

Multi-UAV Path Planning and Collision Avoidance: Algorithms and Emerging Trends

Husayn Merchant^{1*}, Sananjay Biswas²

¹American School of Bombay, Mumbai, India

²Pion Academy and Research Centre, Mumbai, India

Email: *husaynm10@gmail.com

How to cite this paper: Merchant, H. and Biswas, S. (2026) Multi-UAV Path Planning and Collision Avoidance: Algorithms and Emerging Trends. *World Journal of Engineering and Technology*, 14, 840-868. <https://doi.org/10.4236/wjet.2026.144052>

Received: August 14, 2026

Accepted: October 16, 2026

Published: October 19, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). <http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Multi-UAV routing requires integration of route planning, time-separation, vehicle dynamics, sensing and communication uncertainties, and computational resources onboard. This critical review provides a comparison of graph search, multi-agent path finding, sampling, optimization, reactive control, safety filters, and multi-agent learning techniques in terms of explicit assumptions and types of evidence supporting their use. Optimization techniques allow generating dynamically feasible trajectories only when the underlying dynamics and constraints are correctly modeled, and a solution is found by the solver; initialization, nonconvexity, and computation may preclude this possibility. Reactive approaches provide fast empirical reaction, yet can experience deadlock problems, whereas learned policies are still hard to certify beyond test environments. We separate guarantees derived from assumptions, conditional properties, and empirical results and contrast multi-UAV flights and simulations to transferable evidence. Specification of the search strategy and audit of the selected literature reveal limitations of the selection process. Our review identifies calibrated safety margins and a practical benchmark for testing collision avoidance and liveness in multi-UAV systems.

Keywords

Multi-UAV Systems, Cooperative Path Planning, Collision Avoidance, Trajectory Optimization, Safety-Critical Control, Multi-Agent Learning

1. Introduction

Multi-UAV systems can inspect larger territories, provide sensing diversity, tolerate individual failures and execute distributed missions faster than a single vehicle. These advantages help to perform inspection, precision farming, emergency

mapping, logistics, environmental monitoring, communications relay, search and rescue [1]. Moreover, the multi-UAV systems introduce dependencies between the navigation decisions of the vehicles, which turns navigation into a coupled problem where each vehicle is simultaneously a planner, an additional constraint, a communication node and a source of aerodynamic disturbances. Thus, the plan of one UAV cannot be determined independently of the plans of other trajectories [2]-[7]. The terms ‘path planning’, ‘trajectory planning’, ‘coordination’ and ‘collision avoidance’ are often used synonymously, despite important differences in meaning. Namely, a path is geometric in nature while a trajectory includes timing information and must comply with velocity, acceleration, jerk, attitude, thrust and sometimes downwash constraints. Collision avoidance can be planned before the flight, resolved during it or guaranteed by a safety layer that provides minimal change to the nominal controller. Multi-agent path finding (MAPF) mostly uses discrete-time-expanded graphs while multi-robot motion planning and UAV trajectory optimization use continuous state and control spaces. Thus, a strong review should connect these communities without losing their distinct characteristics [8]-[15].

Recently, the progress has been very fast. EGO-Swarm implemented decentralized asynchronous onboard perception and trajectory sharing to navigate the swarm in the complex environment [16]. Joint spatial-temporal optimization allowed palm-sized micro-UAVs to navigate the natural environment using limited onboard sensing and computation [17]. MADER used polynomial outer approximation of trajectory segments and separating hyperplanes to incorporate conflict resolution into decentralized optimization [18] while Robust MADER (RMADER) added delay checks, two-step publication and trajectory verification to preserve deconfliction despite bounded communication delay [19]. Meanwhile, MARL, GNNs, safe learning and bio-inspired spiking neural networks provided a wide range of applications of adaptive decentralized control [20]-[25]. However, evidence is scattered. Many papers provide path length and convergence plots but lack minimum separation distance, time-to-collision, constraint violations, computational and communication costs or failure modes. Static 2D environments are often used to illustrate the results of dynamic 3D operation. Simulation of zero collisions is not equal to formal proof; the proof of correctness of an algorithm for double-integrator point agents is not automatically valid for quadrotors with estimation errors and actuator saturation; and scalability obtained by parallel offline planning is not the same as the online real-time scalability. All these distinctions motivate the current critical analysis.

This paper makes four contributions. Firstly, it presents the framework of the multi-UAV navigation as a space-time optimization problem that combines mission goals, geometry, dynamics, uncertainty, communication and safety constraints. Secondly, it classifies the algorithm families specifying their information assumptions and guarantees. Thirdly, it compares the selected algorithms regarding their performance and summarizes relevant quantitative evidence when pos-

sible. Finally, it proposes a unified validation protocol and a research agenda of learning, communication-aware planning, heterogeneity and low-altitude traffic integration. **Figure 1** organizes the review around five interacting algorithmic families: graph search and multi-agent path finding, sampling and metaheuristics, reactive collision avoidance, optimal control and safety filters, and learning-based methods. The lower layer identifies architecture, information availability, performance objectives, and evidence as cross-cutting considerations. This organization enables methods to be compared according to their assumptions and deployment conditions rather than by algorithm name alone.

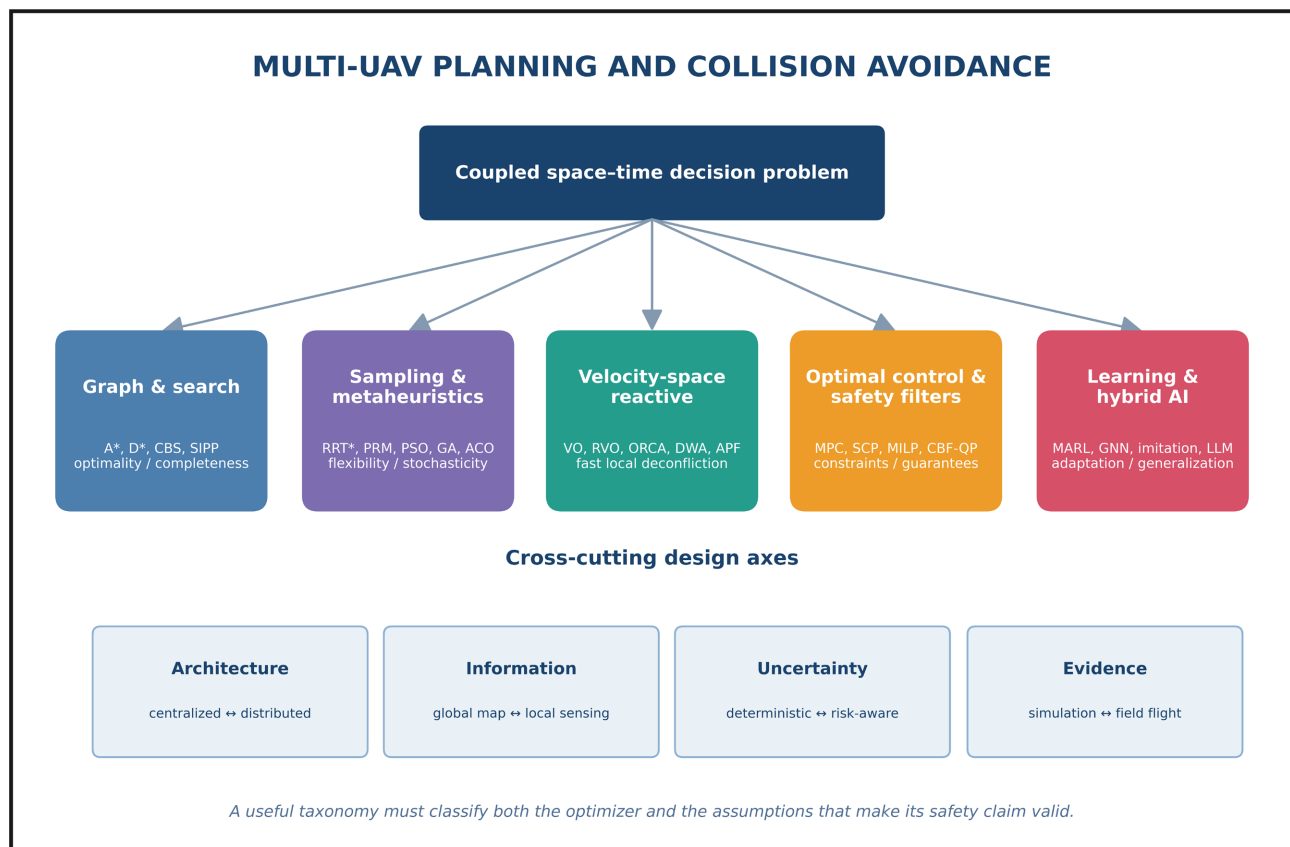


Figure 1. Unified taxonomy for multi-UAV path planning and collision avoidance. The classification distinguishes algorithmic families from cross-cutting architectural and evidential assumptions. Original schematic created for this review.

2. Review Methodology and Scope

In this critical narrative review of multi-UAV path planning and collision avoidance, a retrospective analysis of the sources cited in the manuscript has been carried out. The present paper is not a systematic review or meta-analysis. The bibliography lists 70 references. Of these, 51 different references are used in the 14 algorithm families listed in **Table 1**, while 21 references are cited in the 14 systems described in **Table 2**. In total, the two tables contain references to 55 different publications; the other 15 references give the background and/or context evidence. These are citations within the manuscript; this paper neither performs any

bibliographic database search nor uses the results of systematic screening. Since the search export files and screening decisions are no longer available, the historical retrieval statistics, exclusion counts, and the PRISMA flow cannot be reconstructed.

An update of the current review can be reproduced using IEEE Xplore, Scopus, Web of Science Core Collection, ACM Digital Library, and OpenAlex from their inception until the date of the update, without any restriction on the publication year and language. The following set of keywords should be applied to the title, abstract, and keywords fields: “multi-UAV” OR “multiple UAV” OR “multi-drone” OR “drone swarm” OR “quadrotor swarm” OR “multi-robot” OR “multi-agent path finding” AND (“path planning” OR “trajectory planning” OR “motion planning” OR “collision avoidance” OR “conflict resolution” OR “deconfliction”) AND (“dynamic obstacle” OR “inter-agent” OR “inter-vehicle” OR “coordination” OR “communication delay” OR “safety”), (“UAV” OR “quadrotor”) AND (“model predictive control” OR “control barrier function” OR “velocity obstacle” OR “ORCA” OR “reinforcement learning”) AND (“multi-agent” OR “swarm” OR “collision avoidance”). Also, the reference lists of selected articles can be used to identify additional relevant studies. For each database search, the date of search, the interface name and version, the exact query used, and the retrieval count and export file should be saved. The duplicates can be removed by using the DOI or the title. Further, the records should be screened by titles and abstracts and reviewed in full text according to the specified eligibility criteria. A record of the decisions on inclusion/exclusion and the reason for excluding full texts will allow conducting an audit of the final set of studies. The above description applies to a future update; there is no implication of recovering historical screening data. The sources supporting reported outcomes were classified as direct multi-UAV flight (UF), multi-UAV simulation or hardware-in-the-loop testing without flight (US), transferable research with multi-robots without direct UAV validation (MR), and single-UAV planning (SU). If a publication reported more than one validation method, then for every outcome a category was chosen based on the validation experiment. The claims regarding the safety and performance of the algorithms have been separately classified as formal guarantees under certain assumptions (F), conditional properties depending on feasibility and valid model/uncertainty bound (C), or an empirical result (E).

This comparison analyzed the presence of eight reporting criteria: the representation of planning, coordination architecture, the environment, dynamics, safety approach, type of guarantee, scalability of computations and communications, and the validation. For each criterion, the source information was coded as reported (R), partially reported (P), or not reported (NR). This coding was done based on explicit methodological and experimental information contained in the cited publication; a missing capability was not assumed from the algorithm class. There was no aggregation into a numerical quality score since the criteria have no standardized weighting. **Table 1** and **Table 2** include the results of this descriptive

assessment along with the comparisons of methods and systems.

3. Problem Formulation and Evaluation Dimensions

3.1. Coupled Space-Time Trajectory Model

Mathematical representation of cooperative UAV planning

For a system of N unmanned aerial vehicles (UAVs), define the agent set as $\mathcal{N} = \{1, \dots, N\}$. The state and control input of UAV i are denoted by $\mathbf{x}_i(t) \in \mathbb{R}^{n_i}$ and $\mathbf{u}_i(t) \in \mathbb{R}^{m_i}$, respectively. The motion of each vehicle is governed by

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}_i(\mathbf{x}_i(t), \mathbf{u}_i(t)), \quad i \in \mathcal{N}.$$

The general planning problem in cooperation may be stated as

$$\min_{\{\mathbf{x}_i(\cdot), \mathbf{u}_i(\cdot)\}_{i=1}^N} \mathcal{J},$$

where the cost is decomposed into

$$\mathcal{J} = \mathcal{J}_{\text{mission}} + \mathcal{J}_{\text{performance}} + \mathcal{J}_{\text{risk-comm}},$$

where

$$\mathcal{J}_{\text{mission}} = \sum_{i=1}^N \mathcal{J}_{i,\text{mission}},$$

$$\mathcal{J}_{\text{performance}} = \lambda_t \mathcal{J}_{\text{time}} + \lambda_e \mathcal{J}_{\text{energy}} + \lambda_s \mathcal{J}_{\text{smooth}},$$

and

$$\mathcal{J}_{\text{risk-comm}} = \lambda_r \mathcal{J}_{\text{risk}} + \lambda_c \mathcal{J}_{\text{comm}}.$$

where the non-negative weighting coefficients λ_t , λ_e , λ_s , λ_r , and λ_c regulate the relative importance of mission duration, energy consumption, trajectory smoothness, collision risk, and communication performance. The optimization is subject to the vehicle dynamics, prescribed initial and terminal conditions,

$$\mathbf{x}_i(0) = \mathbf{x}_{i,0}, \quad \mathbf{x}_i(T_i) \in \mathcal{X}_{i,\text{goal}},$$

and actuator and flight-envelope constraints,

$$\mathbf{x}_i(t) \in \mathcal{X}_i, \quad \mathbf{u}_i(t) \in \mathcal{U}_i, \quad t \in [0, T_i].$$

For a possibly time-varying obstacle $\mathcal{O}_k(t)$, collision avoidance requires

$$d(\mathbf{p}_i(t), \mathcal{O}_k(t)) \geq r_i + s_{ik}^{\text{obs}}(t), \quad \forall i, k, t,$$

where $\mathbf{p}_i(t)$ is the position component of $\mathbf{x}_i(t)$, r_i is the effective physical radius of UAV i , and $s_{ik}^{\text{obs}}(t)$ is the additional obstacle-clearance margin. Pair-wise separation is imposed through

$$d_{ij}(t) \equiv \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|_2 \geq r_i + r_j + s_{ij}(t), \quad \forall i \neq j, t.$$

The safety margin $s_{ij}(t)$ should not be treated as an arbitrary tuning constant. It must account for the principal sources of uncertainty affecting vehicle separation and may be represented conceptually as

$$s_{ij}(t) = s_{ij}^{\text{loc}}(t) + s_{ij}^{\text{track}}(t) + s_{ij}^{\text{pred}}(t) + s_{ij}^{\text{comm}}(t) + s_{ij}^{\text{aero}}(t),$$

where the individual terms represent localization uncertainty, trajectory-tracking error, motion-prediction error, communication delay or packet loss, and aerodynamic or wake-interaction effects, respectively. The effective radius r_i should already incorporate the physical dimensions of the airframe and any necessary geometric buffer.

Calibrate localization bounds from held-out ground-truth residuals at the relevant update rate and in representative occlusion/GNSS conditions; estimate tracking bounds from flight logs over wind, payload, battery, controller latency, and actuator saturation; and use prediction residual quantiles or conservative reachable sets at each forecast horizon. Bound communication displacement using measured end-to-end age-of-information and packet-loss bursts multiplied by bounded relative speed/acceleration, including clock error. Estimate aerodynamic exclusion volumes from measured vertical/horizontal disturbance and attitude/altitude recovery tests at the tested rotor spacing, rather than treating downwash as a uniform sphere. Use simultaneous confidence bounds or a clearly budgeted risk allocation across errors and agent pairs; naive addition of sample standard deviations is not a deterministic bound. Report calibration sample size, quantile/confidence method, update interval, extrapolation horizon, and observed bound exceedances. If a bound expires or cannot be established, do not claim assured separation: execute a pre-verified braking, hover, loiter, altitude-separation, or landing contingency only if its own reachable tube is known to remain conflict-free; otherwise stop autonomous mission progression and defer to controlled separation/abort procedures.

If the uncertainty is modeled stochastically, deterministic separation can be replaced with the chance constraint

$$\Pr[d_{ij}(t) \geq d_{ij}^{\min}] \geq 1 - \varepsilon_{ij},$$

where $d_{ij}^{\min}$ is the required minimum separation and ε_{ij} is the maximum acceptable probability of constraint violation. If the underlying probability distribution is itself uncertain, a distributionally robust formulation may be adopted:

$$\inf_{\mathbb{P} \in \mathcal{P}} \mathbb{P}[d_{ij}(t) \geq d_{ij}^{\min}] \geq 1 - \varepsilon_{ij},$$

where $\mathcal{P}$ denotes an ambiguity set containing plausible probability distributions. Risk may alternatively be quantified using conditional value-at-risk, probabilistic occupancy maps, reachability tubes, or robust controlled-invariant sets. These formulations demonstrate why nominal shortest-path solutions cannot be transferred directly to uncertain and dynamically evolving airspace: a geometrically valid route may still become unsafe when sensing errors, prediction uncertainty, communication imperfections, and closed-loop tracking dynamics are considered.

3.2. Coordination Architecture

A centralized planner has global state information and is able to optimize for the entire problem, but the decision dimension and pairwise constraints become large

and there is a single point of delay or failure of the coordinator. Decentralized planners distribute computation and offer automatic scalability and fault isolation properties, but need coordination protocols for intent exchange, priority, reciprocity or conflict resolution. Distributed optimization techniques let agents solve local sub-problems and coordinate by means of consensus, alternating direction, sequential negotiation or safe corridor construction [26]. Hybrid approaches explicitly employ different techniques for mission, route, trajectory and safety levels [18] [19] [27]-[32]. **Figure 2** translates these complementary roles into a hierarchical autonomy loop. Mission-level objectives guide global coordination, local prediction and trajectory generation, and safety-critical control before commands reach the vehicles. Feedback links the execution layer to subsequent planning decisions. The diagram represents a proposed integration architecture; its practical value depends on compatible interfaces, timely feedback, and feasible safety interventions, rather than implying an independently established end-to-end guarantee.

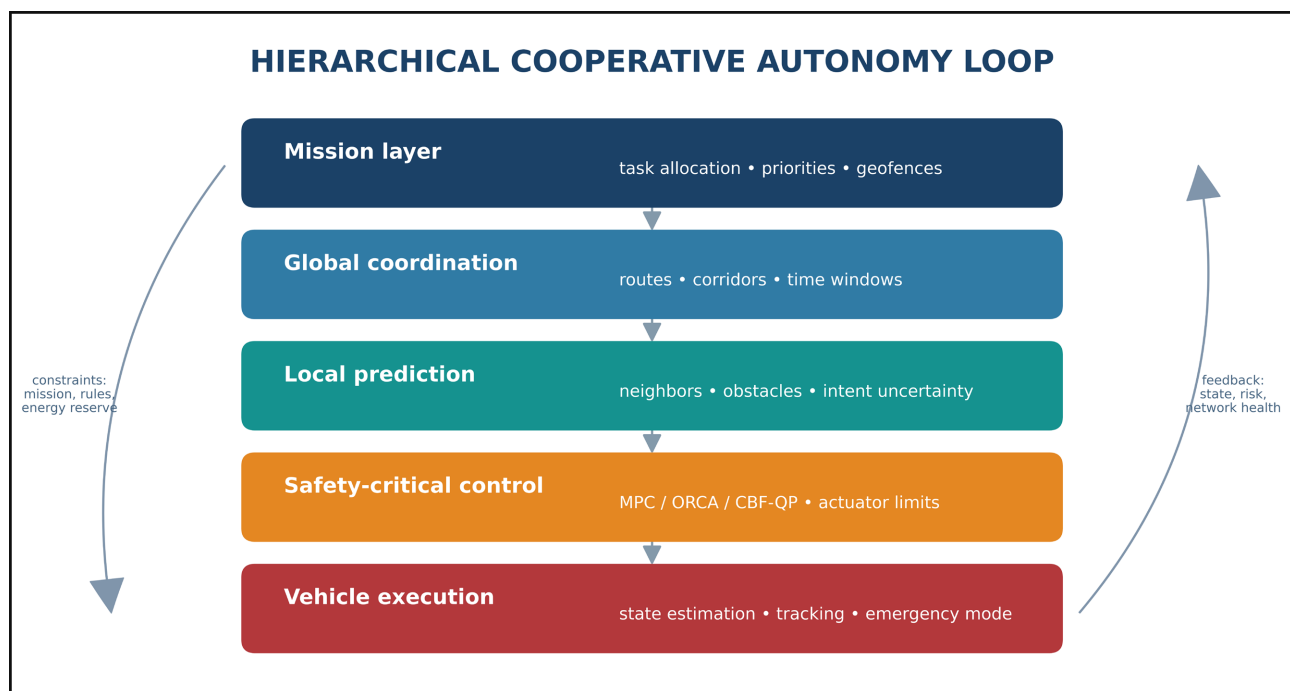


Figure 2. Recommended hierarchical architecture coupling mission planning, global coordination, local prediction, safety-critical control, and vehicle execution. Original schematic created for this review.

3.3. What Should Be Measured?

Minimum acceptable metrics include mission success, collision and near-miss rate, minimum separation, cost of paths or trajectories, makespan, energy consumption proxy, latency of planning cycle and tail latency, memory and CPU/GPU load, communication bytes and age of information, constraint violations, deadlock rate and recovery mechanism. Scalability must be analyzed as a function of the number of UAVs, number of neighbors, obstacle density, map size, and horizon. Experiments for robustness must include perturbation of localization, dy-

namics, wind, sensing range, packet delay, packet losses and unmodeled agent behaviors. Hardware tests should reveal onboard CPU, estimator, communication graph, control rate, safety radius and criteria for intervention.

Predeclared event definitions: A collision is physical contact or a measured signed surface separation ≤ 0 , including vehicle-obstacle contact. A near miss is $0 < \text{signed surface separation} < s_{\text{req}}(t)$, where s_{req} is the prespecified pair- or obstacle-specific operational buffer; count contiguous excursions as one event and also report minimum clearance, duration, and time-to-collision. Mission success requires every assigned mandatory waypoint/task to be completed within its prespecified deadline, with no collision, geofence breach, energy-reserve violation, or operator rescue. Deadlock is simultaneous lack of mission progress for at least $\tau_D = \max(10, s 2T_H)$, where T_H is the planning-horizon duration, while goals remain feasible, using displacement toward the assigned goal < 0.1 m for each blocked vehicle in the window. Livelock is repeated replanning or route/velocity reversal at least twice in the same τ_D window with net goal progress < 0.1 m and no mission completion. Report thresholds in metres and seconds, frame rate, uncertainty of separation measurement, timeouts, trial denominator, censored runs, and whether simultaneous near-miss pairs are counted separately. The 0.1 m and 10 s defaults should be adapted before a test to vehicle size/speed and then held fixed across algorithms.

Table 1 compares the 14 algorithm families using a common set of descriptors covering representation, vehicle dynamics, environmental assumptions, information requirements, safety claims, and computational or communication limits. It distinguishes formal guarantees from conditional properties and empirical results, making clear why a method that performs well under one model may not retain the same properties under different sensing, dynamics, or coordination conditions. The comparison is intended to guide method selection rather than provide a numerical ranking.

Table 1 appraisal key: F = formal under stated assumptions, C = conditional on feasibility and model/bounds, E = empirical; NR = not reported. Each row summarizes the source method, not a guarantee for an untested UAV. Evidence type for individual implementations is reported in **Table 2**. An R/P/NR checklist for the eight criteria is defined in Section 2; these rows are qualitative and unweighted.

Table 1. Algorithm-family comparison for multi-UAV cooperative planning and collision avoidance.

Family	Methods/representation	Vehicle dynamics/environment	Architecture/information	Safety strategy/claim level	Compute and communication scaling; limits	Refs.
Graph search	A*, D*, D* Lite, JPS	Static graph or discrete steps; multirotor dynamics NR; Grid or time graph	Central or prioritized	Obstacle occupancy; F search optimality only on specified graph	Search nodes/map increase; shared map needs updates	[8] [9] [33]
MAPF	CBS, ECBS, SIPP, MO-CBS	Discrete time and geometry; continuous UAV dynamics absent unless extended; Time-expanded vertex/edge graph	Mostly centralized	Vertex/edge conflict rejection; F graph-optimal CBS; C flyability after refinement	Conflict tree can grow exponentially; central plans need shared state	[9]-[13] [34]

Continued

Sampling	PRM, RRT, RRT*, kinodynamic RRT	Geometric or kinodynamic model as implemented; often single UAV; Configuration/state samples	Central or per-agent; source validation varies	Sampled obstacle screening; F probabilistic completeness only with assumptions; E UAV success	Joint dimension grows with N; prioritized plans need updates	[35]-[37]
Trajectory optimization	SCP, SQP, DC programming	Limits explicitly encoded only if enforced; moving obstacles forecast; Spline/polynomial/controls	Central or distributed	Continuous deconfliction with feasible solver; C dynamics and safety	Pairwise constraints scale with neighbors $\times$ horizon; exchange trajectories	[18] [28] [38]-[40]
Mixed-integer optimization	MILP, MIQP	Simplified aircraft/robot model, discretized time; Mixed discrete and continuous plan	Central/distributed receding horizon	Hard modeled constraints; F bounded-model optimum only if solver proves it; C UAV safety	Binary conflict decisions scale steeply; central coordination burden	[27] [41]
Metaheuristics	PSO, GA, DE, ACO, GWO	Typically geometry only; post-smoothing needed; Waypoint/control-point population	Usually centralized; source validation varies	Penalty-based rejection/repair; E feasibility and performance	Population $\times$ N $\times$ obstacles $\times$ iterations; usually central	[42]-[48]
Potential fields	APF and improved/hybrid APF	Point-mass/kinematic local step; Reactive artificial forces	Naturally decentralized	Local repulsion; E no general clearance guarantee	Local degree $\times$ obstacle count; sensing/local state only	[49] [50]
Velocity obstacles	VO, RVO, ORCA, V-RVO	Usually constant velocity discs; UAV acceleration/downwash require extension; Velocity-cone/reciprocal constraints	Decentralized reciprocal; source validation varies	Reciprocal half-plane filtering; F only ideal model, C when trackable	Local neighbors and pairwise exchange; delays/reciprocity limit	[51]-[54]
Dynamic window	DWA and adaptive DWA	Velocity/acceleration bounds at local horizon; Reachable velocity window	Local/decentralized	Short-horizon obstacle score; E tested clearance	Window samples $\times$ obstacles $\times$ neighbors; local sensing	[55] [56]
Model predictive control	DMPC/NMPC	Flight envelope/dynamics only to fidelity encoded; Finite-horizon states and inputs	Central or distributed	Hard MPC constraints; C feasible solver/model and terminal fallback	Per-agent neighbor constraints $\times$ horizon; exchange predictions	[29]-[31] [57]
Control barrier functions	CBF-QP, HOCBF, collision-cone CBF	Relative dynamics/actuator limits as formulated; Barrier on nominal control	Distributed safety filter	F safe-set invariance if model valid and QP feasible; otherwise C	Local pairwise barriers; dense neighbors cause infeasibility	[58]-[62]
Deep RL/MARL	DQN, PPO, SAC, MAPPO	Simulator/training model; transfer uncertain; Learned policy/local observations	Decentralized execution; source validation varies	Reward/shield if implemented; E unless filter separately proved	Inference vs training separated; local comm as configured	[20]-[25] [63]
Graph learning	GNN/GAT + MARL	Learnt agent model; dynamics depend on training; Interaction graph/messages	Decentralized/local message passing; source validation varies	Learned coordination; E outside proven shield	Message edges, bandwidth, topology shift; primarily simulation	[23] [64]
Hybrid safe learning	RL proposal + MPC/CBF/shield	Depends on specified filter dynamics; Policy + constrained projection	Hierarchical	C safety inherited only if verified feasible filter holds	Policy inference + QP/MPC; neighbor state/delay overhead	[21] [58] [65]

4. Classical and Optimization-Based Planning

4.1. Graph Search and Multi-Agent Path Finding

D* and A* are appealing when it is possible to construct an occupancy map and apply efficient heuristics for the search. D* Lite and other incremental algorithms reuse search work performed before the map change, while jump-point algorithms eliminate symmetric expansions on uniform maps. In the multi-agent case, naïve sequential planning followed by conflict correction is incomplete and introduces loops of replanning. MAPF introduces time into the model and forbids both vertex collisions and edge swap collisions. CBS decomposes the problem into the high-level constraint tree and the low-level single-agent planning problem and produces an optimal solution, ECBS and focal-search variations sacrifice optimality for speed [8]-[11].

The mismatch with UAV physics is significant. Classical MAPF models agents as point masses, with unitary moves, synchronous decisions, and discrete collisions. Continuous-time, kinematic, heterogeneous geometry, and kinodynamic extensions alleviate the discrepancy but raise the complexity of search [12] [13] [34]. Hence, in a practical pipeline for UAVs, one could use CBS to compute the order of routes and time windows and refine trajectories with continuous collision checking using a polynomial approach. Thus, it preserves the combinatorial nature of MAPF while avoiding pretending that paths in grids are flyable.

4.2. Sampling-Based Planning

PrPMSM provides efficient solutions for repeated problems in a mostly static environment, while RRT-like algorithms quickly explore the high-dimensional space of possibilities. RRT* gives asymptotic optimality, while informed, bi-directional, kinodynamic, and safe-corridor variants improve convergence and feasibility, respectively. The interaction of multiple UAVs could be considered either in the joint configuration space, by means of prioritized planning, or by sampling individual routes and scheduling time separately. Joint-space planning is conceptually elegant but suffers from the curse of dimensionality. On the other hand, prioritized planning scales better, although it might be incomplete and affected by the order of priorities [35]-[37]. Finally, sampling is particularly relevant in serving as a global frontend for discovering homotopy classes of routes amidst difficult obstacles. The sampled route should normally be smoothed and time parameterized with B-splines, Bézier curves, minimum snap polynomials, or some sort of constrained optimization. In dynamic environments, a local replanner or safety controller would still be needed because building a global tree on sensor rate is seldom possible.

4.3. Mathematical Programming and Trajectory Optimization

Trajectory optimization represents each route in terms of states, controls, waypoints, splines, or polynomial coefficients and tries to minimize time, energy, smoothness, exposure, or formation error [66] [67]. The pairwise separation constraint makes

the feasible set non-convex. SCP, sequential quadratic programming, alternating minimization, difference of convex programming, and separating-hyperplanes reformulation convert the non-convex problem into a series of solvable subproblems [18] [28] [38]-[40]. Their performance depends on initialization, trust region, penalty construction, and collision detection methods. MADER algorithm is an example of a decentralized asynchronous scheme. Each agent is trying to optimize a piecewise polynomial trajectory using convex outer approximations of trajectory segments; separating planes between polyhedra are made optimization variables, making possible time continuous collision detection with static obstacles, moving obstacles, and other agents [18]. RMADER uncovers the communication assumption that has been implicit so far: a trajectory optimized on the basis of outdated information about neighbors might be unsafe until they get the information and update. Its delay checking and two stages of publication make sure that the last committed safe trajectory will survive a bounded-delay verification period [19]. Dynamic feasibility follows only if a dynamically valid trajectory is returned and validated at the intended controller rate; nonconvex initialization or infeasibility can defeat this condition. RMADER's guarantee depends on a known delay bound and successful delay check. In its published hardware evaluation, the 50 ms bound was exceeded for approximately 25% of mesh-network delays, so collision-free observed flights in those cases are E, not guaranteed [19].

4.4. Bio-Inspired and Evolutionary Optimization

Particle swarm optimization, genetic algorithms, differential evolution, ant colony optimization, artificial bee colony, grey wolf and wolf pack algorithms are popular due to the ease of encoding the problem with multiple objectives and arbitrary cost for threat, while requiring no gradient computation. The candidate solution generally consists of waypoint coordinates or spline control points. Hybrid versions include chaotic initialization, opposition learning, adaptive parameters, local search, or deterministic smoothing [42]-[48] [68]. Their adaptability is real, although many comparisons are poorly conducted. New metaheuristics are often compared with untuned canonical alternatives on a limited number of synthetic scenarios, reporting convergence iterations without wall clock latency or flight dynamics feasibility. Stochastic planners should provide several repeated runs, confidence intervals, sensitivity analysis of population and penalty parameters, constraint repair overhead, and inference time. For online replanning, a deterministic solver with a warm start might beat a metaheuristic when the latter just barely finds a slightly shorter offline route.

Figure 3 illustrates how piecewise Bézier trajectories negotiate static threat regions in two planning scenarios [47]. The curved routes demonstrate the geometric relationship between trajectory smoothness and obstacle clearance. These examples help explain the role of continuous trajectory representations, but the displayed paths alone do not establish performance under moving obstacles, inter-UAV conflicts, or uncertain vehicle dynamics.

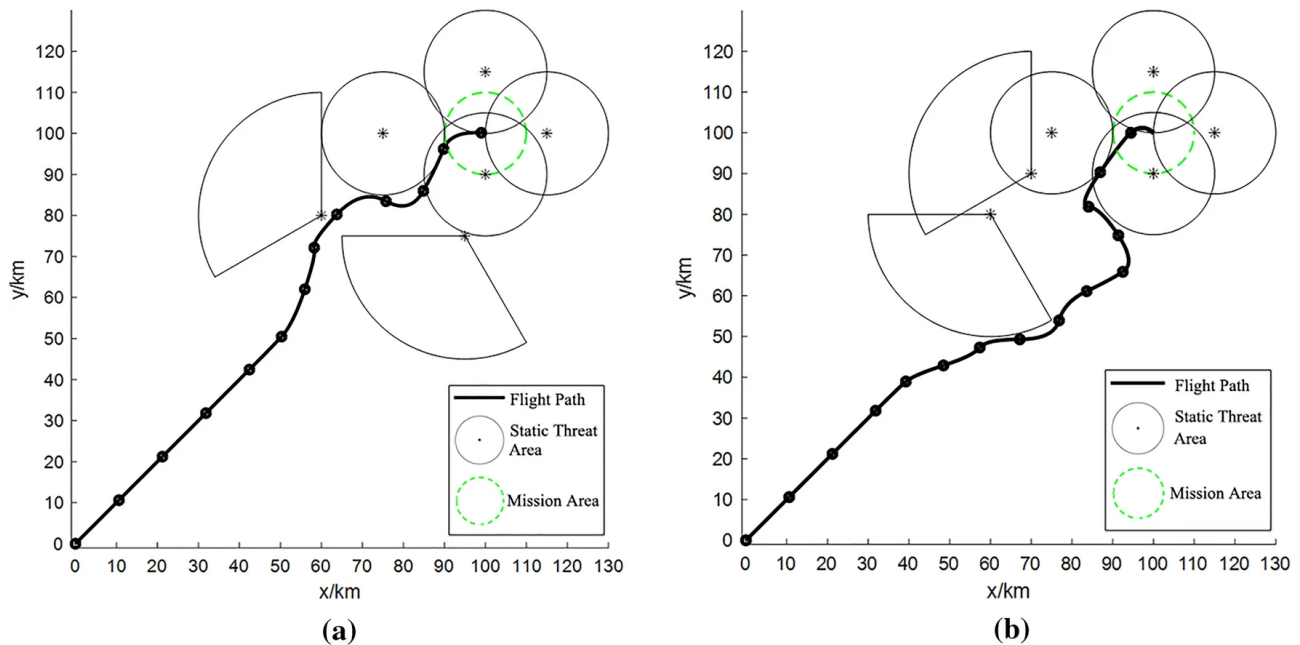


Figure 3. Example of optimization-based UAV path planning around static full-coverage threat regions using piecewise Bézier trajectories. Reproduced from Zhang, Liu and Feng (2023) [47], Scientific Reports 13, 16453, under the Creative Commons Attribution 4.0 licence; panels show two task scenarios from the source article's **Figure 2**.

5. Reactive, Predictive, and Safety-Critical Collision Avoidance

5.1. Artificial Potential Fields and Dynamic-Window Methods

Potential fields use attraction to goals and repulsion from obstacles and neighbors. They are often inexpensive locally but can oscillate, become trapped, or deadlock in clutter; modifications such as tangential forces and virtual targets do not establish general completeness [49] [50]. DWA scores velocity commands reachable over a short horizon; the modeled velocity and acceleration limits condition whether a command is flyable [55]. Chang *et al.* [56] report MATLAB 2019b simulations with three UAVs in each of two tasks (aggregation and target search), five randomly generated groups per task. Across the 10 groups, **Table 1** gives success of 2/10 for DWA, 1/10 for APF, and 10/10 for the improved DWA-ORCA method, with average distance 19.91, 16.75, and 14.35 m and mean planning times 56.46, 36.95, and 48.75 ms per instance, respectively. The publication does not provide per-trial timing variance or processor specification for that comparison. These descriptive, small-sample simulation differences do not establish a general improvement or real-time flight safety.

5.2. Velocity Obstacles and Reciprocal Avoidance

Velocity obstacles consist of velocities resulting in a collision within a preselected horizon if another agent holds its estimated velocity. Reciprocal Velocity Obstacles distribute the avoidance responsibility, while ORCA transforms pairwise constraints into half-planes whose intersection determines allowed velocity. The al-

gorithm is decentralized and efficient, being a de facto local multi-agent baseline [51]-[54]. Three assumptions warrant highlighting: estimates of states and velocities must be sufficiently accurate; agents must reciprocate in a reasonable manner; and the selected command must be dynamically trackable. Non-cooperative traffic, delayed intent information, narrow passages, and high-density infeasibility may break those assumptions. V-RVO and Voronoi-based variants reduce oscillations and incorporate local free space. Extensions for priority awareness and non-holonomy improve applicability. For UAVs, vertical clearance, anisotropic safety volumes, downwash, and acceleration constraints must be explicitly represented rather than projecting a planar disc model to 3-D. ORCA's collision-avoidance result applies to reciprocal agents satisfying its constant-velocity, sensing, horizon, and feasible-velocity assumptions; UAV flight with delay and acceleration limits requires separate validation.

5.3. Model Predictive Control

MPC repetitively predicts the evolution of the system, solves a finite-horizon optimization problem with constraints, applies the resulting first control and shifts the horizon. Distributed MPC allows each UAV to solve an optimization problem based on the predicted trajectories of neighboring agents. Distributed MPC unifies tracking, separation, obstacle avoidance, formation preservation, input limitations, and energy optimization into a single objective and constraint set [29]-[31] [57]. Warm-starting and sparse nonlinear programming enable real-time processing for modest-sized teams. However, the computational complexity is not the only limitation. Recursive feasibility breaks when several hard collision constraints become incompatible with each other, late prediction comes in, or horizon cutting hides a looming dead-end. Slack variables ensure feasibility but degrade hard safety guarantees if there is no separate invariant fallback. Terminal sets, safe corridors, priority negotiation, tube MPC, and distributed consensus alleviate those problems. Mean solve time without worst-case analysis and deadline-miss statistics is inadequate as evidence for safety-critical claims. Constraint satisfaction refers to feasible, completed optimization under the modeled disturbance and communication bounds; real-time completion and recursive feasibility must be demonstrated or separately proved.

5.4. Control Barrier Functions and Reachability

Control barrier functions $h(x)$ define the safe set $C = \{x: h(x) \geq 0\}$. Ensuring $\dot{h}(x,u) + \alpha(h(x)) \geq 0$ by means of quadratic programming yields the minimal change of a nominal command sufficient for maintaining forward invariance provided that the model is correct and the QP stays feasible. Higher-order CBFs accommodate relative degree constraints, while collision-cone CBFs capture relative motions. Multi-UAV safety can be made decentralized by imposing pairwise barrier constraints between local neighbors [58]-[62]. Large swarms expose the core limitation: several barrier constraints and actuator constraints may become mutually

incompatible. Priority assignment, adaptive relaxation, responsibility allocation, backup controllers, and feasibility-aware CBFs are actively studied topics. Reachability analysis and Hamilton-Jacobi approaches offer stronger guarantees but suffer from a dimensionality curse. However, a practical design can employ reachability analysis offline to build invariant safety envelopes and CBF-QP online to enforce them. Forward invariance holds only on the certified safe set when the barrier inequality is continuously enforceable, state estimation and dynamics satisfy the assumptions, and the QP remains feasible; a relaxed or infeasible QP removes the claim. **Figure 4** extends the geometric illustration to moving threat regions by showing successive receding-horizon path segments [47]. Updating the planned route as the threat configuration changes highlights the need to couple prediction with repeated replanning. The figure illustrates this planning mechanism; it does not by itself demonstrate a multi-UAV safety guarantee under sensing errors or communication delay.

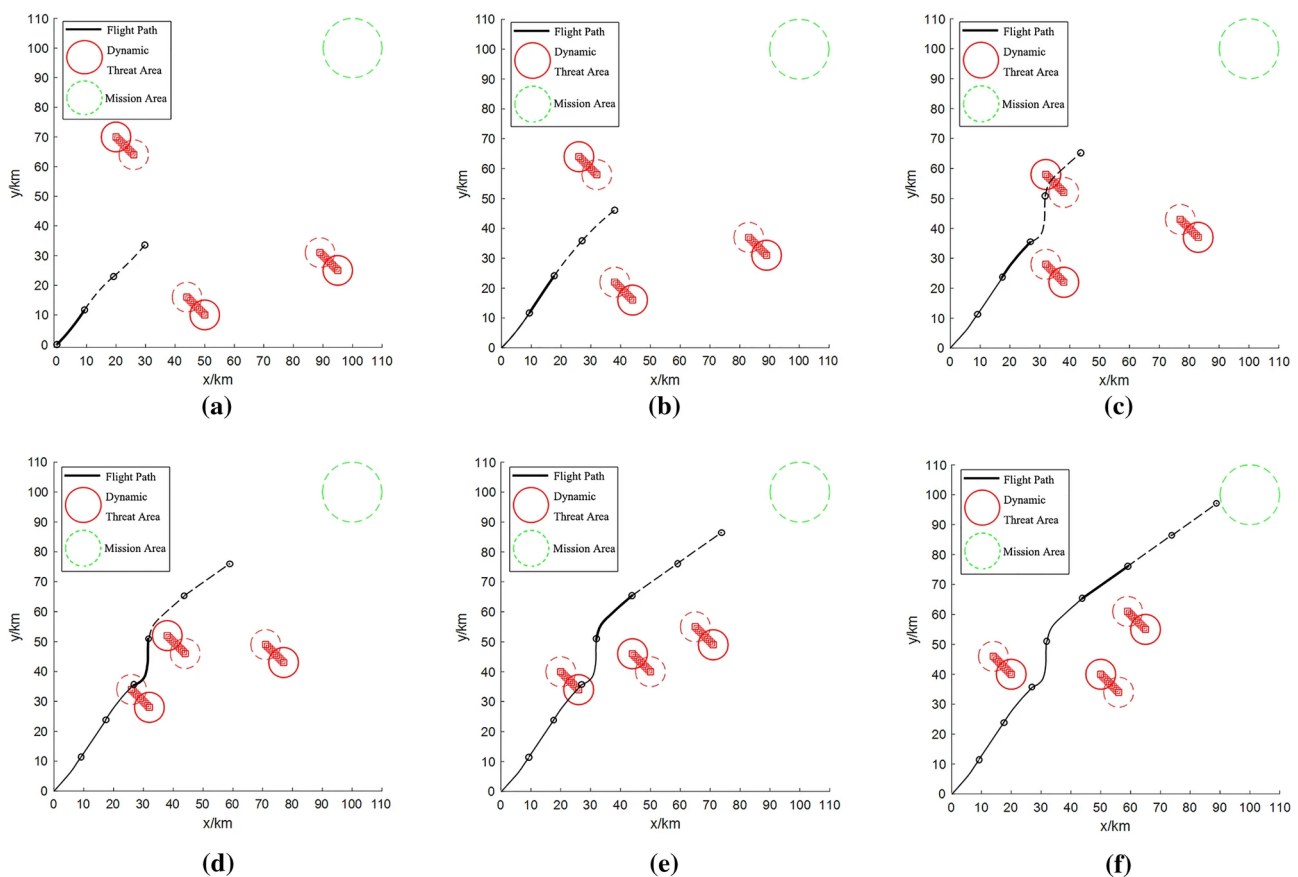


Figure 4. Receding-horizon path segments generated against moving threat regions, illustrating the need to couple prediction and repeated replanning. Reproduced from Zhang, Liu and Feng (2023) [47], Scientific Reports 13, 16453, under the Creative Commons Attribution 4.0 licence.

Table 2 complements the algorithm-family comparison with 14 representative systems and their deployment evidence. It relates each mechanism to its dynamics, communication assumptions, claim level, validation setting, and remaining

limitations. Separating simultaneous multi-UAV flight from simulation, transferable multi-robot evidence, and single-UAV demonstrations prevents results obtained under different experimental conditions from being treated as equivalent evidence of field readiness.

Table 2 appraisal key: UF = simultaneous multi-UAV flight; US = multi-UAV simulation or HIL; MR = transferable multi-robot; SU = single-UAV. F/C/E have the meanings in Section 2. A grouped row may mix evidence types; individual numerical outcomes inherit the exact source setting. NR means not reported or not verifiable from the cited source. This representative table does not constitute an independently screened cohort.

Table 2. Representative multi-UAV systems and their deployment evidence.

Method/study	Mechanism	Environment and flight dynamics	Information/communication	Claim level and compute/comm scaling	Evidence category/validation	Safety mechanism and limits	Refs.
EGO-Swarm	Decentralized asynchronous polynomial optimization	Unknown clutter; quadrotor polynomial trajectories	Broadcast trajectories	E onboard replanning; scaling with swarm density/bandwidth NR	UF; additional US	Trajectory exchange and local collision checks; delay assumptions	[16]
Spatial-temporal micro-swarm	Joint shape and time allocation	Outdoor natural clutter; palm-sized multirotor	Local trajectory exchange	E millisecond optimization on tested hardware; message-load scaling NR	UF; US	Spatial/temporal optimization, local estimates; density/occlusion limitations	[17]
MADER	Decentralized polynomial optimization + separating planes	Dynamic/obstacle scenes; polynomial quadrotor trajectories	Asynchronous trajectory exchange	C continuous modeled separation with feasible solution; delay load grows with peers	UF/US as source reports	Separating planes/committed trajectory; stale information limits safety	[18]
RMADER	MADER + delay check and two-stage publication	Dynamic obstacles; polynomial trajectories; bounded delay	Mesh or centralized network	F deconfliction only if delay-check bound holds; neighbor checks/messages scale	UF + US	Two-stage publication/delay check; 50 ms bound exceeded in ~25% mesh delays [19]	[19]
Linear safe corridor	Distributed trajectory optimization	Obstacle-rich; quadrotor polynomial and corridor model	Neighbor trajectories and priorities	C feasible corridor only under source assumptions; coordination overhead NR	UF/US where documented	Linear safe corridors/priority; conservative or infeasible corridors	[32]
AsyncBVC	Asynchronous buffered Voronoi constraints	Free-space partition; dynamic vehicle model varies by cited paper	Neighbor plans	C buffered partition under specific update and bound assumptions	MR/US; no direct UF attributed	Voronoi buffer; sources [19] [53] do not establish a single unified system	[19] [53]
DWA-ORCA hybrid	Bidirectional/global guidance + local DWA + triggered ORCA	Three simulated quadrotors; 0.7 m/s maximum	Local state exchange/sensing	E: 10-group simulation; latency 48.75 ms mean; processor and scaling NR	US; MATLAB 2019b [56]	Local DWA/triggered ORCA; no flight or system-wide guarantee	[56]

Continued

Downwash-aware planner	Discrete graph plan + continuous refinement	Dense multi-quadrotor formations; ellipsoidal downwash model	Central planning	E 200-agent simulation, 32-quadrotor test; offline/minute-scale planning	UF + US	Downwash-aware geometry and continuous refinement; limited online response	[36]
CBF guidance	Nominal guidance filtered by pairwise CBF-QP	Relative-state model varying by study; actuation bounds vary	Relative states	F barrier invariance only if feasible and accurate; QP cost increases with neighbors	MR/US/UF only where each study reports	CBF-QP pairwise constraints; infeasibility and estimator error	[58]-[62]
End-to-end MARL swarm	Shared policy, decentralized execution	Policy trained on simulated agent interactions; flight model varies	Local observations or learned messages	E inference speed and swarm behavior, no distribution-wide certification	MR/US; UF only for UAV study [23] when verified	Training reward and any separate shield; OOD risk	[20]-[23]
RSNN swarm avoidance	Reward-modulated spiking network	Shared airspace; spiking network drone model	Local observation; no central control	E behavior on tested cases; runtime/message scaling NR	UF + US, per [24]	Reactive learned avoidance; generalization and safety limits	[24]
PPO multi-UAV avoidance	Centralized training/decentralized execution	Multi-UAV simulator; continuous action model	Training-time joint information	E simulated avoidance; training compute distinct from inference	US [25]	Policy reward only; no formal bound	[25]
Filtered RL	RL proposal passed through model-based safety filter	Multi-agent model with action filter	State/trajectory estimates	C if filter proof and state bounds valid; filter cost increases with constraints	MR/US; UF NR for [65]	Projection/backup filter; infeasible filter removes guarantee	[65]
GNN-PPO/attention MARL	Dynamic graph aggregation + policy optimization	Varying neighbor topology; model depends on source	Learned local messages	E graph generalization in evaluated sizes; bandwidth/degree needed	US; do not infer flight from [23] [64] grouping	Learned attention/messages; topology and latency shift	[23] [64]

6. Learning-Based and Hybrid Autonomy

6.1. Multi-Agent Reinforcement Learning

MARL models a cooperative navigation task as a decentralized partially observable Markov decision process. Centralized learning uses joint state and reward representations, while decentralized execution translates local observations into actions. Value-based methods fit well for discrete actions, while actor-critic methods like PPO, MAPPO, SAC, and TD3 fit better for continuous commands. Parameter sharing enables homogeneous swarms, while curriculum learning grows the team size, obstacle density or dynamics [20]-[25]. Learning resolves implicitly interaction problems that can hardly be solved manually, and ensures constant inference time. However, a poorly designed reward function may make a trade-off between rare collisions and average progress invisible. Training distributions typically exclude sensor failures, extreme density, non-reciprocal agents, actuator faults, and wind. Average success rate alone does not bound worst-case risk. Evaluation should thus cover out-of-distribution maps, randomized dynamics and la-

tency, adversarial agents, uncertainty calibration, and confidence intervals over random seeds.

6.2. Graph Neural Networks, Attention, and Communication Learning

Swarm is naturally a dynamic graph: UAVs are nodes and sensing or communication relations are edges. GNN aggregates local neighbor features using the same weights, providing permutation invariance and an option for scaling team sizes. Attention mechanisms learn which neighbors or obstacles affect the action the most, while recurrent networks encode temporal context. Learned communication reduces intent exchange overhead but becomes brittle to topology changes, packet loss, or malicious communications [23] [64].

The important comparison is not between GNN and non-GNN, but between learned messages and explicit exchange of relative-state and predicted-trajectory information at the same bandwidth and latency. Ablations need to include variations in neighbor-radius, graph-degree, drop-rate, quantization, and unseen team-size. Safety-critical systems require detection of graph-distribution shift and falling back to a provably safe local controller.

6.3. Safe Learning and Hybrid Planners

Hybridization is the most credible path to practical use. A global optimizer or learned policy suggests a route; trajectory optimization guarantees smoothness and satisfies actuator limitations; MPC tracking with replanning handles unexpected perturbations; ORCA, CBF-QP, or reachability provide the safety-critical fallback. Filtered reinforcement learning can project an unsafe action into the feasibility set in training and execution [65]. Residual learning learns a corrective term to a stable model-based controller, shrinking the space that must be learned. Smoothness and safety constraints in this architecture depend on feasible optimization, accurate models, valid uncertainty bounds, and a working fallback. The architecture itself provides no unconditional system-wide guarantee.

Nevertheless, safety filter is not a ceremonial end point. Feasibility region of the safety intervention, computation time deadline, switching transient effects, dependency on state estimation, and multiple constraint interactions must be analyzed. In case the learning policy frequently requests unsafe actions, the safety filter can induce deadlock or severe performance loss without collision. Rate and severity of safety interventions should be a reported metric, accordingly.

Figure 5 compares the success proportions and mean planning times reconstructed from the ten simulation groups reported by Chang *et al.* [56]. The improved DWA-ORCA method achieved success in all ten groups, while APF and DWA succeeded in one and two groups, respectively. APF had the lowest reported mean planning time, showing why computational cost and task success should be assessed together. The broad Wilson intervals reflect the small group count, and the absence of timing distributions and processor details limits conclusions about

statistical superiority or real-time deployment.

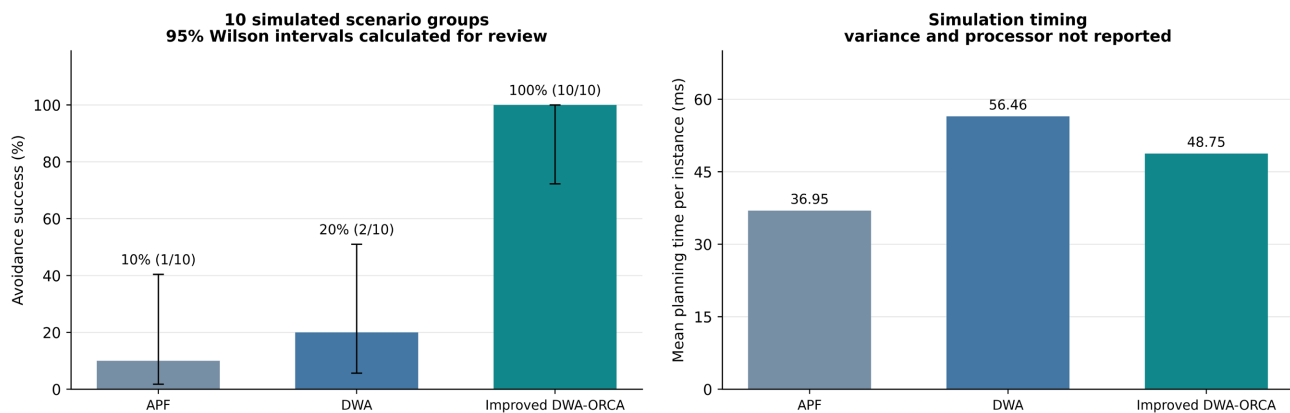


Figure 5. Descriptive reconstruction of Chang *et al.* (2025) [56], **Table 1:** avoidance successes APF 1/10 (10%), DWA 2/10 (20%), improved DWA-ORCA 10/10 (100%), and mean planning time per instance APF 36.95 ms, DWA 56.46 ms, improved method 48.75 ms. The source reports 10 MATLAB 2019b multi-UAV simulation groups: five aggregation and five target-search tasks, each with three simulated UAVs, randomized starts/targets and obstacles, a 0.5 m obstacle radius, 1 m avoidance radius, 0.7 m/s maximum simulated speed, and 0.1 s default step [56]. Approximate post hoc 95% Wilson intervals for 1/10, 2/10, and 10/10 are 1.8% - 40.4%, 5.7% - 51.0%, and 72.2% - 100%, respectively; these intervals were calculated for this review, not reported by the source. The source does not disclose a hardware processor specification or trial-level timing spread, so timing uncertainty and deadline-miss probability cannot be estimated. The panels are descriptive within this benchmark, not a statistical superiority or flight validation claim.

7. Cross-Cutting Constraints That Determine Real-World Validity

7.1. Communication Delay, Loss, and Age of Information

Decentralized planner usually treats the latest obtained trajectory as up-to-date. Instead of considering only packet delay, we should consider age of information—a combination of sensing, estimation, computing, queuing, transmitting, and clock differences. Bounding delay can be achieved by robust tubes, bounded-prediction sets, asynchronous commitment protocol, event triggered updates, or conservative reachable sets. RMADER provides a good example: formally, the requirement is to check the time passed since the last observation at least once in a period equal to maximum possible communication delay [19]. In case the time passed is larger than the chosen bound, formal guarantee does not hold even if no collision happened experimentally. Complete packet loss is a special case which needs a special approach. Vehicle can reserve braking or loitering trajectories, increase uncertainty tubes, return to right-of-way rule, separate altitude layer, perform mission abort, etc. Communication-motion code design should solve both how to fly and what to communicate, because the geometry of flight determines link quality, and network load—safety [69].

7.2. Perception, Localization, and Intent Uncertainty

Static map becomes outdated, dynamic object tracks are corrupted, and visual or lidar occlusions are correlated between nearby vehicles. Neighbor intent can be

modeled as a communicated trajectory, constant velocity forecast, interaction model, Gaussian process, multimodal learned predictor, or bounded reachable set. Deterministic single mode forecast is dangerous in intersections as two individually plausible evasive actions can be incompatible. Chance-constrained or scenario MPC and belief-space planning allow trading efficiency for risk, requiring calibrated uncertainty. Statistical coverage or risk control requires explicit calibration and distribution/shift assumptions; unknown or expired bounds trigger the contingency specified in Section 3.1.

7.3. Vehicle Dynamics, Energy, and Downwash

Point mass or double integrator models are helpful for algorithm development, but they might inflate the agility of the system. Fixed-wing UAVs have minimum speed, turn radius, and climb requirements; multirotors are differentially flat but have thrust, tilt, jerk, battery, and tracking bandwidth constraints. Downwash effects make safety anisotropic in vertical formations since the vertical separation volume is greater than the horizontal airframe radius. Downwash-aware trajectory planning has shown smooth trajectories for up to 200 simulated agents and 32 nanoquadrotor hardware experiments [36]. However, the offline-to-minutes planning approach contrasts with reactive unknown environment flights.

An energy model should incorporate more than just path length. Turns, acceleration, climbs, winds, payloads, communication, computations, and hovers cost energy. A swarm planner should reserve energy for contingencies, return to home base, or landing and should redistribute tasks if batteries are inconsistent. Multi-objective Pareto optimal analysis is preferable to combining all costs into unknown weights.

7.4. Heterogeneity and Human/Traffic Interaction

Heterogeneous swarms incorporate various radii, dynamics, sensors, endurance, and communication range. Symmetrical reciprocal algorithms and shared MARL policies may not work when avoiding capabilities differ. The responsibility must be allocated based on maneuverability, urgency, energy, and right-of-way. In a mixed airspace, cooperative UAVs need to coexist with unconnected drones, birds, helicopters, cranes, and emergency aircraft. Geofences, corridors, strategic de-confliction, remote identification, and contingency plans by standards and unmanned traffic management interfaces cannot be reduced to local obstacle avoidance anymore [70].

Table 3 converts the preceding comparisons into a proposed benchmark and reporting protocol spanning 14 evaluation dimensions. It pairs minimum metrics with reporting requirements and stress-test scenarios so that safety, mission completion, computational demand, communication, robustness, and deployment evidence can be assessed together. References attached to each dimension identify relevant methodological or application literature. The combined checklist, prescribed reporting requirements, and numerical event thresholds are recommen-

dations of this review; they should not be interpreted as requirements or measurements reported by every cited source.

Benchmark definitions (prospective): Collision = physical contact or signed surface gap ≤ 0 ; near miss = positive gap below the prespecified safety buffer; mission success = mandatory objectives completed before the deadline with no collision, geofence or energy violation, or manual rescue; deadlock = goal progress < 0.1 m for each blocked agent over $\tau_D = \max(10, s 2T_H)$, where T_H is the planning-horizon duration; livelock = ≥ 2 route or velocity reversals in the same τ_D window with < 0.1 m net goal progress. Adapt numerical defaults to the platform before testing, hold them fixed for all methods, and report near-miss pairs and zero-success trials.

Table 3. Recommended benchmark and reporting protocol for deployable multi-UAV planning.

Dimension	Minimum metrics	Reporting requirement	Stress-test scenarios	Ref.
Safety	Collision count; near-miss count; minimum pairwise/obstacle separation; time-to-collision; barrier violation; apply signed-distance collision/near-miss thresholds defined below Section 3.3	Report full distributions and worst-case values, not only means; define the UAV body geometry and safety margin; and specify the prespecified buffer s_{req} , trajectory-sampling and interpolation methods, and event denominators.	Static/dynamic obstacles; crossing, head-on, overtaking; dense bottleneck	[18] [19] [58]-[60]
Mission	Success rate; makespan; flow time; target coverage; formation error	Separate failure causes: collision, timeout, deadlock, energy, solver failure; use prespecified deadline and completion/energy/geofence conditions	Increasing targets, team size, deadlines, priority classes	[8] [9] [16] [17]
Path/trajectory quality	Length; duration; curvature; acceleration/jerk; control effort; Pareto hypervolume	Normalize by lower bound or single-agent optimum; state dynamics and smoothing method	Open space, clutter, narrow passage, altitude layers	[18] [28] [36] [67]
Computation	Mean, p95/p99 and maximum planning latency; deadline misses; CPU/GPU and memory	Disclose hardware, threads, solver, warm start, horizon, control rate; include processor model, OS, MATLAB/runtime version, N and p95/p99 or NR	Sweep N, neighbor degree, obstacle density and sensing range	[18] [19] [29] [56]
Communication	Bytes/s; message rate; delay distribution; age of information; drop and burst loss	Include clock synchronization and trajectory timestamp policy	Mesh vs centralized; congestion; disconnection; delayed/reordered packets	[5] [7] [19] [69]
Robustness	Success under wind, mass/model error, localization drift, detection error and prediction shift	Randomize jointly; publish seeds and uncertainty ranges	Nominal, domain-randomized, adversarial and out-of-distribution	[19] [27] [66]
Scalability	Performance vs UAV count, local degree and map volume	Distinguish centralized wall time, per-agent compute and total compute	2, 5, 10, 25, 50, 100+ agents where meaningful	[2] [17] [26] [66]
Liveness	Deadlock/livelock frequency; waiting time; recovery time	Collision-free but permanently blocked operation is a failure; apply the fixed τ_D and displacement and reversal thresholds defined in Section 3.3.	Swap, ring exchange, doorway, opposing flows, cul-de-sac	[9] [21] [51] [52]

Continued

Learning quality	Return, collision rate, sample efficiency, seed variance, OOD generalization	Include classical and safe-hybrid baselines at equal sensing/action limits	Unseen maps, team sizes, sensor noise, dynamics and policies	[20]-[25] [65]
Safety-filter burden	Intervention rate; action correction norm; infeasible-QP rate; fallback duration	Report performance with/without filter and why interventions occur	Dense simultaneous constraints, actuator saturation, degraded state estimate	[58]-[60] [65]
Energy	Modelled/measurement energy; reserve at completion; imbalance across fleet	State propulsion model, wind and payload; validate proxy against current/power where possible	Long range, climb, hover, task reassignment, return-to-home	[1] [6] [70]
Hardware maturity	Number of UAVs/flights; cumulative flight time; intervention and failure log	Report all trials, not only successful video; disclose safety pilot and motion-capture dependence; separate UF from US, MR and SU in each claimed result	Indoor mocap, outdoor GNSS, onboard vision/lidar, mixed dynamic obstacles	[16]-[19] [23] [36]
Reproducibility	Code, maps, logs, parameters, vehicle model and container/environment	Archive versions and licenses; provide standard import format	At least one public benchmark plus one application-specific scenario	[16] [17] [19] [23]
Regulatory fit	Geofence violations; strategic conflict; contingency compliance; remote-ID/network dependence	Separate algorithm performance from operational authorization	Urban low-altitude corridor, emergency priority, communication outage	[70]

8. Emerging Trends

8.1. Risk-Aware and Distributionally Robust Planning

The current state-of-the-art moves from a deterministic clearance for a particular forecast to risk-constrained planning in a multimodal uncertainty. The scenario MPC, distributionally robust optimization, conformal prediction, and reachable-set learning are capable to deliver a coverage regardless of the unknown probability distributions. The challenge is computational—the number of pairwise uncertain constraints grows with the size of the swarm and the planning horizon. The local interaction graph, shared latent intention, and adaptive risk assignment might make this approach computationally feasible. Conformal or distributionally robust coverage depends on the particular assumptions about exchangeability or the ambiguity-set; they cannot be claimed under any distribution shift.

8.2. Foundation Models and Language-Guided Autonomy

Large language and vision-language models are able to interpret mission rules, build a semantic cost map, resolve conflicts, or propose high-level contingency actions. The models are not meant to command collision-sensitive maneuvers without deterministic interface. The credible pipeline includes a foundation model that produces a goal, constraints, or plan candidates; a compiler to verify

their syntax and policy; an optimizer to assure their feasibility; and a safety controller to guarantee safe execution. Hallucinations, prompt injection with an external message, timing uncertainties, and auditability have to be considered.

8.3. Digital Twins, Differentiable Simulation, and Sim-to-Real Assurance

Simulations enable curriculum learning and generation of rare events, and digital twins can have platform-specific propulsion, sensors, networks, weather, and urban geometry. The differentiable physics and system identification allow better calibration. However, validation should not stop here—an assurance ladder approach is needed to move from unit tests and falsification through Monte Carlo simulations, software/hardware-in-the-loop, cage flight, controlled outdoor flight, and representative missions. The counterexamples discovered at every step are supposed to go back to training and validation phases.

8.4. Event-Triggered and Semantic Communication

Periodic broadcasting of the entire trajectory uses unnecessary bandwidth in the case of a predictable movement, overloads dense swarms. The event-driven broadcasting broadcasts changes once the prediction error, the collision risk, or the intent change passes a certain threshold. Semantic communication communicates latent task- or safety-relevant variables, not raw data. The planner and the network should be designed jointly such that the compression error and silence encode the safety signal. Security and authentication are crucial since spoofing of the intent can lead to intentional collisions and denial of service.

8.5. Low-Altitude Economy and Urban Air Mobility

Delivery and inspection missions of dense swarms involve strategic deconfliction at minutes-to-hours timescale and tactical avoidance at seconds-to-milliseconds timescale. Corridor assignment, altitude stratification, vertiport capacity, weather, noise, energy pricing, and priority traffic produce a multi-layered optimization problem. The swarm planning algorithms should interface with the traffic management system while maintaining safety under network degradation. It requires compositional guarantees in the strategic, tactical, and control layers.

8.6. Self-Reconfiguring Heterogeneous Swarms

In the future, the teams will reconfigure themselves in response to failure, recharge, communication relaying, and sensing roles. Hence, the planning should focus not only on trajectory but on interaction graph, information value, and task redundancy. Meta-learning and graph policies will help in the adaptation process while formal contracts will define the expectations from the neighbors. The graceful degradation resulting in partial mission completion while preserving safety is a preferable objective to nominal optimality.

9. Open Research Questions and Recommended Agenda

Safety with liveness: how can a planner ensure collision avoidance and progress in a symmetric conflict situation?

Bounding Uncertainty: What is a way to incorporate localization, tracking, intent, and communication uncertainty without being overly conservative about safety volumes? Create schemes of risk budgeting and decomposition, and dynamic online set adaptation.

Scalable continuous-time coordination: How to merge CBS-like combinatorial reasoning with nonlinear dynamics and continuous collision avoidance for tens to hundreds of UAVs? Consider hierarchical time windows, safe corridors, and parallel kinodynamic planning.

Learned policy certification: How to learn a policy that preserves the adaptive capabilities but provides a basis for certification? Policy verification over bounded regions, runtime monitoring, abstaining due to uncertainty, and design of a safety shield.

Motion and communication interaction: At which moments should agents communicate, what should they exchange, and how to design the motion for maintaining network connectivity? The age of information and reachable set growth should be optimized along with the trajectory optimization objective.

Representative benchmarks: Is there a possibility of agreeing upon 3D maps, dynamics, communication network traces, wind field models, and reporting format? The benchmark should include logs, failures, and hardware implementation.

Human and rule interaction: How do the issues of right-of-way, explainability, operator override, and regulation impact the optimizer? Operational rules encoding for machine checks and decision provenance.

Energy and sustainability: Can task assignment, scheduling, and battery state be optimized simultaneously without depleting reserves? The path length should be substituted with more realistic propulsion and degradation models.

Adversarial robustness: How should swarms react to localization attacks, broadcasting of malicious intent, jamming, or individual robot compromise? The combination of secure estimation, trust-based neighbor graph, and separation policies resilient to failure.

10. Practical Method-Selection Guide

No algorithm works in all cases. If we have a small swarm, known map, and strict dynamic requirements, the centralized or distributed MPC/SCP with continuous collision avoidance seems to be reasonable. In the case of a large sparse swarm operating in space, RVO and local CBF provide the opportunity for efficient decentralization. For a crowded environment without a prior map, it is better to use a hierarchical stack incremental global search algorithm, local trajectory optimization, prediction of neighbors' trajectories, and independent safety filters. Learning is useful for perception, intent prediction, cost shaping or generating a nominal policy in combination with constraint enforcement. The selection of the meth-

ods is an engineering recommendation depending on agent density, vehicle dynamics, feasibility of solvers, and communication health.

The choice of methods has to start with a guarantee contract: what is known? What are the bounded delays and tracking error? Do the other agents behave reciprocally? What happens if the solver fails or the communication is lost? Only then can the path length and average mission time be optimized. In critical autonomous systems, assumptions become a part of the algorithm.

11. Conclusions

The multi-UAV path planning and collision avoidance evolves from geometric routing to integrated cooperative autonomy. The algorithms for search and MAPF are proficient in combinatorial reasoning; sampling and meta-heuristic approaches handle complex nonconvex environments; velocity obstacle and dynamic window approaches make instant reaction possible; trajectory optimization and MPC take care of constraints; CBFs provide a safety margin; and MARL/GNN approaches provide adaptation in distributed inference. No limitation of any of these approaches precludes another one—it only complements it. The main take-away is that the hybrid multi-layer solution is needed, not the proclamation of the universal optimizing engine superiority. The coordination has to provide a variety of routes and liveness of the solutions; prediction has to account for uncertain dynamics and possibly outdated estimates; optimization has to result in trackable control inputs; and the safety-critical layer is necessary to provide safety during execution as soon as assumptions fail. The communication delays, downwash effect, energy reserves, solver tail latencies, and hardware faults have to become the routine evaluation variables. The success has to be measured through reproducible risk-aware performance through the assurance ladder from simulation to prolonged field flight. The proposed stack is just a design hypothesis; end-to-end safety and liveness require verified interfaces and hardware evidence.

Conflicts of Interest

The authors declare no known competing financial interests or personal relationships that could have appeared to influence the work reported in this manuscript.

References

- [1] Nagasawa, R., Mas, E., Moya, L. and Koshimura, S. (2021) Model-Based Analysis of Multi-UAV Path Planning for Surveying Postdisaster Building Damage. *Scientific Reports*, **11**, Article No. 18599. <https://doi.org/10.1038/s41598-021-97804-4>
- [2] Chung, S., Paranjape, A.A., Dames, P., Shen, S. and Kumar, V. (2018) A Survey on Aerial Swarm Robotics. *IEEE Transactions on Robotics*, **34**, 837-855. <https://doi.org/10.1109/tro.2018.2857475>
- [3] Coppola, M., McGuire, K.N., De Wagter, C. and de Croon, G.C.H.E. (2020) A Survey on Swarming with Micro Air Vehicles: Fundamental Challenges and Constraints. *Frontiers in Robotics and AI*, **7**, Article ID: 18. <https://doi.org/10.3389/frobt.2020.00018>

- [4] Schranz, M., Umlauf, M., Sende, M. and Elmenreich, W. (2020) Swarm Robotic Behaviors and Current Applications. *Frontiers in Robotics and AI*, **7**, Article ID: 36. <https://doi.org/10.3389/frobt.2020.00036>
- [5] Champion, M., Ranganathan, P. and Faruque, S. (2019) UAV Swarm Communication and Control Architectures: A Review. *Journal of Unmanned Vehicle Systems*, **7**, 93-106. <https://doi.org/10.1139/juvs-2018-0009>
- [6] Khamis, A., Hussein, A. and Elmogy, A. (2015) Multi-robot Task Allocation: A Review of the State-of-the-Art. In: Koubâa, A. and Martínez-de Dios, J., Eds., *Studies in Computational Intelligence*, Springer International Publishing, 31-51. https://doi.org/10.1007/978-3-319-18299-5_2
- [7] Gielis, J., Shankar, A. and Prorok, A. (2022) A Critical Review of Communications in Multi-Robot Systems. *Current Robotics Reports*, **3**, 213-225. <https://doi.org/10.1007/s43154-022-00090-9>
- [8] Hart, P., Nilsson, N. and Raphael, B. (1968) A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, **4**, 100-107. <https://doi.org/10.1109/tssc.1968.300136>
- [9] Sharon, G., Stern, R., Felner, A. and Sturtevant, N.R. (2015) Conflict-Based Search for Optimal Multi-Agent Pathfinding. *Artificial Intelligence*, **219**, 40-66. <https://doi.org/10.1016/j.artint.2014.11.006>
- [10] Barer, M., Sharon, G., Stern, R. and Felner, A. (2014) Suboptimal Variants of the Conflict-Based Search Algorithm for the Multi-Agent Pathfinding Problem. In: *Frontiers in Artificial Intelligence and Applications*, IOS Press, 19-27. <https://doi.org/10.3233/978-1-61499-419-0-961>
- [11] Li, J., Harabor, D., Stuckey, P.J., Felner, A., Ma, H. and Koenig, S. (2019) Disjoint Splitting for Multi-Agent Path Finding with Conflict-Based Search. *Proceedings of the International Conference on Automated Planning and Scheduling*, **29**, 279-283. <https://doi.org/10.1609/icaps.v29i1.3487>
- [12] Andreychuk, A., Yakovlev, K., Atzmon, D. and Stern, R. (2019) Multi-Agent Pathfinding with Continuous Time. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 10-16 August 2019, 39-45. <https://doi.org/10.24963/ijcai.2019/6>
- [13] Lin, W., Song, W., Zhu, Q. and Zhu, S. (2024) Multi-Agent Path Finding with Heterogeneous Geometric and Kinematic Constraints in Continuous Space. *IEEE Robotics and Automation Letters*, **10**, 492-499.
- [14] Rezaee, M.R., Hamid, N.A.W.A., Hussin, M., *et al.* (2024) Comprehensive Review of Drones Collision Avoidance Technologies. *IEEE Sensors Journal*, **24**, 6397-6426.
- [15] Sharma, G., Jain, S. and Sharma, R.S. (2025) Path Planning for Fully Autonomous UAVs-A Taxonomic Review and Future Perspectives. *IEEE Access*, **13**, 13356-13379. <https://doi.org/10.1109/access.2025.3529754>
- [16] Zhou, X., Zhu, J., Zhou, H., Xu, C. and Gao, F. (2021) Ego-Swarm: A Fully Autonomous and Decentralized Quadrotor Swarm System in Cluttered Environments. 2021 *IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, 30 May 2021-5 June 2021, 4101-4107. <https://doi.org/10.1109/icra48506.2021.9561902>
- [17] Zhou, X., Wen, X., Wang, Z., Gao, Y., Li, H., Wang, Q., *et al.* (2022) Swarm of Micro Flying Robots in the Wild. *Science Robotics*, **7**, eabm5954. <https://doi.org/10.1126/scirobotics.abm5954>
- [18] Tordesillas, J. and How, J.P. (2022) MADER: Trajectory Planner in Multiagent and Dynamic Environments. *IEEE Transactions on Robotics*, **38**, 463-476.

- <https://doi.org/10.1109/tro.2021.3080235>
- [19] Kondo, K., Tordesillas, J., Figueroa, R., Rached, J., Merkel, J., Lusk, P.C., *et al.* (2023) Robust MADER: Decentralized and Asynchronous Multiagent Trajectory Planner Robust to Communication Delay. 2023 *IEEE International Conference on Robotics and Automation (ICRA)*, London, 29 May 2023-2 June 2023, 1687-1693. <https://doi.org/10.1109/icra48891.2023.10161244>
 - [20] Chen, Y.F., Liu, M., Everett, M. and How, J.P. (2017) Decentralized Non-Communicating Multiagent Collision Avoidance with Deep Reinforcement Learning. 2017 *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 29 May 2017-3 June 2017, 285-292. <https://doi.org/10.1109/icra.2017.7989037>
 - [21] Semnani, S.H., Liu, H., Everett, M., de Ruiter, A. and How, J.P. (2020) Multi-Agent Motion Planning for Dense and Dynamic Environments via Deep Reinforcement Learning. *IEEE Robotics and Automation Letters*, **5**, 3221-3226. <https://doi.org/10.1109/lra.2020.2974695>
 - [22] Fan, T., Long, P., Liu, W. and Pan, J. (2020) Distributed Multi-Robot Collision Avoidance via Deep Reinforcement Learning for Navigation in Complex Scenarios. *The International Journal of Robotics Research*, **39**, 856-892. <https://doi.org/10.1177/0278364920916531>
 - [23] Batra, S., Huang, Z., Petrenko, A., Kumar, T., Molchanov, A. and Sukhatme, G.S. (2022) Decentralized Control of Quadrotor Swarms with End-to-End Deep Reinforcement Learning. *5th Conference on Robot Learning*, **164**, 576-586.
 - [24] Zhao, F., Zeng, Y., Han, B., Fang, H. and Zhao, Z. (2022) Nature-Inspired Self-Organizing Collision Avoidance for Drone Swarm Based on Reward-Modulated Spiking Neural Network. *Patterns*, **3**, Article 100611. <https://doi.org/10.1016/j.patter.2022.100611>
 - [25] Liang, C., Liu, L. and Liu, C. (2023) Multi-UAV Autonomous Collision Avoidance Based on PPO-GIC Algorithm with CNN-LSTM fusion Network. *Neural Networks*, **162**, 21-33. <https://www.sciencedirect.com/science/article/abs/pii/S0893608023000904>
 - [26] Hou, J., Zhou, X., Gan, Z. and Gao, F. (2022) Enhanced Decentralized Autonomous Aerial Robot Teams with Group Planning. *IEEE Robotics and Automation Letters*, **7**, 9240-9247. <https://doi.org/10.1109/lra.2022.3191037>
 - [27] Kuwata, Y. and How, J.P. (2011) Cooperative Distributed Robust Trajectory Optimization Using Receding Horizon MILP. *IEEE Transactions on Control Systems Technology*, **19**, 423-431. <https://doi.org/10.1109/tcst.2010.2045501>
 - [28] Chen, Y., Cutler, M. and How, J.P. (2015) Decoupled Multiagent Path Planning via Incremental Sequential Convex Programming. 2015 *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, 26-30 May 2015, 5954-5961. <https://doi.org/10.1109/icra.2015.7140034>
 - [29] Luis, C.E., Vukosavljev, M. and Schoellig, A.P. (2020) Online Trajectory Generation with Distributed Model Predictive Control for Multi-Robot Motion Planning. *IEEE Robotics and Automation Letters*, **5**, 604-611. <https://doi.org/10.1109/lra.2020.2964159>
 - [30] Toumeh, C. and Lambert, A. (2022) Decentralized Multi-Agent Planning Using Model Predictive Control and Time-Aware Safe Corridors. *IEEE Robotics and Automation Letters*, **7**, 11110-11117. <https://doi.org/10.1109/lra.2022.3196777>
 - [31] Li, B., Song, C., Bai, S., Huang, J., Ma, R., Wan, K., *et al.* (2023) Multi-UAV Trajectory Planning during Cooperative Tracking Based on a Fusion Algorithm Integrating

- MPC and Standoff. *Drones*, **7**, Article 196. <https://doi.org/10.3390/drones7030196>
- [32] Park, J., Kim, D., Kim, G.C., Oh, D. and Kim, H.J. (2022) Online Distributed Trajectory Planning for Quadrotor Swarm with Feasibility Guarantee Using Linear Safe Corridor. *IEEE Robotics and Automation Letters*, **7**, 4869-4876. <https://doi.org/10.1109/lra.2022.3152702>
- [33] Koenig, S. and Likhachev, M. (2002) D* Lite. *Proceedings of the AAAI Conference on Artificial Intelligence*, **18**, 476-483.
- [34] Moldagalieva, A., Ortiz-Haro, J., Toussaint, M. and Hönig, W. (2024) DB-CBS: Discontinuity-Bounded Conflict-Based Search for Multi-Robot Kinodynamic Motion Planning. 2024 *IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, 13-17 May 2024, 14569-14575. <https://doi.org/10.1109/icra57147.2024.10610999>
- [35] LaValle, S.M. (1998) Rapidly-Exploring Random Trees: A New Tool for Path Planning. Technical Report, Iowa State University.
- [36] Preiss, J.A., Honig, W., Ayanian, N. and Sukhatme, G.S. (2017) Downwash-Aware Trajectory Planning for Large Quadrotor Teams. 2017 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, 24-28 September 2017, 250-257. <https://doi.org/10.1109/iros.2017.8202165>
- [37] Karaman, S. and Frazzoli, E. (2011) Sampling-Based Algorithms for Optimal Motion Planning. *The International Journal of Robotics Research*, **30**, 846-894. <https://doi.org/10.1177/0278364911406761>
- [38] Robinson, D.R., Mar, R.T., Estabridis, K. and Hewer, G. (2018) An Efficient Algorithm for Optimal Trajectory Generation for Heterogeneous Multi-Agent Systems in Non-Convex Environments. *IEEE Robotics and Automation Letters*, **3**, 1215-1222. <https://doi.org/10.1109/lra.2018.2794582>
- [39] Ngo, A.P., Thomas, C., Karimoddini, A. and Nguyen, H.T. (2023) Multi-UAV Trajectory Planning Problem Using the Difference of Convex Function Programming. 2023 62nd *IEEE Conference on Decision and Control (CDC)*, Singapore, 13-15 December 2023, 7367-7372. <https://doi.org/10.1109/cdc49753.2023.10383259>
- [40] Park, J., Kim, J., Jang, I. and Kim, H.J. (2025) Decentralized Deadlock-Free Trajectory Planning for Quadrotor Swarm in Obstacle-Rich Environments. 2023 *IEEE International Conference on Robotics and Automation (ICRA)*, London, 29 May 2023-2 June 2023, 1428-1434. <https://ieeexplore.ieee.org/document/10160847>
- [41] Richards, A. and How, J.P. (2002) Aircraft Trajectory Planning with Collision Avoidance Using Mixed Integer Linear Programming. *Proceedings of the 2002 American Control Conference*, Anchorage, 8-10 May 2002, 1936-1941. <https://doi.org/10.1109/acc.2002.1023918>
- [42] Sathiyaraj, B.M., Jain, L.C., Finn, A. and Drake, S. (2008) Multiple UAVs Path Planning Algorithms: A Comparative Study. *Fuzzy Optimization and Decision Making*, **7**, 257-267. <https://doi.org/10.1007/s10700-008-9035-0>
- [43] Zhang, X. and Duan, H. (2015) An Improved Constrained Differential Evolution Algorithm for Unmanned Aerial Vehicle Global Route Planning. *Applied Soft Computing*, **26**, 270-284. <https://doi.org/10.1016/j.asoc.2014.09.046>
- [44] Chen, Y., Yu, J., Mei, Y., Zhang, Y. and Ai, X. (2016) Modified Central Force Optimization (MCFO) Algorithm for 3D UAV Path Planning. *Neurocomputing*, **171**, 878-888. <https://doi.org/10.1016/j.neucom.2015.07.044>
- [45] Chu, H., Yi, J. and Yang, F. (2022) Chaos Particle Swarm Optimization Enhancement Algorithm for UAV Safe Path Planning. *Applied Sciences*, **12**, Article 8977.

- <https://doi.org/10.3390/app12188977>
- [46] Cheng, J. (2023) Dynamic Path Optimization Based on Improved Ant Colony Algorithm. *Journal of Advanced Transportation*, **2023**, Article ID: 7651100. <https://doi.org/10.1155/2023/7651100>
 - [47] Zhang, Z., Liu, X. and Feng, B. (2023) Research on Obstacle Avoidance Path Planning of UAV in Complex Environments Based on Improved Bézier Curve. *Scientific Reports*, **13**, Article No. 16453. <https://doi.org/10.1038/s41598-023-43783-7>
 - [48] Wang, X., Feng, Y., Tang, J., Dai, Z. and Zhao, W. (2024) A UAV Path Planning Method Based on the Framework of Multi-Objective Jellyfish Search Algorithm. *Scientific Reports*, **14**, Article No. 28058. <https://doi.org/10.1038/s41598-024-79323-0>
 - [49] Khatib, O. (1986) Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, **5**, 90-98. <https://doi.org/10.1177/027836498600500106>
 - [50] Hu, Y., Wu, Y. and Chen, W. (2025) Multi-UAV Swarm Obstacle Avoidance Based on Potential Field Optimization. arXiv:2511.16911.
 - [51] Fiorini, P. and Shiller, Z. (1998) Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research*, **17**, 760-772. <https://doi.org/10.1177/027836499801700706>
 - [52] van den Berg, J., Guy, S.J., Lin, M. and Manocha, D. (2011) Reciprocal N-Body Collision Avoidance. In: Pradalier, C., Siegwart, R. and Hirzinger, G., Eds., *Springer Tracts in Advanced Robotics*, Springer, 3-19. https://doi.org/10.1007/978-3-642-19457-3_1
 - [53] Zhou, D., Wang, Z., Bandyopadhyay, S. and Schwager, M. (2017) Fast, On-Line Collision Avoidance for Dynamic Vehicles Using Buffered Voronoi Cells. *IEEE Robotics and Automation Letters*, **2**, 1047-1054. <https://doi.org/10.1109/lra.2017.2656241>
 - [54] Arul, S.H. and Manocha, D. (2021) V-RVO: Decentralized Multi-Agent Collision Avoidance Using Voronoi Diagrams and Reciprocal Velocity Obstacles. 2021 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, 27 September 2021-1 October 2021, 8097-8104. <https://doi.org/10.1109/iro51168.2021.9636618>
 - [55] Fox, D., Burgard, W. and Thrun, S. (1997) The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics & Automation Magazine*, **4**, 23-33. <https://doi.org/10.1109/100.580977>
 - [56] Chang, X., Wang, J., Li, K., Zhang, X. and Tang, Q. (2025) Research on Multi-UAV Autonomous Obstacle Avoidance Algorithm Integrating Improved Dynamic Window Approach and Orca. *Scientific Reports*, **15**, Article No. 14646. <https://doi.org/10.1038/s41598-025-99111-8>
 - [57] Ramezani, M., Habibi, H., Sanchez-Lopez, J.L. and Voos, H. (2023) UAV Path Planning Employing MPC-Reinforcement Learning Method Considering Collision Avoidance. 2023 *International Conference on Unmanned Aircraft Systems (ICUAS)*, Warsaw, 6-9 June 2023, 507-514. <https://doi.org/10.1109/icuas57906.2023.10156232>
 - [58] Ames, A.D., Xu, X., Grizzle, J.W. and Tabuada, P. (2017) Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Transactions on Automatic Control*, **62**, 3861-3876. <https://doi.org/10.1109/tac.2016.2638961>
 - [59] Wang, L., Ames, A.D. and Egerstedt, M. (2017) Safety Barrier Certificates for Collisions-Free Multirobot Systems. *IEEE Transactions on Robotics*, **33**, 661-674. <https://doi.org/10.1109/tro.2017.2659727>
 - [60] Chen, Y., Singletary, A. and Ames, A.D. (2021) Guaranteed Obstacle Avoidance for

- Multi-Robot Operations with Limited Actuation: A Control Barrier Function Approach. *IEEE Control Systems Letters*, **5**, 127-132.
<https://doi.org/10.1109/lcsys.2020.3000748>
- [61] Lv, X., Peng, C. and Ma, J. (2024) Control Barrier Function-Based Collision Avoidance Guidance Strategy for Multi-Fixed-Wing UAV Pursuit-Evasion Environment. *Drones*, **8**, Article 415. <https://doi.org/10.3390/drones8080415>
- [62] Zhao, J., Zhong, L., Li, H. and Hou, Z. (2024) Safe Formation Control and Collision Avoidance for Multi-UAV Systems Using Barrier Lyapunov Function. 2024 43rd *Chinese Control Conference*, Kunming, 28-31 July 2024, 5504-5510.
- [63] Su, W., Gao, M., Gao, X. and Xuan, Z. (2024) Enhanced Multi-UAV Path Planning in Complex Environments with Voronoi-Based Obstacle Modelling and Q-Learning. *International Journal of Aerospace Engineering*, **2024**, Article ID: 5114696.
<https://doi.org/10.1155/2024/5114696>
- [64] Marino, A., Pacchierotti, C. and Giordano, P.R. (2024) Multi-UAVs End-to-End Distributed Trajectory Generation over Point Cloud Data. *IEEE Robotics and Automation Letters*, **9**, 7629-7636. <https://doi.org/10.1109/lra.2024.3426374>
- [65] Vinod, A.P., Safaoui, S., Chakrabarty, A., Quirynen, R., Yoshikawa, N. and Di Cairano, S. (2022) Safe Multi-Agent Motion Planning via Filtered Reinforcement Learning. 2022 *International Conference on Robotics and Automation (ICRA)*, Philadelphia, 23-27 May 2022, 7270-7276.
<https://doi.org/10.1109/icra46639.2022.9812259>
- [66] Wang, Z., Xu, C. and Gao, F. (2022) Robust Trajectory Planning for Spatial-Temporal Multi-Drone Coordination in Large Scenes. 2022 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, 23-27 October 2022, 12182-12188.
<https://doi.org/10.1109/iros47612.2022.9982032>
- [67] Ryou, G., Tal, E. and Karaman, S. (2022) Cooperative Multi-Agent Trajectory Generation with Modular Bayesian Optimization. *Robotics: Science and Systems XVIII*, New York, 27 June-1 July 2022. <https://doi.org/10.15607/rss.2022.xviii.060>
- [68] Zhai, L., Wu, H., Lai, L. and Gao, Z. (2025) Intelligent Optimization Algorithms for Multi-UAV Path Planning: A Comprehensive Review. *IEEE Access*, **13**, 101106-101130. <https://doi.org/10.1109/access.2025.3577598>
- [69] Gao, Y., Wang, Y., Zhong, X., Yang, T., Wang, M., Xu, Z., *et al.* (2022) Meeting-merging-Mission: A Multi-Robot Coordinate Framework for Large-Scale Communication-Limited Exploration. 2022 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, 23-27 October 2022, 13700-13707.
<https://doi.org/10.1109/iros47612.2022.9981544>
- [70] Checker, L., Xie, H., Khaksar, S. and Murray, I. (2025) Systematic Review of Multi-Objective UAV Swarm Mission Planning Systems from Regulatory Perspective. *Drones*, **9**, Article 509. <https://doi.org/10.3390/drones9070509>