for a single scenario. For this, a two-sided null hypothesis is considered, and the statistical significance of the test is set to 5%. The two-sided null hypothesis claims that the median of two distributions is identical, and can be rejected if the resulting *p-value* is less than 5%.

The results of the Wilcoxon test are observed in Table 3. They are also grouped by scenario and show all possible pairwise tests, as well as their resulting *p-value*. Moreover, this table includes the R_+ and R_- values. The R_+ value indicates the times that the first optimizer overcomes the second one, while the R_- value indicates the opposite. The best alternative in a pairwise test is shown in boldface.

In most of the pairwise tests, there is a clear winner, i.e., the *p-value* allows the rejection of the null hypothesis. Only in the cases of S2 and S5, the performance of DE and the GA are equally good, and the null hypothesis is accepted.

The results of the Wilcoxon test are summarized in Table 4 to aid the selection of the most suitable optimizer for the path-planning problem. According to the total number of wins in this table, the DE algorithm has the overall best performance, and is followed by the GA and the PSO in this order.

TABLE 3. Results of the non-parametric statistical analysis.

Scenario	Pairwise test	R_+	R_-	$p\text{-}value$
S1	**DE** vs GA	381	84	0.0015
(ρ=1.0640E-3)	**DE** vs PSO	465	0	1.8626E-09
	GA vs PSO	465	0	1.8626E-09
S2	DE vs GA	238	227	0.9192
(ρ=0.0000)	**DE** vs PSO	465	0	1.8626E-09
	GA vs PSO	465	0	1.8626E-09
S3	**DE** vs GA	334	131	0.0364
(ρ=0.0000)	**DE** vs PSO	465	0	1.8626E-09
	GA vs PSO	465	0	1.8626E-09
S4	**DE** vs GA	343	122	0.0220
(ρ=9.5000E-5)	**DE** vs PSO	465	0	1.8626E-09
	GA vs PSO	465	0	1.8626E-09
S5	DE vs GA	162	303	0.1518
(ρ=0.0000)	**DE** vs PSO	465	0	1.8626E-09
	GA vs PSO	465	0	1.8626E-09
S6	DE vs **GA**	104	361	0.0071
(ρ=0.0000)	**DE** vs PSO	465	0	1.8626E-09
	GA vs PSO	465	0	1.8626E-09

TABLE 4. Summary of the non-parametric statistical analysis.

Scenario	DE	GA	PSO
S1 (ρ=1.0640E-3)	2	1	0
S2 (ρ=0.0000)	1	1	0
S3 (ρ=0.0000)	2	1	0
S4 (ρ=9.5000E-5)	2	1	0
S5 (ρ=0.0000)	1	1	0
S6 (ρ=0.0000)	1	2	0
Total wins	**9**	7	0

Additionally, the paths achieved by the Pruned A-Star algorithm are considered as a point of comparison to determine the advantages and drawbacks of the proposal. The information of these paths can be consulted in Table 5. This table indicates the length J and the number of control points n of the paths obtained by the Pruned A-Star for each scenario.

Compared to the worst paths found by DE (the best-performing alternative), the ones of the Pruned A-Star are between 10% and 37% longer. Even if the paths of the Pruned A-Star are compared to those of PSO (the worst-performing alternative), a noticeable improvement is achieved with the proposal. This fact is attributed to the use of the enhanced path-search mechanism in all the meta-heuristics, which ensures that the obtained solutions are at least equally good as those obtained by the Pruned A-Star.

The above differences describe a strong advantage of the proposed path-planning approach concerning the power saving in the UAV, whose onboard battery is currently a critical limitation.

A possible drawback of the proposal regarding the Pruned A-Star method is the computational cost. Since the proposal requires the path calculated by the Pruned A-Star method, its cost is always greater than that of the last. In terms of computational time, the proposal requires about triple the time of the

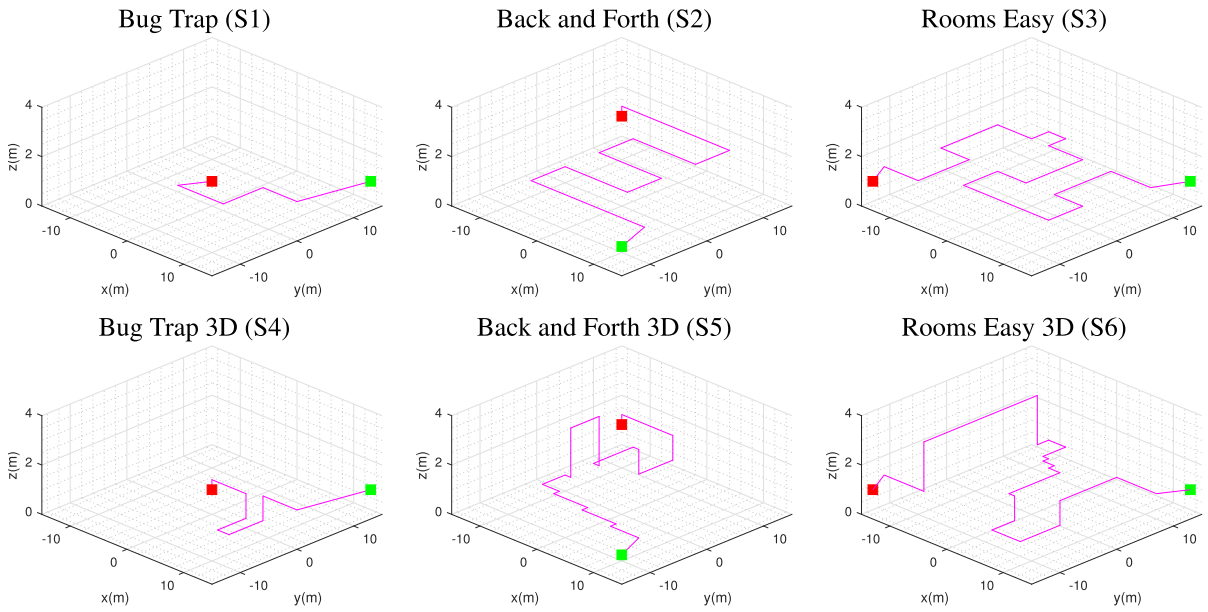

FIGURE 8. Paths obtained by the Pruned A-Star for all scenarios.

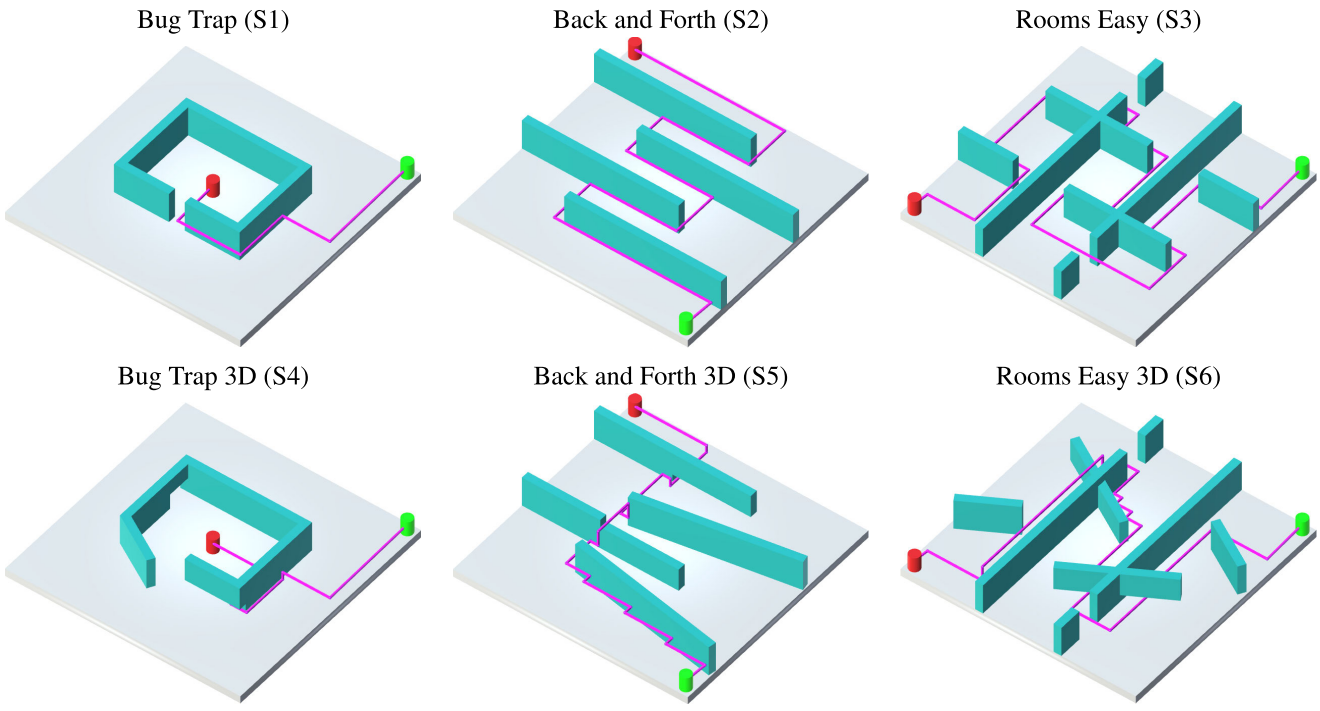

FIGURE 9. Paths obtained by the Pruned A-Star for all scenarios.

Pruned A-Star to generate an optimized path. Nevertheless, the cost is affordable with the considered hardware, and the mean time required to compute a path with the proposal for scenario S6 (considered as the most complex) is 15 (s).

Regarding the shapes of the paths obtained by the Pruned A-Star, they are shown in Fig. 8. Based on these shapes, it is possible to observe some similarities with the paths presented in Fig. 7. For example, the paths obtained by the PSO appear to preserve many features of the original Pruned A-Star solutions; hence its low performance can be explained. In the cases of DE and GA, their paths present few coincidences with those of the Pruned A-Star, which is confirmed by the notable performance differences in Tables 2 and 5.

Finally, the paths achieved by the Pruned A-Star and the best ones computed by DE are depicted in Figures 9 and 10, respectively. Contrasting the paths in both figures, DE obtains

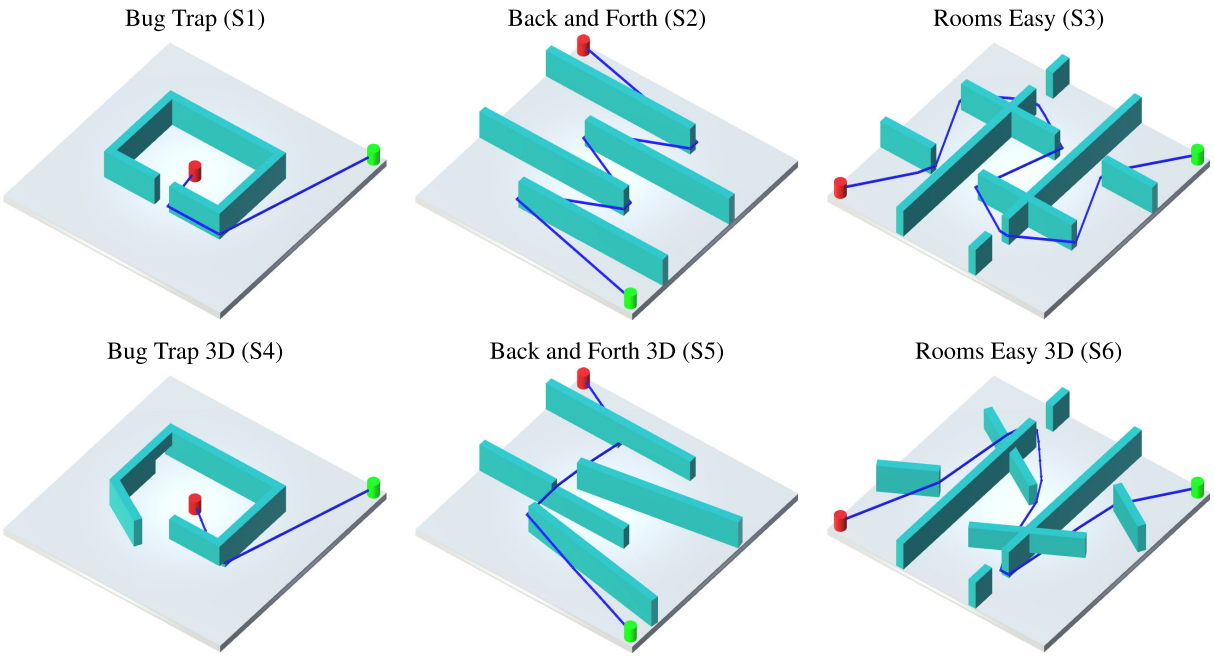

FIGURE 10. Best paths obtained by DE for all scenarios.

TABLE 5. Deterministic results of the Pruned A-Star.

Scenario	J (m)	Control points (n)
S1 (ρ=1.0640E-3)	41.0568	4
S2 (ρ=0.0000)	100.4367	10
S3 (ρ=0.0000)	124.1915	18
S4 (ρ=9.5000E-5)	41.3845	8
S5 (ρ=0.0000)	66.4641	20
S6 (ρ=0.0000)	100.1966	22

visibly smoother paths than the Pruned A-Star, i.e., paths with fewer direction changes.

IV. CONCLUSIONS AND FUTURE WORK

The proposed path-planning approach finds feasible (collision-free) paths for UAVs in continuous 3D environments. It is based on the solution of a global constrained optimization problem. This problem aims to minimize the objective function, which considers the overall path length while satisfying the constraints related to the collisions of the 3D obstacles geometry and the path 3D shape.

The statement of the path-planing problem as a global constrained optimization one entails several advantages. The most representative is related to the computation of paths in a continuous search space. In this way, the optimized paths have higher performance when compared to solutions from path-planning methods based on the discretization of the search space such as A-Star. The above because the latter methods only find a limited number of paths, while the number of alternatives that the proposal can find is incommensurable.

On the other hand, the use of simple 3D shapes to represent the workspace (using an AABB), the obstacles (through OBBs), and the path (adopting OBBs and Spherical geometries), allows the efficient and affordable calculation of collisions (by the SAT). Moreover, it conforms to a framework to establish a proper trade-off between path-planning accuracy and computational cost.

By its own nature, the path-planning problems are complex regarding features like high nonlinearity, multimodality, non-differentiability, and the lack of an initial guess solution. So, meta-heuristics are exceptional candidates to find suitable solutions. Due to the absence of a universal meta-heuristic (as stated in the "No Free Lunch" theorem), three well-known meta-heuristics, including DE, PSO, and the GA, were adopted to show the proposal effectiveness for three complex scenarios. All these methods are endowed with an enhanced path-search mechanism in the initial solution set and feasibility rules to handle the constraints.

The penalization methods included in meta-heuristics to handle constrained problems do not discriminate unfeasible solutions. The above may be dangerous for the UAV flight mission. This difficulty is overcome in this work through the use of a constraints-handling mechanism that allows the searching for solutions by pondering first their feasibility and then their convergence.

Moreover, the use of the enhanced path-search mechanism in the meta-heuristics based on the proposed Pruned A-Star method reduces the computation effort required to achieve the first feasible path. Then, the computational burden is invested in finding better performing paths.

After descriptive and non-parametric statistical analyses over the results of six complex test cases, the paths found by DE showed outstanding behavior in terms of convergence and confidence level. In addition, the proposed path-planning approach presents performance advantages over a well-known and widely used method like A-Star.

As future work, the global constrained optimization problem presented in this work can be extended to a dynamic

one [68] to handle changing and uncertain environments. The new problem must consider the time-varying information of obstacles and threats within a visibility range. Then, the solution to that problem will be the online tracking of the optimal path [69]. Moreover, the assignation of simpler 3D shapes to the workspace elements and the use of dynamic versions of meta-heuristics [69] could be necessary to achieve a real-time implementation and prevent the path calculation from scratch.

REFERENCES

[1] J. M. Beer, A. D. Fisk, and W. A. Rogers, "Toward a framework for levels of robot autonomy in human-robot interaction," *J. Hum.-Robot Interact.*, vol. 3, no. 2, p. 74, Jun. 2014.

[2] R. M. C. Santiago, A. L. De Ocampo, A. T. Ubando, A. A. Bandala, and E. P. Dadios, "Path planning for mobile robots using genetic algorithm and probabilistic roadmap," in *Proc. IEEE 9th Int. Conf. Hum., Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Dec. 2017, pp. 1–5.

[3] L. Liu, H. Wang, C. Li, and C. Zhao, "Research and application of plan recognition in intelligent tutoring system," in *Proc. IEEE Int. Symp. IT Med. Educ.*, vol. 2, Dec. 2011, pp. 53–56.

[4] N. Mahmoudian, J. Geisbert, and C. Woolsey, "Approximate analytical turning conditions for underwater gliders: Implications for motion control and path planning," *IEEE J. Ocean. Eng.*, vol. 35, no. 1, pp. 131–143, Jan. 2010.

[5] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. Cambridge, MA, USA: MIT Press, 2011.

[6] J. Wirtz, P. G. Patterson, W. H. Kunz, T. Gruber, V. N. Lu, S. Paluch, and A. Martins, "Brave new world: Service robots in the frontline," *J. Service Manage.*, vol. 29, no. 5, pp. 907–931, Oct. 2018.

[7] H. Duan and P. Qiao, "Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning," *Int. J. Intell. Comput. Cybern.*, vol. 7, no. 1, pp. 24–37, Mar. 2014.

[8] V. Lumelsky and K. Sun, "A unified methodology for motion planning with uncertainty for 2D and 3D two-link robot arm manipulators," *Int. J. Robot. Res.*, vol. 9, no. 5, pp. 89–104, Oct. 1990.

[9] M. Elhoseny, A. Tharwat, and A. E. Hassanien, "Bezier curve based path planning in a dynamic field using modified genetic algorithm," *J. Comput. Sci.*, vol. 25, pp. 339–350, Mar. 2018.

[10] R. Mahony and V. Kumar, "Aerial robotics and the quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, p. 19, 2012.

[11] H. A. Ruff, S. Narayanan, and M. H. Draper, "Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles," *Presence Teleoperators Virtual Environ.*, vol. 11, no. 4, pp. 335–351, Aug. 2002.

[12] V. Roberge, M. Tarbouchi, and G. Labonte, "Fast genetic algorithm path planner for fixed-wing military UAV using GPU," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 5, pp. 2105–2117, Oct. 2018.

[13] B. Hubbard, H. Wang, M. Leasure, T. Ropp, T. Lofton, S. Hubbard, and S. Lin, "Feasibility study of UAV use for RFID material tracking on construction sites," in *Proc. 51st ASC Annu. Int. Conf.*, 2015, pp. 1–8.

[14] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[15] M. J. Lum, J. Rosen, H. King, D. C. W. Friedman, G. Donlin, G. Sankaranarayanan, B. Harnett, L. Huffman, C. Doarn, T. Broderick, and B. Hannaford, "Telesurgery via unmanned aerial vehicle (UAV) with a field deployable surgical robot," in *Proc. MMVR*, 2007, pp. 313–315.

[16] F. P. Kemper, K. A. O. Suzuki, and J. R. Morrison, "UAV consumable replenishment: Design concepts for automated service stations," *J. Intell. Robotic Syst.*, vol. 61, nos. 1–4, pp. 369–397, Jan. 2011.

[17] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of robot 3D path planning algorithms," *J. Control Sci. Eng.*, vol. 2016, pp. 1–22, Mar. 2016.

[18] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[19] Y. Lin and S. Saripalli, "Sampling-based path planning for UAV collision avoidance," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3179–3192, Nov. 2017.

[20] X. Wu, L. Xu, R. Zhen, and X. Wu, "Biased sampling potentially guided intelligent bidirectional RRT algorithm for UAV path planning in 3D environment," *Math. Problems Eng.*, vol. 2019, pp. 1–12, Nov. 2019.

[21] K. J. Obermeyer, P. Oberlin, and S. Darbha, "Sampling-based path planning for a visual reconnaissance unmanned air vehicle," *J. Guid., Control, Dyn.*, vol. 35, no. 2, pp. 619–631, Mar. 2012.

[22] Z. Wang, Y. Li, H. Zhang, C. Liu, and Q. Chen, "Sampling-based optimal motion planning with smart exploration and exploitation," *IEEE/ASME Trans. Mechatronics*, early access, Feb. 13, 2020, doi: 10.1109/TMECH.2020.2973327.

[23] H. Huang, A. V. Savkin, and W. Ni, "Energy-efficient 3D navigation of a solar-powered UAV for secure communication in the presence of eavesdroppers and no-fly zones," *Energies*, vol. 13, no. 6, p. 1445, Mar. 2020.

[24] W. Zhan, W. Wang, N. Chen, and C. Wang, "Efficient UAV path planning with multiconstraints in a 3D large battlefield environment," *Math. Problems Eng.*, vol. 2014, pp. 1–12, Jan. 2014.

[25] Y. Zhou, X. Jin, and T. Wang, "FPGA implementation of a algorithm for real-time path planning," *Int. J. Reconfigurable Comput.*, vol. 2020, Aug. 2020, Art. no. 8896386.

[26] M. Naazare, D. Ramos, J. Wildt, and D. Schulz, "Application of graph-based path planning for UAVs to avoid restricted areas," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Sep. 2019, pp. 139–144.

[27] F. L. L. Medeiros and J. D. S. Da Silva, "A Dijkstra algorithm for fixed-wing UAV motion planning based on terrain elevation," in *Proc. Brazilian Symp. Artif. Intell.* Berlin, Germany: Springer, 2010, pp. 213–222.

[28] Y.-B. Chen, G.-C. Luo, Y.-S. Mei, J.-Q. Yu, and X.-L. Su, "UAV path planning using artificial potential field method updated by optimal control theory," *Int. J. Syst. Sci.*, vol. 47, no. 6, pp. 1407–1420, Apr. 2016.

[29] Z. Zhang, J. Li, and J. Wang, "Sequential convex programming for nonlinear optimal control problems in UAV path planning," *Aerosp. Sci. Technol.*, vol. 76, pp. 280–290, May 2018.

[30] D. Hashemi and H. Heidari, "Trajectory planning of quadrotor UAV with maximum payload and minimum oscillation of suspended load using optimal control," *J. Intell. Robotic Syst.*, vol. 2020, pp. 1–13, Mar. 2020.

[31] M. Tuqan, N. Daher, and E. Shammas, "A simplified path planning algorithm for surveillance missions of unmanned aerial vehicles," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2019, pp. 1341–1346.

[32] I. H. Osman and G. Laporte, "Metaheuristics: A bibliography," *Ann. Oper. Res.*, vol. 63, pp. 511–623, 1996.

[33] M. Gendreau J.-Y. Potvin, *Handbook of Metaheuristics*, vol. 2. New York, NY, USA: Springer, 2010.

[34] T. Bäck, D. B. Fogel, and Z. Michalewicz, "Handbook of evolutionary computation," *Release*, vol. 97, no. 1, p. B1, 1997.

[35] J. Kennedy, "Swarm intelligence," in *Handbook of Nature-Inspired and Innovative Computing*. New York, NY, USA: Springer, 2006, pp. 187–219.

[36] Y. Chen, Y. Mei, J. Yu, X. Su, and N. Xu, "Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm," *Neurocomputing*, vol. 266, pp. 445–457, Nov. 2017.

[37] V. Roberge, M. Tarbouchi, and G. Labonté, "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.

[38] Y. He, Q. Zeng, J. Liu, G. Xu, and X. Deng, "Path planning for indoor UAV based on ant colony optimization," in *Proc. 25th Chin. Control Decis. Conf. (CCDC)*, May 2013, pp. 2919–2923.

[39] S. Shao, Y. Peng, C. He, and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Trans.*, vol. 97, pp. 415–430, Feb. 2020.

[40] S. Girija and A. Joshi, "Fast hybrid PSO-APF algorithm for path planning in obstacle rich environment," *IFAC-PapersOnLine*, vol. 52, no. 29, pp. 25–30, 2019.

[41] X. Zhang and H. Duan, "An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning," *Appl. Soft Comput.*, vol. 26, pp. 270–284, Jan. 2015.

[42] G. C. Burdea and P. Coiffet, *Virtual Reality Technology*. Hoboken, NJ, USA: Wiley, 2003.

[43] C. Tu and L. Yu, "Research on collision detection algorithm based on AABB-OBB bounding volume," in *Proc. 1st Int. Workshop Educ. Technol. Comput. Sci.*, vol. 1, 2009, pp. 331–333.

[44] C. Tu and L. Yu, "Research on collision detection algorithm based on AABB-OBB boundin," in *Proc. 1st Int. Workshop Edu. Technol. Comput. Sci.*, vol. 1, Mar. 2009, pp. 331–333.
[45] D. T. Nguyen, K. H. Jung, K.-Y. Kwon, N. Ku, and J. Lee, "A kinematic collision box algorithm applied for the anti-collision system of offshore drilling vessels," *J. Mar. Sci. Eng.*, vol. 8, no. 6, p. 420, Jun. 2020.
[46] M. Caserta and S. Voß, "Metaheuristics: Intelligent problem solving," in *Matheuristics*. Boston, MA, USA: Springer, 2009, pp. 1–38.
[47] S. S. Rao, *Engineering Optimization: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2019.
[48] J. Crispim and J. Brandão, "Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls," *J. Oper. Res. Soc.*, vol. 56, no. 11, pp. 1296–1302, Nov. 2005.
[49] K. Sörensen and F. Glover, "Metaheuristics," *Encyclopedia of Operations Research and Management Science*, vol. 62. New York, NY, USA: Springer, 2013, pp. 960–970.
[50] 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.
[51] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer, 2006.
[52] L. Davis, *Handbook of Genetic Algorithms*. Brussels, Belgium: CumInCAD, 1991.
[53] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Jun. 1995, pp. 1942–1948.
[54] V. Kachitvichyanukul, "Comparison of three evolutionary algorithms: GA, PSO, and DE," *Ind. Eng. Manage. Syst.*, vol. 11, no. 3, pp. 215–223, Sep. 2012.
[55] P. Ouyang and V. Pano, "Comparative study of DE, PSO and GA for position domain PID controller tuning," *Algorithms*, vol. 8, no. 3, pp. 697–711, Aug. 2015.
[56] O. Serrano-Pérez, M. G. Villarreal-Cervantes, J. C. González-Robles, and A. Rodríguez-Molina, "Meta-heuristic algorithms for the control tuning of omnidirectional mobile robots," *Eng. Optim.*, vol. 52, no. 2, pp. 325–342, 2020.
[57] A. E. Eiben, *Introduction to Evolutionary Computing*, vol. 53. Berlin, Germany: Springer, 2003.
[58] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, Jun. 2000.
[59] M. Gen and R. Cheng, "A survey of penalty techniques in genetic algorithms," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jun. 1996, pp. 804–809.
[60] X. Liang, G. Meng, Y. Xu, and H. Luo, "A geometrical path planning method for unmanned aerial vehicle in 2D/3D complex environment," *Intell. Service Robot.*, vol. 11, no. 3, pp. 301–312, Jul. 2018.
[61] J. Li, S. Liu, B. Zhang, and X. Zhao, "RRT–A* motion planning algorithm for non-holonomic mobile robot," in *Proc. SICE Annu. Conf. (SICE)*, Sep. 2014, pp. 1833–1838.
[62] I. Buyuksalih, S. Bayburt, G. Buyuksalih, A. Baskaraca, H. Karim, and A. A. Rahman, "3D modelling and visualization based on the unity game engine–advantages and challenges," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, p. 161, 2017.
[63] *Intelligent and Mobile Robotics Group in the Department of Cybernetics of the Czech Technical University in Prague: Motion planning Maps*. Accessed: Aug. 25, 2020. [Online]. Available: http://imr.ciirc.cvut.cz/planning/maps.xml
[64] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proc. 8th Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2006, pp. 485–492.
[65] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, J. Álvarez-Gallegos, and M. Aldape-Pérez, "Bio-inspired adaptive control strategy for the highly efficient speed regulation of the DC motor under parametric uncertainty," *Appl. Soft Comput.*, vol. 75, pp. 29–45, Feb. 2019.
[66] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Proc. 3rd World Congr. Nature Biol. Inspired Comput.*, Oct. 2011, pp. 633–640.
[67] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.
[68] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 6, pp. 1–24, Oct. 2012.
[69] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 33, pp. 1–17, Apr. 2017.

GEOVANNI FLORES-CABALLERO received the B.S. degree in control and automation engineering from the National Polytechnic Institute, Mexico City, Mexico, in 2012, and the M.Sc. degree in computer technology from the Center for Innovation and Technological Development in Computing, CIDETEC, National Polytechnic Institute, Mexico City, in 2016, where he is currently pursuing the Ph.D. degree in robotic and mechatronic systems engineering. His current research interests include design and implementation of bio-inspired meta-heuristics for optimization and their application to engineering problems.

ALEJANDRO RODRÍGUEZ-MOLINA (Member, IEEE) received the B.S. degree in computer systems engineering from the Escuela Superior de Cómputo (ESCOM), Instituto Politécnico Nacional (IPN), in 2013, the M.Sc. degree in computer science from the Centro de Investigación y de Estudios Avanzados (CINVESTAV), IPN, in 2015, and the Ph.D. degree in robotics and mechatronic systems engineering from the Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC), IPN, in 2019. He is currently a full-time Professor with the Research and Postgraduate Division, Instituto Tecnológico de Tlalnepantla (ITTLA), Tecnológico Nacional de México (TecNM). His research interests are the design and implementation of bio-inspired meta-heuristics for optimization and their application to engineering problems.

MARIO ALDAPE-PÉREZ received the Ph.D. degree in computer science from the National Polytechnic Institute of Mexico, in 2011. He was the Head of the Research and Technology Innovation Department, CIDETEC, from 2013 to 2016. He is currently a Professor of Computer Architecture with CIDETEC, National Polytechnic Institute of Mexico. His current research interests include associative memories, soft computing, and FPGA implementation of high performance pattern classification algorithms. He is a member of the IEEE Computer Society and the ACM.

MIGUEL GABRIEL VILLARREAL-CERVANTES (Member, IEEE) received the B.S. degree in electronics engineering from the Veracruz Technologic Institute, Veracruz, Mexico, in 2003, and the M.Sc. and Ph.D. degrees in electrical engineering from the Center for Research and Advanced Studies, CINVESTAV, Mexico City, Mexico, in 2005 and 2010, respectively. He was the Head of both the Mechatronic Section with CIDETEC-IPN, from 2013 to 2016, and also the Head of ENRM-IPN, from 2012 to 2017. He is currently a Full Professor with the Postgraduate Department, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional (CIDETEC-IPN), Mexico City. His current research interests include mechatronic design based on optimization (mono and multiobjective optimization), bio-inspired meta-heuristics in the optimal mechatronic design, optimal tuning for the mechatronic system control based on meta-heuristic algorithms (offline and online strategies), and robotics. He is a member of the National System of Researchers and the Expert Network on Robotics and Mechatronics, IPN (ENRM-IPN).