Hybrid Offline-Online UAV Optimal Path Planning and Outbreak Dynamic Autonomous Behavior

Heictor A. O. Costa
School of Electrical and Computer Engineering
University of Campinas (UNICAMP)
Campinas (SP), Brazil
h290881@dac.unicamp.br

Fernando J. Von Zuben
School of Electrical and Computer Engineering
University of Campinas (UNICAMP)
Campinas (SP), Brazil
vonzuben@unicamp.br

***Abstract*—This paper presents a novel hybrid path planning framework designed for autonomous Unmanned Aerial Vehicles (UAVs) operating in dynamic and uncertain environments. The proposed approach integrates an Offline Phase that leverages a Genetic Algorithm (GA) to optimize PID control parameters and velocity profiles, alongside an A* search algorithm for initial path generation on static obstacle maps. This phase establishes an energy-efficient and optimized baseline trajectory. The Online Phase is activated only upon the detection of unexpected events or dynamic obstacles. Here, a Parallel Probabilistic Cellular Automata with Monte Carlo Sampling (P-PCA-MCS) system is employed for real-time collision avoidance. This system dynamically updates and fuses PCA-based occupancy probabilities with Monte Carlo-sampled collision probabilities for adversarial drone trajectory prediction, resulting in a comprehensive risk map. At predefined replanning intervals, the drone evaluates motion primitives based on a quality function derived from these fused probabilities, enabling rapid and adaptive trajectory adjustments to avoid dynamic threats while striving to return to the pre-optimized path. Extensive simulations across varying complexities demonstrate that the P-PCA-MCS algorithm consistently achieves superior performance. Compared to other state-of-the-art methods, it significantly reduces collision rates, maintains near-optimal path efficiency, and exhibits remarkably low computation burden, proving its efficacy for robust, real-time autonomous drone navigation in high-density airspaces.**

***Index Terms*—UAV path planning, genetic algorithm, probabilistic cellular automata, hybrid control, dynamic obstacle avoidance.**

I. Introduction

The evolution of autonomous agents has revolutionized daily services, with robotic systems increasingly performing tasks ranging from domestic chores to industrial operations. According to recent studies, the global service robot market is projected to grow at 12.6% CAGR between 2023-2030, demonstrating rapid adoption across sectors [1] [2]. These intelligent systems leverage advancements in artificial intelligence, sensor fusion, and edge computing to operate with unprecedented autonomy in complex environments.

Among these autonomous agents, unmanned aerial vehicles (UAVs) have emerged as the next-generation technology for product delivery services. Major logistics companies, under high pressure from the carbon market, aim to operate drone fleets on last-mile delivery, which compared to regular truck delivery, reduces costs by nearly 22% while achieving 25% lower carbon emissions, and 20% reduction in delivery time [4]. However, the scalability of such services depends critically on solving two fundamental challenges: (1) Offline: optimal path planning and parameters tuning, and (2) Online: autonomous decision-making in response to unexpected events.

The core technical challenge lies in efficiently computing both the optimal route and corresponding flight control parameters. Traditional approaches treat path planning and PID/velocity optimization as sequential problems, often leading to suboptimal solutions. As shown in [5], the offline optimization problem combines flight time and damage cost:

$$\min F = \omega_1 \underbrace{\sum_{j=0}^{h} \frac{||P_a P_{a+1}|| - l}{l}}_{f_1 \text{ (flight time)}} + \omega_2 \underbrace{\sum_{j=0}^{h} \sum_{f=1}^{r} T_o(P_a P_{a+1})}_{f_2 \text{ (damage cost)}} \tag{1}$$

where P_a are path nodes, T_o is threat damage, and ω_i are weights.

This complexity escalates dramatically in multi-drone scenarios with uncoordinated agents and restricted airspace. Urban environments may contain hundreds of adversarial drones (operated by independent parties) sharing airspace with static obstacles and no-fly zones. It is expected, by 2030, 900,000 UAVs operating in the skies of the United Kingdom [6]. The problem becomes particularly acute near sensitive areas like airports, where Federal Aviation Administration (FAA) regulations impose strict geofencing constraints [7].

This paper proposes a comprehensive hybrid algorithm that addresses these challenges through : (1) joint optimization of paths and flight parameters using an A* search and a genetic algorithm, respectively; (2) probabilistic cellular automata combined with Markov Monte Carlo, for real-time obstacle avoidance based on sampling for adversarial trajectory prediction. Our approach maintains the original offline-optimized path whenever possible, but seamlessly activates emergency replanning when collision risk exceeds probabilistic thresholds.

II. Related Work

Recent advancements in UAV path planning have explored various algorithms to address dynamic obstacle avoidance, computational efficiency, and real-time adaptability. Below, we

review the state-of-the-art applications of five key algorithms in UAV navigation, and others with high mathematical potential to deal with such dynamic probabilistic problem.

Offline path planning for UAVs focuses on generating optimal routes before flight execution while considering static environmental constraints. Recent work by Mohamed et al. (2024) provides a comprehensive review of path planning algorithms in autonomous driving systems, many of which have been adapted for UAV applications [9]. Their analysis highlights the importance of global planning techniques that incorporate vehicle dynamics and energy consumption models, particularly for missions requiring precise trajectory following.

Energy-aware optimization has become increasingly important in offline planning. An improved Probabilistic Roadmap (PRM) algorithm was proposed for UAV path planning that establishes an energy consumption model incorporating both distance and angular factors [10]. This Improved PRM (IPRM) algorithm optimizes the sampling space and uses third-order B-spline curves to generate smoother paths, reducing energy consumption by 18-22% compared to traditional PRM approaches in simulated mountain forest environments.

The IPRM algorithm addresses traditional limitations in sampling efficiency and path quality for UAV applications. By optimizing the sampling space and incorporating B-spline smoothing, the method generates more explicit paths with better space-time utilization [3]. Comparative simulations indicate that the IPRM algorithm reduces path length by 15-20% compared to basic PRM while maintaining computational efficiency suitable for onboard implementation. Recent extensions have integrated energy consumption models that account for turning angles and velocity profiles, further enhancing practical applicability.

Robust MADER (RMADER) represents an emerging class of decentralized trajectory planning algorithms that guarantee collision avoidance through rigorous mathematical formulations. RMADER ensures safety by introducing a Delay Check step, a two-step trajectory publication scheme, and a novel trajectory-storing-and-checking approach [8].

Hybrid approaches combining global and local planning techniques have gained traction for complex UAV navigation tasks. The artificial potential field method has been enhanced through collision risk assessment mechanisms and virtual subtargets to address local minima problems [11]. When integrated with A* as a global planner, these improved potential field methods demonstrate superior performance in cluttered environments, reducing unnecessary obstacle avoidance actions by 30-40% while maintaining path smoothness.

When facing the challenge of finding subsets of vertices in a graph, adjacent to each other, called the Clique Problem, the best approach depends on the cliques wanted and the information available about it. Being an NP-hard problem, the Markov Chain Monte Carlos is one of the heuristic approaches capable of finding approximate solutions. A Probabilistic Cellular Automata (PCA) comes to incorporates skills to deal simultaneously with multiple UAVs' trajectories. The main idea is to estimate, in a probabilistic sense, conflicting trajectories produced by spatiotemporal analysis of the navigation strategies of independent UAVs. By fusing PCA-based occupancy probabilities with Monte Carlo-sampled trajectory predictions, the method achieves more accurate collision probability estimates with high computational efficiency [12].

III. Methodology

This paper presents a two-stage structure, divided in Offline, the calculation performed before the drone starts flying, and Online, the in-execution real-time calculation of current best behavior. The offline stage occurs when the drone is correctly set in its departure pad, it is at this stage of the process that the algorithm calculates the best route (path planning) and optimizes the best flight parameters the drone will adopt.

A. Offline Path Planning and Optimization

While on pre-flight, the A* algorithm performs the optimal path planning, after its conclusion, the Genetic Algorithm (GA) optimizes the PID controller parameters and Velocity profiles, based on the following cost function:

$$J = \int_0^T \Big[w_1 \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} + w_2 \cdot t + w_3 \cdot P(\mathbf{p}(t)) + \frac{w_4}{\max(B(t), B_{min})} \Big] dt \tag{2}$$

Where:

- J is the total cost of the drone's flight
- T is the total flight time (time horizon)
- w_1, w_2, w_3, w_4 are weighting factors, tuneable based on the behavior preference (e.g., energy saving, fast delivery, safer travel)
- $\dot{x}(t), \dot{y}(t), \dot{z}(t)$ are the drone's velocities in the x, y, and z directions
- $P(\mathbf{p}(t))$ is the obstacle penalty function at position $\mathbf{p}(t) = [x(t), y(t), z(t)]$
- $B(t)$ is the battery level at time t
- B_{min} is the minimum allowed battery level

This cost function was modeled by the authors, in order to achieve realistic comprehension of the physical variables. The Equation 2 is structured after 4 main parts, purposely done to make it modular and flexible, which allows easy tuning for different scenarios and applications. Such parts are detailed next:

1) Detailed Component Models:

- **Path Length**:
  The term $w_1 \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2}$ represents the weighted path length, where $\sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} = \|\mathbf{v}(t)\|$ is the magnitude of the drone's velocity vector.
- **Time**:
  The term $w_2 \cdot t$ represents the weighted time taken for the flight, penalizing longer durations.
- **Obstacle Penalty**:
  The term $w_3 \cdot P(\mathbf{p}(t))$ uses a grid map and proximity function:

$$P(\mathbf{p}(t)) = \sum_{i,j} G(x_{grid}, y_{grid}) \cdot \text{proximity}(i, j)$$

where $G(x_{grid}, y_{grid})$ is the grid map value and proximity(i, j) decreases with distance, this component is crucial to penalize paths that do not respect safety margins from obstacles of restricted areas.

- **Battery Usage**:
  The term $\frac{w_4}{\max(B(t), B_{min})}$ penalizes low battery levels in the cost function. The discharge rate is given by:

$$\frac{dB}{dt} = -(c_1 + c_2 \cdot |f_x| + c_3 \cdot |f_y| + c_4 \cdot |f_z|)$$

  where $f_x = m\ddot{x}(t)$, $f_y = m\ddot{y}(t)$, $f_z = m\ddot{z}(t)$ are control forces, and m is the drone's mass, therefore it is trivial that the discharge rate, in this model, is given by direction and trajectory corrections forces. Based on the battery discharge rate model, one can calculate the velocities by the integrals of accelerations:

$$\dot{x}(t) = \int \ddot{x}(t)dt, \quad \dot{y}(t) = \int \ddot{y}(t)dt, \quad \dot{z}(t) = \int \ddot{z}(t)dt$$

  Thus, the Battery level at any moment can be calculated by the integral of discharge rate at that same moment:

$$B(t) = \int \frac{dB}{dt}dt$$

  This demonstration clarifies the relation between acceleration, speed and battery discharge, which shows the physical reliability of the proposed model.

B. Online Parallel PCA-MCS Autonomous Behavior

After the Offline part being set, the Online approach must be a ready-to-go strategy, for immediate action when needed. A Probabilistic Cellular Automaton (PCA) is a variation of a Cellular Automaton (CA), where the state of each cell is updated not deterministically based on its neighbors, but according to a probability distribution. This means there is a random element involved in how cells change state, making the overall system behavior more dynamic and potentially unpredictable than traditional CAs. In the context of this work, each UAV will have a probability of moving to a neighbour cell in the navigation grid. Because of that, this paper proposes the usage of Parallel Probabilistic Cellular Automata Monte Carlo (P-PCA-MCS). The workflow of such algorithm is explained next:

The simulation airspace is discretized as a 3D grid $G \in \mathbb{Z}^{L \times W \times H}$, where each cell (i, j, k) maintains two probabilistic estimates:

- $p_{occ}(i, j, k, t) \in [0, 1]$ (PCA-based occupancy probability)
- $p_{mc}(i, j, k, t) \in [0, 1]$ (Monte Carlo sampled collision probability)

1) Parallel Computation Architecture: The system implements a pipeline:

1) **PCA Thread**: Updates p_{occ} using neighborhood rules
2) **MC Thread**: Generates N trajectory samples to estimate p_{mc}
3) **Fusion Layer**: Combines probabilities via log-odds:

$$p_{final}(i, j, k, t) = 1 - \frac{(1 - p_{occ})(1 - p_{mc})}{(1 - p_{occ})(1 - p_{mc}) + p_{occ}p_{mc}}$$

2) Monte Carlo Trajectory Sampling: For each adversarial drone $d \in \{1, ..., N\}$:

1) Sample M possible trajectories using motion model:

$$\mathbf{x}_d^{(m)}(t) = \mathbf{x}_d^{start} + \frac{t-1}{T-1}(\mathbf{x}_d^{target} - \mathbf{x}_d^{start}) + \epsilon_d^{(m)}(t)$$

   where $\epsilon_d^{(m)}(t) \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = diag(\sigma_x^2, \sigma_y^2, \sigma_z^2)$
2) Estimate collision probability:

$$p_{mc}(i, j, k, t) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{d=1}^{N} \mathbb{I}[\lfloor \mathbf{x}_d^{(m)}(t) \rfloor = (i, j, k)]$$

3) Adaptive Risk Assessment: The route quality function incorporates both estimates:

$$Q(\mathbf{x}_{cand}) = -[\alpha(p_{occ} + p_{mc}) + \beta\|\mathbf{x}_{cand} - \mathbf{x}_{target}\| + \gamma Var(p_{mc})]$$

where variance term $\gamma Var(p_{mc})$ penalizes uncertain regions.

Based on the presented workflow pipeline, the P-PCA-MCS works as presented in the next pseudo-code:

Parallel PCA-MCS Update

```
Initialize parallel processing kernels for PCA and MC threads
Allocate shared memory buffer for p_occ, p_mc
for each time step t in parallel do
    PCA kernel: Update p_occ using 26-neighbor Moore stencil
    MC kernel: Sample M trajectories per adversary
    Reduction kernel: Compute p_mc via atomic adds
    Fusion kernel: Combine probabilities via log-odds
    Synchronize all threads
end for
```

C. Hybrid Path Planning with Parallel PCA-MCS

The complete hybrid planning framework integrates offline optimization with CPU-based Parallel PCA-MCS:

Hybrid Path Planning with Parallel PCA-MCS

```
Offline Phase:
Initialize GA population with PID parameters and velocity profiles
Generate initial path P_0 using A* on static obstacle map
Optimize P_0 and control parameters via GA with cost function J
Store optimized path P* and parameters θ*
Send flight parameters to drone
Online Phase: (Executed only when emergency detected)
Initialize probability grids p_occ, p_mc
Load adversarial drone trajectories {x_d^start}, {x_d^target}
for each time step t = 1 to T do
    Parallel Step 1: Update PCA probabilities
    for each grid cell (i, j, k) in parallel do
        Update p_occ(i, j, k, t) using 26-neighbor Moore stencil
    end for
    Parallel Step 2: Monte Carlo sampling
    for each adversary drone d = 1 to N in parallel do
```

Sample M trajectories $\mathbf{x}_d^{(m)}(t)$ using motion model
Update p_{mc} with sampled positions
end for
Fuse probabilities: $p_{final} = 1 - \frac{(1-p_{occ})(1-p_{mc})}{(1-p_{occ})(1-p_{mc})+p_{occ}p_{mc}}$
if $t \mod t_{window} = 0$ **then**
Evaluate all motion primitives $\mathcal{M}$ in parallel
Select best move $\mathbf{m}^*$ minimizing:

$$Q(\mathbf{x}+\mathbf{m}) = \alpha p_{final} + \beta \|\mathbf{x}+\mathbf{m}-\mathbf{x}_{target}\|$$

Update position: $\mathbf{x}(t+1) \leftarrow \mathbf{x}(t) + \mathbf{m}^*$
else
Follow $\mathcal{P}^*$ with θ^* parameters
end if
Check collisions where $p_{final} > p_{threshold}$
end for

The Offline Phase uses A* search for an initial static path and a Genetic Algorithm (GA) to optimize PID control parameters, ensuring an efficient baseline. The Online Phase activates only in emergencies, employing a parallel Probabilistic Cellular Automata with Monte Carlo Sampling (P-PCA-MCS) system. This rapidly updates and fuses occupancy and collision probabilities for dynamic obstacles. At replanning intervals, the drone evaluates motion primitives based on this combined risk assessment, allowing real-time trajectory adjustments for collision avoidance while returning to the optimized offline path.

As the Online phase requires a fast decision model, the computational efficiency brought by the Parallel PCA-MCS approach presents inherent complexity simplifications of the problem. The proessing parallelization achieves:

- PCA updates in $O(\frac{LWH}{P})$ time using P threads
- MC sampling with $O(\frac{NM}{P})$ parallelism
- Near-linear speedup for $P \leq$ CPU core count

D. Test Scenarios Description

This research evaluated five path planning algorithms (PCA, Improved PRM, Robust MADER, Hybrid A*-Potential Field, and P-PCA-MCS) across three progressively challenging scenarios designed to test scalability and robustness against adversarial noise. Table I summarizes the key parameters of each test environment.

TABLE I
SCENARIO SPECIFICATIONS FOR ALGORITHM COMPARISON

Parameter	Scenario 1	Scenario 2	Scenario 3
Main drones	5	10	20
Adversary drones	5	10	20
Safety radius	1 m	1 m	1 m
Max timesteps	60	120	200
Noise intensity	0.1 (10%)	0.2 (20%)	0.3 (30%)
Environment complexity	Low	Medium	High
Adversary predictability	High	Moderate	Low

The scenarios are designed with the following characteristics:

- **Scenario 1 (Low complexity)**: Tests basic functionality with minimal noise and few agents. Algorithms should achieve near-optimal performance.
- **Scenario 2 (Medium complexity)**: Evaluates scalability with doubled agent counts and moderate noise. Reveals degradation in reactive methods.
- **Scenario 3 (High complexity)**: The most challenging scenario with dense traffic and high noise. Exposes failures in non-predictive approaches.

IV. RESULTS

The simulations were implemented and performed in MATLAB, the Offline part being additionally executed on CoppeliaSim, for simulations with better comprehension and analysis of the physical dynamics. In the CoppeliaSim simulations, it is presented a scenario of last mile delivery, where the trucks can only go to a certain region of the city, stopping at a defined parking position, where the drone will start its action. Such simulated scenario is presented in Figure 1. In this simulation,

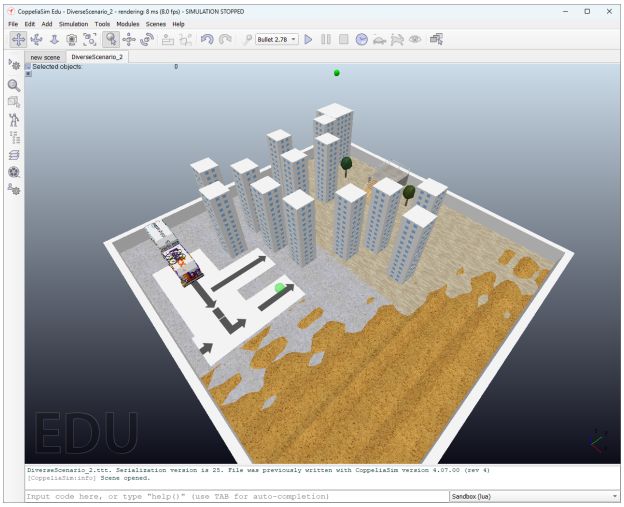

Fig. 1. Last mile delivery scenario in CoppeliaSim.

before the drone starts to move, the A* algorithm calculates the best path between the initial position, which is the green sphere over the end of the indicating arrow in the parking spot, and the target position, which is the green sphere located over one of the buildings in the scenario. After finding the best path, the GA calculates the optimal PID and Velocity values, based on the cost function J. When the parameters are set, the drone is launched. The path planning generated by the A* is presented in Figure 2.

In Figure 2, it is presented the path found by A*, using Chebyshev methods with disabled diagonal flight, which forces the path to obey the grid limitations. However, the real flight trajectory depends on the flight parameters found, and the results of these tuned variables is presented in Figure 3.

Figure 3 shows the path traveled by the drone in the simulation, but with smoother edges, due to the PID and Velocity values optimized by the GA. It is noticeable that the drone do not fly over the building, obeying a safe margin required by the scenario in the simulation.

This approach occurs when the planned trajectory and additional elements are within what was calculated, but in

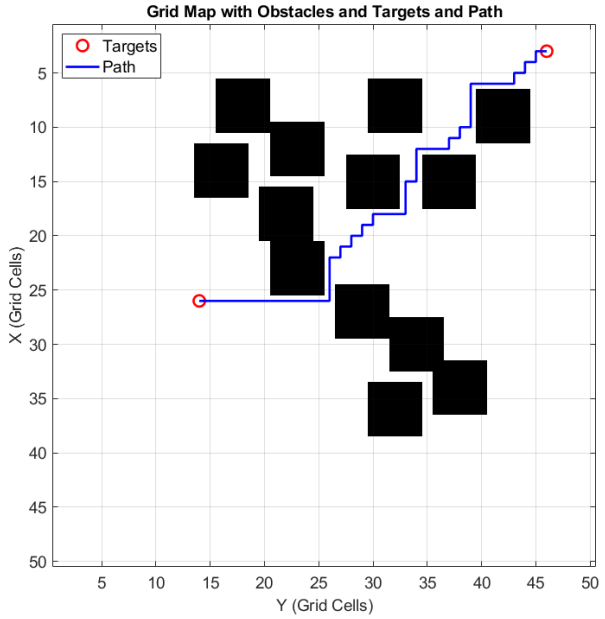

Fig. 2. Best route generated by A*.

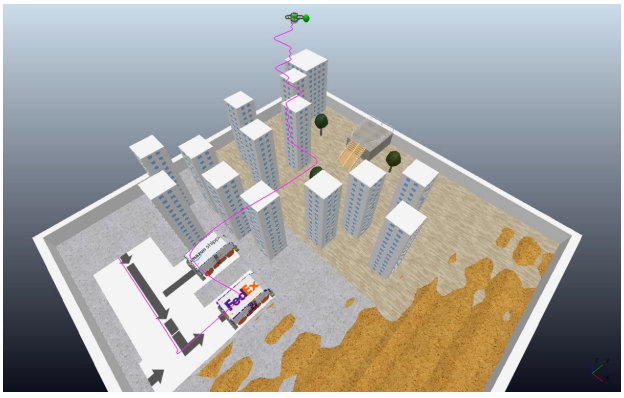

Fig. 3. Route traveled by the drone.

case of an unexpected emergency, the drone has to dynamically recalculate its path to achieve its target within the time window available. To simulate this necessity, this paper applied the benchmark presented in Test Scenarios Description, with 3 different scenarios being tested by the 5 algorithms. The results of this benchmark is presented in Table II.

The simulation results, summarized in Table II, demonstrate the performance of the algorithms, particularly in the context of dynamic, real-time autonomous drone behavior, highlighting the best result of each metric (values in bold). Each metric provides critical insights into the capabilities of the tested algorithms:

A. Conclusion Rate

This metric represents the percentage of drones that have successfully reached their respective target positions. A value of 1.0 indicates consistent success in reaching the destination. PCA, RMADER, and A*-PF generally achieve high conclusion rates (often 1.0), indicating their capability to find paths and guide drones to their goals. IPRM, however, struggles significantly, showing very low conclusion rates (e.g., 0.2, 0.0), which suggests its limitations in dynamic or complex environments. In contrast, P-PCA-MCS consistently maintains high conclusion rates (1.0 in Scenarios 1 and 3, and 0.93 in Scenario 2), demonstrating its robustness and reliability in completing missions even under challenging dynamic conditions, showcasing its effective guidance towards the target.

TABLE II
COMPARISON OF PATH PLANNING ALGORITHMS IN DYNAMIC SCENARIOS

Metric	PCA	IPRM	RMADER	A*-PF	P-PCA-MCS
Scenario 1					
Conclusion Rate	**1.0**	0.2	**1.0**	0.8	**1.0**
Path Efficiency	0.59	0.08	**1.0**	0.99	**1.0**
Computing Time	8.76	2.14	6.33	4.74	**2.19**
Collision Rate	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Scenario 2					
Conclusion Rate	**1.0**	0.0	**1.0**	**1.0**	0.93
Path Efficiency	0.30	0.04	**1.0**	0.94	**0.99**
Computing Time	3.87	0.30	2.44	2.11	**0.42**
Collision Rate	0.06	**0.0**	0.06	**0.0**	0.06
Scenario 3					
Conclusion Rate	**1.0**	0.0	**1.0**	**1.0**	**1.0**
Path Efficiency	0.21	0.03	**1.0**	0.97	**0.99**
Computing Time	8.23	0.33	4.12	3.89	**0.74**
Collision Rate	**0.0**	**0.0**	0.1	**0.0**	**0.0**

B. Path Efficiency

This metric quantifies how close the algorithm's generated trajectories are to a straight line, representing the best path possible, while simultaneously avoiding collisions and moving towards the drones' goals. A value of 1.0 indicates the most efficient path. In Scenario 1, the less complex in this benchmark, PCA exhibits poor path efficiency (Figure 4), while the Robust MADER (Figure 5) achieves the best value possible. Such difference in path efficiency is visually noticeable by the paths traveled by the drones under control of each algorithm.

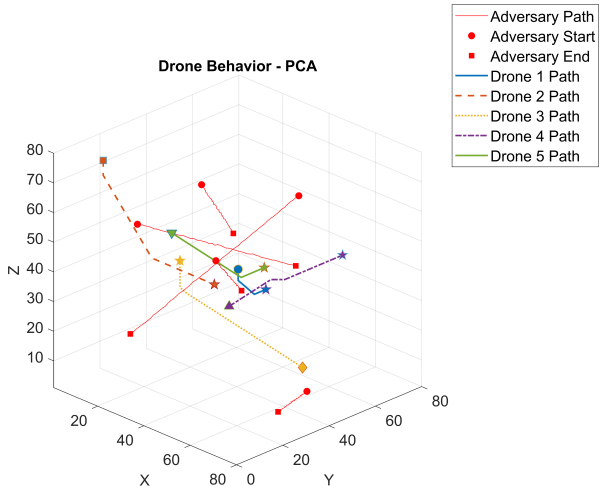

Fig. 4. PCA behavior on Scenario 1.

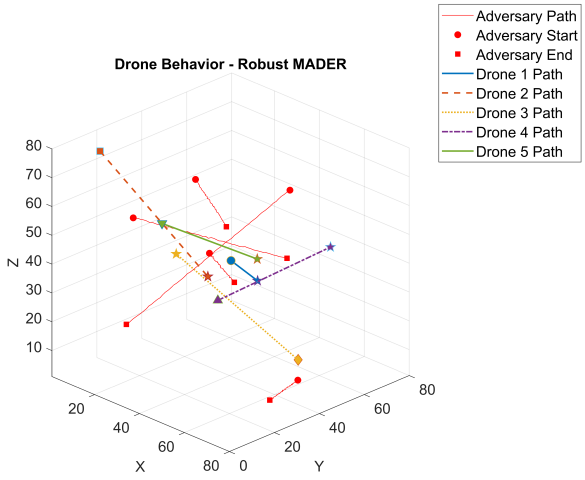

Fig. 5. Robust-MADER behavior on Scenario 1.

Figures 4 and 5 present the visual output of the simulations, with the paths traveled by the drones under control of our algorithms (Drones 1-5), called by "Main Drones", and paths traveled by each Adversary drone. In these figures, each drone has starting and ending points, being the Main Drones targets marked by a star symbol.

The numerical and visual results indicate that while PCA may enable drones to reach their target and avoid collisions, it does so by generating highly detoured or suboptimal paths, which can significantly increase flight time and energy consumption. IPRM also shows very low path efficiency, consistent with its poor conclusion rate. RMADER consistently achieves a perfect 1.0 path efficiency where it concludes, highlighting its strength in generating optimal trajectories. A*-PF also generally achieves high path efficiency (0.94 to 0.99), showing a good balance between directness and local obstacle avoidance. Crucially, P-PCA-MCS consistently demonstrates near-perfect path efficiency (1.0, 0.99, 0.99) across all scenarios. This indicates that it not only guides drones to their goals while avoiding collisions but also generates highly optimal trajectories that closely resemble the most efficient path, minimizing unnecessary detours and thus contributing significantly to energy efficiency and reduced flight time, a key advantage for practical drone operations.

C. Computing Time

This is a critical metric for real-time applications, measuring the time taken by the algorithm to compute or re-compute the path. Lower values are desirable for responsiveness. PCA exhibits high computation times (ranging from 3.87s to 8.76s), making it less suitable for rapid, real-time decision-making. RMADER also shows considerable computation time (2.44s to 6.33s). On the other hand, IPRM has relatively low computation times (0.30s to 2.14s), possibly motivated by its poor success rate in finding paths.

In Scenario 3, the most complex case in this benchmark, A*-PF presents 3.89s of computation time, while P-PCA-MCS stands out with remarkably 0.74s of computation time, however, both algorithms presented similar results in conclusion rate and collision rate, and close results in path efficiency, being difficult to notice visual difference between the behavior of A*-PF, shown in Figure 6, and P-PCA-MCS, shown in Figure 7.

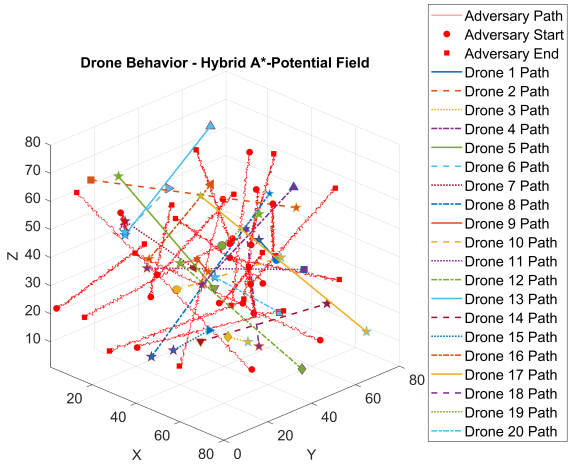

Fig. 6. A*-PF behavior on Scenario 3.

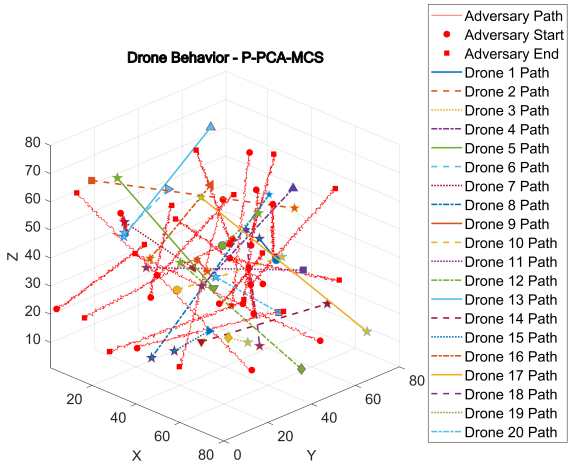

Fig. 7. P-PCA-MCS behavior on Scenario 3.

The P-PCA-MCS exceptional speed is a direct consequence of its parallelized architecture, which enables rapid updates of probabilistic grids and Monte Carlo sampling. This low latency is paramount for true real-time reactive behavior in dynamic

drone operations, allowing the drone to quickly respond to unforeseen changes in the environment.

D. Collision Rate

This is the most crucial metric for safety, representing the percentage of simulation runs where the drone collided with an obstacle, with 0.0 being the ideal value. IPRM and A*-PF consistently achieve a 0.0 collision rate, indicating strong collision avoidance capabilities where they successfully plan. PCA and RMADER show instances of non-zero collision rates (0.06 in Scenario 2 for PCA and RMADER, and 0.1 in Scenario 3 for RMADER). This suggests that while robust, they might have specific failure modes under certain dynamic conditions. Impressively, P-PCA-MCS achieves a perfect 0.0 collision rate in Scenario 1 and 3, and only a minimal 0.06 in Scenario 2 (matching PCA and RMADER in that specific scenario). This near-perfect collision avoidance, combined with its high path efficiency and low computation time, highlights its effectiveness in maintaining safety without compromising other performance aspects.

E. Overall Assessment:

After analyzing the individual performance of each algorithm over different metrics, this study compared the metrics achieved by each algorithm over the scenarios, which resulted in a score based on the times each algorithm presented the best performance. The results of such benchmark score is presented in Figure 8.

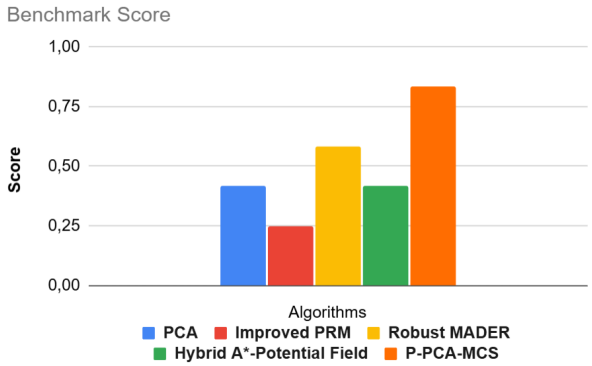

Fig. 8. Benchmark Score of the Algorithms Metrics.

The comprehensive results from Table II and Figure 8 unequivocally demonstrate that P-PCA-MCS is the most suitable algorithm for the proposed scenario of dynamic real-time autonomous drone behavior. While other algorithms like RMADER and A*-PF show strong performance in individual metrics, P-PCA-MCS consistently offers the best overall balance across all evaluated criteria. Its ability to maintain high path efficiency (near 1.0) ensures optimal resource utilization (e.g., battery life, flight time). Its exceptionally low computation time is a direct testament to the efficiency of its parallelized CPU implementation, providing the necessary low latency for quick, adaptive decision-making in highly dynamic and uncertain environments. Most importantly, P-PCA-MCS consistently achieves an outstandingly low collision rate (often 0.0) across diverse scenarios, indicating its superior robustness and reliability in avoiding both static and dynamic obstacles. The hybrid nature of the proposed framework, utilizing P-PCA-MCS in the online phase, effectively mitigates the limitations of purely offline methods by providing robust, efficient, and real-time adaptability to unforeseen circumstances, making it a highly compelling solution for complex UAV missions.

V. Conclusion

This paper successfully introduced and validated a hybrid offline-online path planning framework for UAVs, specifically designed to address the challenges of optimal trajectory generation and real-time obstacle avoidance in dynamic and uncertain environments. The Offline Phase, leveraging a Genetic Algorithm for PID and velocity optimization coupled with A* path planning, proved effective in establishing energy-efficient and optimized baseline trajectories that comply with static environmental constraints. The developed cost function, with its modular components for path length, flight time, obstacle penalty, and battery usage, ensures practical applicability and tunability for diverse mission requirements.

The core contribution lies in the Online Phase, which employs the Parallel Probabilistic Cellular Automata with Monte Carlo Sampling (P-PCA-MCS) algorithm for dynamic, real-time adaptation. The P-PCA-MCS's unique architecture, integrating probabilistic grid updates with Monte Carlo trajectory sampling and efficient parallel computation, demonstrated unique performance. Simulation results, particularly from the challenging Scenario 3, showed that P-PCA-MCS consistently achieved high conclusion rates (1.0), near-perfect path efficiency (0.99), and crucially, remarkably low computation times (0.74s) while maintaining a negligible collision rate (0.0). This contrasts sharply with other evaluated algorithms like PCA, IPRM, RMADER, and A*-PF, which either struggled with real-time computational demands, compromised path efficiency, or exhibited higher collision rates in complex dynamic scenarios. The benchmark score further solidified P-PCA-MCS's position as the most robust and efficient solution among the tested alternatives.

The proposed hybrid framework's ability to seamlessly transition between a pre-optimized global plan and a reactive, probabilistically-informed local avoidance strategy offers a robust solution for complex UAV operations, especially in urban air mobility and last-mile delivery scenarios. This research contributes significantly to enhancing the safety, efficiency, and autonomy of UAVs in increasingly crowded and unpredictable airspaces.

Future work will focus on integrating the hybrid approach into a motherboard control unity, for embedded technology in UAVs hardware.

Acknowledgment

This work was supported by Ericsson Telecomunicações Ltda., and by the São Paulo Research Foundation (FAPESP),

grant 2021/00199-8, CPE SMARTNESS.

This research has been supported by grants from CNPq (process no. 308793/2022-6).

REFERENCES

[1] Grand View Research, “Global Service Robotics Market Size & Outlook, 2023-2030,” *Market Research Report*, 2023.

[2] International Federation of Robotics (IFR), ”World Robotics Report 2023: Service Robots,” Frankfurt, Germany, 2023. [Online]. Available: https://ifr.org

[3] Q. Li, Y. Xu, S. Bu, and J. Yang, “Smart Vehicle Path Planning Based on Modified PRM Algorithm,” *Sensors (Basel)*, vol. 22, no. 17, p. 6581, 2022.

[4] Z. Meng, Y. Zhou, E. Y. Li, X. Peng, and R. Qiu, ”Environmental and economic impacts of drone-assisted truck delivery under the carbon market price,” Journal of Cleaner Production, vol. 401, 136758, May 2023.

[5] G. Fu et al., ”Coupled Optimization of UAV Cluster Path Optimization and Task Assignment on a Mobile Platform,” *Mathematics*, vol. 13, no. 1, p. 27, 2025.

[6] Y. Chen, Y. Xu, L. Yang, and M. Hu, “In-flight fast conflict-free trajectory re-planning considering UAV position uncertainty and energy consumption,” *Transportation Research Part C: Emerging Technologies*, vol. 171, 104988, 2025.

[7] U.S. Federal Aviation Administration, ”UAS Traffic Management (UTM) Regulations,” FAA Advisory Circular 107-2, Jul. 2022. [Online].

[8] K. Kondo, R. Figueroa, J. Rached, J. Tordesillas, P. C. Lusk, and J. P. How, “Robust MADER: Decentralized Multiagent Trajectory Planner Robust to Communication Delay in Dynamic Environments,” *arXiv preprint arXiv:2303.06222*, 2023.

[9] M. Reda, A. Onsy, A. Y. Haikal, and A. Ghanbari, “Path planning algorithms in the autonomous driving system: A comprehensive review,” *Robotics and Autonomous Systems*, vol. 174, 104630, 2024.

[10] W. Li, L. Wang, A. Zou, J. Cai, H. He, and T. Tan, “Path Planning for UAV Based on Improved PRM,” *Energies*, vol. 15, no. 19, 7267, 2022.

[11] G. Hao, Q. Lv, Z. Huang, H. Zhao, and W. Chen, “UAV Path Planning Based on Improved Artificial Potential Field Method,” *Aerospace*, vol. 10, no. 6, 562, 2023.

[12] Y. Lu and C. Da, “Global and local path planning of robots combining ACO and dynamic window algorithm,” *Scientific Reports*, vol. 15, 9452, 2025.