DOI:10.11918/202510107

自适应 A* 算法增强时空优化的无人机协同轨迹规划

王羽楠[1], 罗 炯[2], 刘宇寰[2], 化永朝[1], 李晓多[3], 董希旺[1,2,3]

(1. 北京航空航天大学 人工智能学院,北京 100191; 2. 北京航空航天大学 自动化科学与电气工程学院,北京 100191; 3. 北京航空航天大学 无人系统研究院,北京 100191)

摘 要: 为解决分层规划中的前端基于固定分辨率的路径搜索效率与精度不能兼顾、后端多约束的时空联合优化计算复杂度高且求解困难的问题,提出了一种基于自适应 A*-MINCO(adaptive A*-minimum control, AAM)规划器的分层运动规划框架。首先,该框架由前端自适应搜索模块与后端时空联合优化模块组成,并以滚动时域的方式协同运行,实现从几何路径生成到动态可行轨迹的全流程规划。其次,前端部分设计的自适应 A* 算法能够根据环境结构复杂度动态调整搜索分辨率,从而在开阔区域保持高效率、在狭窄区域保证高精度;后端部分则构建了基于 MINCO 参数化的时空联合优化模型,将轨迹平滑度、时间效率、动力学可行性、环境安全,以及集群协同避碰等多目标整合于一个可微的代价函数中。最后,采用 L-BFGS (limited-memory broyden-fletcher-goldfarb-shanno)算法实现高效求解。在立柱阵列、窄墙道,以及长墙壁等多种复杂仿真场景中的对比实验表明,所提出的 AAM 规划器在规划成功率、计算效率和轨迹质量上均优于基线方法;在保证高规划成功率和轨迹平滑性的前提下,其平均规划效率提升 18.4%,轨迹总代价平均降低 15.3%。

关键词: 无人机集群;轨迹规划;运动规划;时空优化;避障控制

中图分类号: TP13 **文献标志码**: A **文章编号**: 0367-6234(2026)08-0113-12

Adaptive A* algorithm-enhanced spatiotemporal optimization for unmanned aerial vehicle cooperative trajectory planning

Wang Yunan[1], Luo Jiong[2], Liu Yuhuan[2], Hua Yongzhao[1], Li Xiaoduo[3], Dong Xiwang[1,2,3]

(1. School of Artificial Intelligence, Beihang University, Beijing 100191, China; 2. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China; 3. Institute of Unmanned System, Beihang University, Beijing 100191, China)

Abstract: To solve the problems that front-end path search based on a fixed resolution in hierarchical planning cannot balance efficiency and accuracy, and back-end spatiotemporal joint optimization with multiple constraints has high computational complexity and is difficult to solve, this paper proposed a hierarchical motion planning framework based on an adaptive A*-minimum control (AAM) planner. First, this framework consisted of a front-end adaptive search module and a back-end spatiotemporal joint optimization module and operated cooperatively in a receding horizon manner to achieve full-process planning from geometric path generation to dynamically feasible trajectories. Second, the adaptive A* algorithm designed in the front-end part could dynamically adjust the search resolution according to the environmental structure complexity, thereby maintaining high efficiency in open areas and ensuring high precision in narrow areas. The back-end part constructed a spatiotemporal joint optimization model based on MINCO parameterization, integrating multiple objectives such as trajectory smoothness, time efficiency, dynamic feasibility, environmental safety, and swarm cooperative collision avoidance into a differentiable cost function. Finally, this paper adopted the limited-memory broyden-fletcher-goldfarb-shanno (L-BFGS) algorithm to achieve an efficient solution. Comparative experiments in multiple complex simulation scenarios such as pole arrays, narrow gaps, and long walls indicate that the proposed AAM planner outperforms the baseline methods in planning success rate, computational efficiency, and trajectory quality; under the premise of ensuring a high planning success rate and trajectory smoothness, its average planning efficiency increases by 18.4%, and the total trajectory cost decreases by 15.3% on average.

Keywords: unmanned aerial vehicle swarm; trajectory planning; motion planning; spatiotemporal optimization; obstacle avoidance control

收稿日期: 2025-10-31;**录用日期**: 2025-11-25;**网络首发日期**: 2026-05-13
网络首发地址: https://link.cnki.net/urlid/23.1235.T.20260513.1709.005
基金项目: 国家自然科学基金(62473027, U2241217, 62473029, 62403038, 62388101)
作者简介: 王羽楠(2002—),女,硕士研究生;化永朝(1992—),男,副教授,博士生导师;董希旺(1985—),男,教授,博士生导师
通信作者: 化永朝,yongzhaohua@buaa.edu.cn

无人机(unmanned aerial vehicle, UAV)集群凭借其强鲁棒性与高任务弹性,已成为协同侦察、灾害响应等领域的关键平台[1-3]。Jones 等[4]指出,随着无人机集群应用规模扩大,如何在复杂环境下同时兼顾全局路径规划的最优性与实时响应能力,仍是当前研究面临的核心挑战之一。这类集群任务需同时满足局部感知约束、毫秒级严苛实时性要求与多机协同性,对运动规划方法的综合性能提出较高要求。

针对无人机在线轨迹规划问题,目前已形成基于搜索、采样、优化,以及混合方法等多种技术路线。在前端路径规划中,基于搜索的方法如 A* 算法[5]及其变体(如 Hybrid A[6])能够在栅格地图中提供最优或近似最优路径,但其性能严重依赖固定分辨率设定,难以兼顾效率与质量。虽有多尺度分层方法尝试改善,但仍缺乏根据环境特征动态调整分辨率的机制。基于采样的方法及其优化变体[7-8]借助随机采样适应高维规划问题,但存在解质量随机、收敛时间不稳定等问题,在狭窄通道等复杂场景中性能下降。

后端轨迹优化旨在将前端路径转化为平滑、动态可行且安全的连续轨迹。经典方法中,如最小化“snap”的四阶多项式轨迹[9]等,奠定了轨迹平滑性与可行性优化的基础;如改进人工势场法[10]及高阶连续点对点轨迹规划算法[11]等,提升了轨迹的连续性与安全性。然而这些方法大多采用固定时间参数或保守约束,难以实现时间与空间的有效联合优化。近年来,最小控制(minimum control effort, MINCO)轨迹类方法[12-13]通过轨迹形状与时间分配的解耦,为高效的时空联合优化提供了新途径;FASTER(fast and safe trajectory planner)[14]则通过约束凸化进一步提升了求解效率,推动了后端优化方法的发展。

在集群协同规划方面,集中式优化由于计算复杂度高、对通信同步依赖强,难以满足实时在线应用的需求。基于速度障碍(velocity obstacle, VO)方法[15]虽可实现避碰,但规划结果往往次优;而基于分布式优化的方法,如基于 EGO-Planner 演进的集群规划器(ESDF-free Gradient-based onboard planner for swarm, EGO-Swarm)[16]通过引入基于距离场的惩罚项和相邻信息广播机制,实现对邻机状态的局部感知与避碰优化;Kondo 等[17-18]则提出通过参数广播与异步优化机制提高了通信延迟与动态环境下的鲁棒性。Adajania 等[19]提出通过对障碍物可行空间进行隐式非凸分解,实现了在复杂环境中的多机动态协调;Hou 等[20]提出的分布式组规划框架则通过将无人机团队划分为子群体,提高了大型集群任务中的可扩展性与协调效率;基于学习的协同导航方法,如 Huang 等[21]使用深度强化学习方法训练的多机协同导航策略在未知环境中表现出良好的适应性。

然而现有方法仍存在以下不足:1)前端路径搜索依赖固定分辨率设定,高分辨率虽能保证精度但计算开销大,难以满足实时性要求;低分辨率虽提升效率,却易遗漏狭窄通道信息,造成规划失败或路径质量不佳。2)后端轨迹优化多将空间路径与时间分配分离处理,这种时空解耦模式会牺牲轨迹的灵活性与全局最优性;而时空联合优化因计算复杂度高,往往需采用保守简化策略,限制了动态性能发挥。3)集群协同场景下,实时避碰需求与轨迹最优性之间难以实现有效平衡。基于上述问题,本文提出一种面向无人机集群的自适应分层在线轨迹规划框架——自适应 A*-MINCO 规划器(adaptive A*-MINCO planner)。该框架通过前端自适应搜索与后端 MINCO 优化的高效协同,提升了规划质量与响应速度。

1 协同轨迹规划问题建模

1.1 环境建模

为实现高效且可微的障碍物规避,本文采用八叉树结构的占据栅格地图(OctoMap)管理原始点云数据,并在此基础上构建欧几里得符号距离场[22]。ESDF 为空间中任意点 p 赋予一个标量值,其定义为

$$\Phi(p)=\begin{cases}+d(p,O), & p\in F(\text{自由空间})\\ -d(p,O), & p\in O(\text{障碍物})\end{cases} \tag{1}$$

式中,$d(p,O)=\min\limits_{q\in O}\|p-q\|_2$,表示点 p 到最近障碍物 ∂O 的欧氏距离。ESDF 的优势在于它是一个连续函数,并且能够提供解析梯度 $\nabla\Phi(p)$。

1.2 时空解耦的轨迹参数化

多旋翼无人机具有微分平坦特性,其全状态 $x=\{p,v,R,\omega\}$ 和控制输入 $u=\{f,\tau\}$ 可由平坦输出 $z=\{p,\psi\}$ 及其有限阶导数通过映射 $(x,u)=\Phi(z^{[s]})$ 唯一确定[12]。因此,全状态轨迹优化可等价地转化为平坦输出空间轨迹 $z(t)$ 的优化。鉴于偏航角 ψ 通常独立规划,本文主要优化无人机在欧氏空间中运动轨迹 $p(t)\in\mathbf{R}^3$。

在轨迹系数的线性求解上,采用分段多项式来表示轨迹。本文采用最小控制轨迹[12]框架,它将轨迹生成问题构建为一个最优控制问题:在满足一系列航点和连续性约束的前提下,最小化控制输入的某个积分范数(例如,最小化 Jerk 或 Snap 的平方积

分),从而获得最平滑、控制能耗最低的轨迹。

根据最优控制理论[12],MINCO 轨迹由 M 段 $2s-1$ 次多项式构成。其核心在于,所有多项式系数矩阵 $\boldsymbol{c}$ 可以通过求解一个由航点和连续性约束导出的线性方程组 $\boldsymbol{Ac}=\boldsymbol{b}$ 唯一确定[12]。通过求解上述良定义的线性系统,建立了一个从航点 $\boldsymbol{q}$、时间 $\boldsymbol{T}$ 到轨迹系数 c 的确定性映射,记为 $c=C(\boldsymbol{q},\boldsymbol{T})$。上述映射 $c=C(\boldsymbol{q},\boldsymbol{T})$ 是实现时空解耦的核心机制。这意味着将复杂的轨迹优化问题分解为两个更直观、更低维的部分:1)空间路径。轨迹的几何形状主要由中间航点序列 $\boldsymbol{q}$ 决定。2)时间剖面。轨迹的运动速度和时序主要由时间分配向量 $\boldsymbol{T}$ 决定。最后,本文所采用的轨迹参数化形式及最小控制量轨迹可被表示为

$$\xi_{\text{MINCO}}=\{p(t):[0,T]\to\mathbf{R}^m \mid c=C(\boldsymbol{q},\boldsymbol{T}),\ \boldsymbol{q}\in\mathbf{R}^{m\times(M-1)},\boldsymbol{T}\in\mathbf{R}^M_{>0}\} \tag{2}$$

式中,$\boldsymbol{q}$ 与 $\boldsymbol{T}$ 是独立的,实现了轨迹时间域与空间域解耦的参数化,后续将采用该轨迹类进行规划。

1.3　优化目标与约束

基于轨迹参数化方法,本文通过明确决策变量、约束条件与优化目标,将协同轨迹规划问题构建为一个非线性优化模型。

1.3.1　决策变量

优化的核心决策变量是航点序列 $\boldsymbol{q}$ 与时间分配向量 $\boldsymbol{T}$ 的组合,即

$$\theta\triangleq(\boldsymbol{q},\boldsymbol{T})\in Q\times T \tag{3}$$

式中,$T=\{T_i\in[T_{\min},T_{\max}],\ i=1,\cdots,M\}$ 约束了每段时长在一个合理范围内。

为处理时间 T_i 的下界和上界约束(例如 $T_i\in[T_{\min},T_{\max}]$),引入一个无约束的代理变量 $\tau\in\mathbf{R}^M$,并通过一个单调双射函数 g(如 Sigmoid 函数)建立映射关系 $T=g(\tau)$。

为便于使用无约束优化器,引入代理变量 $\tau\in\mathbf{R}^M$ 和一个单调双射函数 $g:\mathbf{R}\to[T_{\min},T_{\max}]$,如 Sigmoid 函数的变种 $g(\tau)=T_{\min}+(T_{\max}-T_{\min})\sigma(\tau)$,使得 $T=g(\tau)$,从而将有界约束转化为无界优化。

1.3.2　约束条件

为了利用高效的无约束优化算法,将各类约束转化为目标函数中的软约束。

1)动力学约束。对速度和加速度的限制为 $\|\dot{p}(t;\theta)\|\leqslant v_m$,$\|\ddot{p}(t;\theta)\|\leqslant a_m$。

2)环境碰撞约束。要求轨迹与障碍物保持最小安全距离 r_{safe} 为 $\Phi(p(t;\theta))\geqslant r_{\text{safe}}$。

3)集群协同避碰约束。对于任意邻机轨迹 $p_j(t)$,要求保持最小安全间距 r_{D} 为 $\|p(t;\theta)-p_j(t)\|\geqslant r_{\text{D}}$。

1.3.3　目标函数

将多个性能指标加权组合成一个单一的、可微的代价函数 $J(\theta)$,优化目标为最小化则该函数为

$$\min_\theta J(\theta)=\lambda_{\text{e}}J_{\text{e}}(\theta)+\lambda_{\text{t}}J_{\text{t}}(T)+\lambda_{\text{d}}J_{\text{d}}(\theta)+\lambda_{\text{o}}J_{\text{o}}(\theta;\boldsymbol{\Phi})+\lambda_{\text{r}}J_{\text{r}}(\theta;\boldsymbol{\Phi}_{\text{neigh}}) \tag{4}$$

式中:J_{e} 为控制代价,$J_{\text{e}}(\theta)=\int_0^{T_{\text{tot}}}\|\dddot{p}(t;\theta)\|^2\text{d}t$,通常用控制能量代理来衡量;$J_{\text{t}}$ 为时间代价,$J_{\text{t}}(T)=\sum_{i=1}^{M}T_i$,表示最小化总飞行时间;$J_{\text{d}}$ 为动力学惩罚代价,用于惩罚超出 v_{m} 和 a_{m} 限制的轨迹部分;J_{o} 为障碍物碰撞惩罚代价,基于 ESDF 值 $\Phi(p)$ 构建,当轨迹与障碍物的距离小于安全阈值 r_{safe} 时产生的光滑惩罚项;J_{r} 为集群避碰惩罚代价,当无人机间的时空距离小于安全阈值时产生的光滑惩罚项。

通过 MINCO 的映射关系和链式法则,代价函数对决策变量 $\boldsymbol{q}$、$\boldsymbol{T}$ 的梯度可以被高效计算,从而能够采用 L-BFGS[23] 等基于梯度的优化算法进行求解。

2　时空协同轨迹规划算法

2.1　框架总览

AAM 规划器整体架构如图 1 所示,集成了感知、前端路径搜索、后端轨迹优化,以及集群协同模块,为无人机集群实时生成安全、平滑的轨迹。图 1 构成一个闭环的在线规划与控制系统,以滚动时域方式持续运行。其工作流程始于感知模块,该模块通过深度相机获取环境点云,并构建全局统一的占据栅格地图与欧几里得符号距离场。其次,规划模块作为核心,接收来自有限状态机(finite state machine,FSM)的局部目标点以及 ESDF 地图,执行两阶段规划:前端的自适应 A* 算法依据环境复杂度动态调整分辨率,高效生成一条无碰撞的初始引导路径;该路径被转化为初始航点向量 $\boldsymbol{q}$ 与时间分配向量 $\boldsymbol{T}$,作为后端时空轨迹优化节点的初值。该优化节点基于 MINCO 框架,融合平滑度、时间、动力学,以及安全等多重目标,并结合通信模块接收到的其他无人机轨迹信息进行协同避碰,生成一条安全、平滑且动态可行的优化轨迹。最后,控制模块将该轨迹传递给 PX4 底层控制器执行,同时无人机自身的位姿状态被反馈回系统,启动下一轮的重规划循环,从而实现对未知环境的持续自适应。

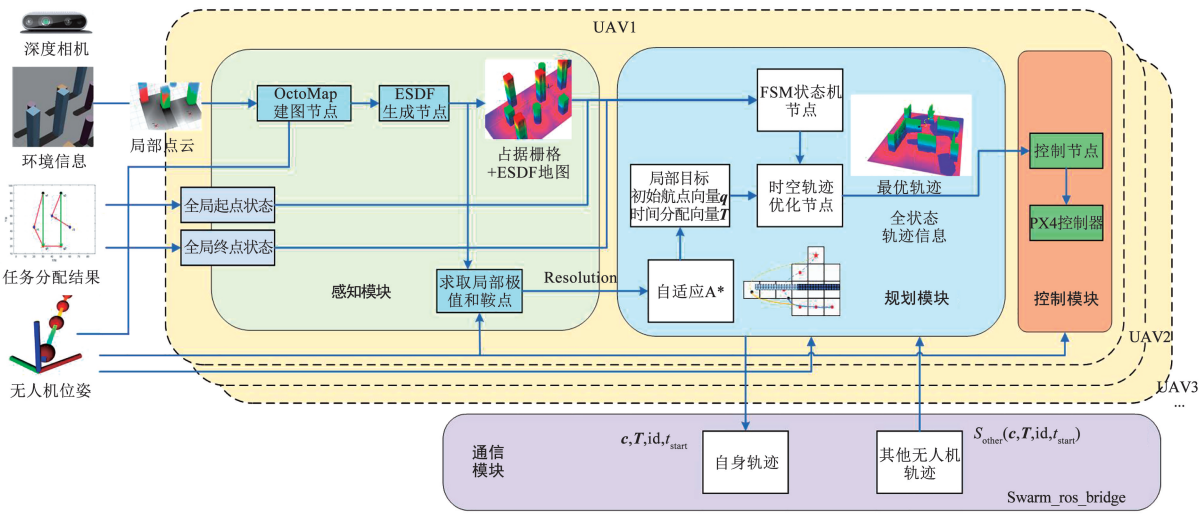


图 1 无人机集群在线规划控制框架

Fig. 1 Online planning and control framework for unmanned aerial vehicle swarms

2.2 前端路径搜索:自适应 A* 算法

在轨迹规划的框架中,全状态轨迹一般由前端的路径搜索算法和后端的轨迹优化算法共同生成。MINCO 轨迹类的时空优化并不依赖自由空间的初始航点向量与时间分配向量 $\boldsymbol{q}$、$\boldsymbol{T}$,但高质量的局部目标和初始航点向量与时间分配向量 $\boldsymbol{q}$、$\boldsymbol{T}$ 能够帮助后端优化过程更快收敛或跳出局部最优。当环境中障碍的形状和分布较为复杂,例如存在长墙或"V、U"形的障碍时,仅依靠轨迹优化往往难以找到可行解,因此需要路径搜索提供更优的初值。

前端路径搜索的目标是在当前局部 ESDF 地图中,快速生成一条从无人机当前位置到局部目标点的无碰撞几何路径。现有的固定分辨率栅格地图上的 A* 算法存在矛盾点:高分辨率能保障路径精度但计算开销大;低分辨率提升效率但可能丢失狭窄通道信息导致规划失败。本文提出自适应 A* 算法,其核心创新在于根据环境局部结构复杂程度动态调整搜索分辨率:在开阔区域使用较低分辨率以提升效率;在狭窄区域自动切换至较高分辨率以保证精度(图 2)。

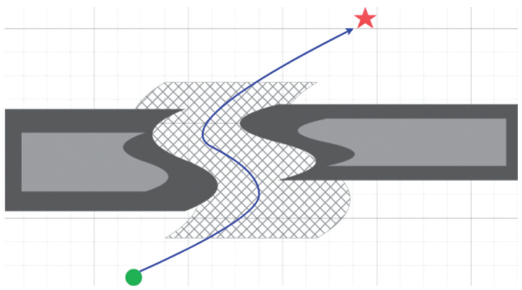


图 2 自适应 A* 算法在经过狭窄区域时的规划示意

Fig. 2 Schematic diagram of adaptive A* algorithm during planning through a narrow area

实现分辨率自适应的关键在于如何量化评估一个区域的"结构复杂程度"。利用 ESDF 地图的内在特性,本文设计了一种轻量化的评估方法。ESDF 地图在计算机中实际以矩阵形式进行存储,因此相应 $\boldsymbol{\Phi}$ 函数是离散化后的结果。ESDF 的局部极大值和鞍点在局部的结构化环境中能有效表征狭窄通道中心的位置。如图 3 所示,对于某个位置,若其在领域范围内是极大值点或鞍点,即满足以下 3 个条件之一:对于 $\forall \tilde{x}_c \in x_c \pm n\cdot\Delta x$ 和 $\forall \tilde{y}_c \in y_c \pm n\cdot\Delta y$,即

$$\begin{cases} \boldsymbol{\Phi}(x_c,y_c)\geqslant\boldsymbol{\Phi}(\tilde{x}_c,y_c),\boldsymbol{\Phi}(x_c,y_c)\geqslant\boldsymbol{\Phi}(x_c,\tilde{y}_c) \\ \boldsymbol{\Phi}(x_c,y_c)\geqslant\boldsymbol{\Phi}(\tilde{x}_c,y_c),\boldsymbol{\Phi}(x_c,y_c)\leqslant\boldsymbol{\Phi}(x_c,\tilde{y}_c) \\ \boldsymbol{\Phi}(x_c,y_c)\geqslant\boldsymbol{\Phi}(x_c,\tilde{y}_c),\boldsymbol{\Phi}(x_c,y_c)\leqslant\boldsymbol{\Phi}(\tilde{x}_c,y_c) \end{cases} \tag{5}$$

式中:Δx 为 ESDF 地图的分辨率,一般为 0.1;n 为领域的计算范围,可根据无人机大小确定。若 (x_c, y_c) 满足上述条件其中之一,则其可以描述狭窄通道的"中心"位置。经过验证,对于平直,倾斜障碍的对称是满足的,即使对于非对称的情况,满足该条件的点也能相对合适地描述障碍附近更安全的点作为"中心"。把满足该条件的点集合成 S_c,并根据其求解最优栅格。

在上述结论支持下,每次重规划时,栅格大小以及对栅格占据情况的判断由以下公式确定为

$$\begin{aligned} \mathrm{res} &= \min(\boldsymbol{\Phi}(x,y)\mid(x,y)\in S_c, D_{\mathrm{UAV}}) \\ \mathrm{safe_dis} &= \mathrm{res}/2 \end{aligned} \tag{6}$$

式中:res 为分辨率,即栅格大小;D_{UAV} 为无人机直径;safe_dis 为判定该栅格是否为障碍物的安全距

离,其通过 ESDF 函数查询,只有当栅格中心到最近障碍物的 ESDF 值大于其自身尺寸的 1/2 时,该栅格才被认为是可通行的。如图 3 所示给出了在当前栅格下最保守的情况,若栅格大于设定值,则在极限条件下无法保证精度,从而导致前端搜索错失最优路径。

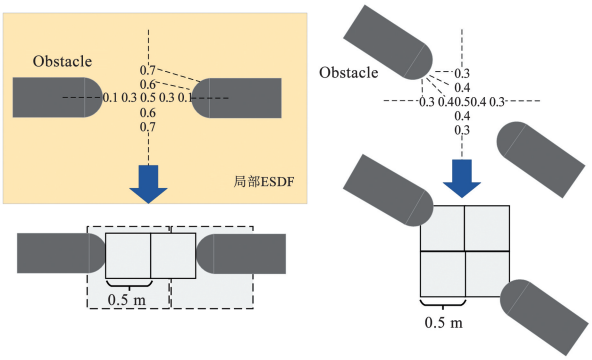


图 3　狭窄处局部 ESDF 值和栅格大小分析

Fig. 3　Analysis of local ESDF values and grid sizes in narrow areas

本文每次只考虑无人机当前自身位置附近的区域,截取部分 ESDF 地图,后续的极大值和鞍点的搜索仅在子区域内。但为了保证算法的完备性,需要记录历史最小的栅格尺寸,在无解时进行二次搜索。

在确定栅格尺寸后,在每次通过 A* 算法搜索初始路径时,还需进行坐标转换。一般情况下,A* 算法是建立在栅格化后的地图上的,但轨迹优化的决策变量却是连续的。因此,对于连续坐标(x,y)和其在 A* 算法栅格地图中对应的索引坐标$(x_{\text{idx}},y_{\text{idx}})$,转换关系如下:

$$\begin{cases} x_{\text{idx}} = \dfrac{x - x_{\text{mo}}}{\text{res}} \\ y_{\text{idx}} = \dfrac{y - y_{\text{mo}}}{\text{res}} \end{cases} \tag{7}$$

式中,$(x_{\text{mo}},y_{\text{mo}})$为占据栅格地图和 ESDF 地图的原点,其在地图扩张时会发生变化,由建图算法确定。

在完成上述关键步骤后,即可以在重规划时使用自适应 A* 算法求解出前端路径,该算法在获得全局终点、当前位置、感知半径,以及 ESDF 地图的输入条件下安装流程输出全局初始路径,并在初始路径上选择局部目标点和初始航点向量、时间分配向量$(\boldsymbol{q}、\boldsymbol{T})$,为后续的时空轨迹优化提供较高质量的初值。同时为增强系统鲁棒性,在第 1 次搜索无解的情况下将直接使用历史记录的最小栅格,避免系统陷入死循环。自适应 A* 算法在轨迹规划中的算法 1 流程如下。

算法 1:自适应 A* 前端路径生成

输入:p_{curr}(当前位置),p_{goal}(全局目标),R_{sensor}(感知半径),M_{ESDF}(ESDF 地图),D_{UAV}(无人机安全距离)

持续变量:$\text{res}_{\text{hist_min}}$(历史最小分辨率)

输出:$p_{\text{local_goal}},\boldsymbol{q},\boldsymbol{T}$(局部目标、初始路径、初始时间) or failure

// 步骤 1:动态分辨率合成(dynamic resolution synthesis)

$M_{\text{local}} \leftarrow \text{CropMap}(M_{\text{ESDF}}, p_{\text{curr}}, R_{\text{sensor}})$ // 根据当前位置和感知半径裁剪地图

$S_c \leftarrow$ Find Critical Points(M_{local})　　//在局部地图中寻找关键点

$\text{res}_{\text{optimal}} \leftarrow D_{\text{UAV}}$　　// 初始化最优分辨率为无人机安全距离

if S_c 非空 **then**

　　$\text{min_ESDF} \leftarrow \min\{\Phi(p) \text{ for all } p \in S_c\}$//获取所有关键点的最小 ESDF 值

　　$\text{res}_{\text{optimal}} \leftarrow \min(\text{min_ESDF}, D_{\text{UAV}})$　　//根据最小 ESDF 值调整最优分辨率

end if

$\text{res}_{\text{hist_min}} \leftarrow \min(\text{res}_{\text{hist_min}}, \text{res}_{\text{optimal}})$　　//更新历史最小分辨率

// 步骤 2:双阶段 A 搜索(two-phase A search)

//阶段 1:使用最优局部分辨率执行主搜索

Path $\leftarrow$ a star search$(p_{\text{curr}}, p_{\text{goal}}, \text{res}_{\text{optimal}}, M_{\text{ESDF}})$

//阶段 2:若主搜索失败,则使用历史最小分辨率执行回退搜索

if Path 未找到 **then**

　　Path $\leftarrow$ a star search$(p_{\text{curr}}, p_{\text{goal}}, \text{res}_{\text{optimal}}, M_{\text{ESDF}})$

end if

// 步骤 3:生成优化器初始值(generation of initial values for optimizer)

if Path 未找到 **then**

　　return failure　　// 如果两次搜索均失败,则返回失败

end if

$p_{\text{local_goal}} \leftarrow$ select local target(Path, p_{curr}, R_{sensor})　　//从路径中选择一个局部目标点

$(\boldsymbol{q},\boldsymbol{T}) \leftarrow$ sample waypoints and time(Path, p_{curr}, $p_{\text{local_goal}}$)// 从路径中采样路标点和时间

return $(p_{\text{local_goal}},\boldsymbol{q},\boldsymbol{T})$

2.3　后端时空轨迹优化

前端自适应 A* 算法提供了一条几何上可行的路径,但它是由离散航路点构成的,并未考虑时间、无人机动力学,以及轨迹的平滑性。因此,后端时空轨迹优化的任务是将这条路径转化为一条物理上可执行、平滑、安全且高效的时空轨迹。时空优化逻辑

结构如图 4 所示。

根据无人机轨迹规划的需求设计具体代价函数,无人机轨迹的基本要求是光滑、满足自身动力学限制、不碰撞环境障碍,应有尽可能小的控制代价和尽可能短的时间,并且支持用户定义的各种约束。在本文的无人机集群背景下,还应当保证不和其他无人机相碰撞。下面结合实例化轨迹优化问题为

$$\min_{\boldsymbol{c},\boldsymbol{T}} J(\boldsymbol{c},\boldsymbol{T}) = [J_{\mathrm{e}},J_{\mathrm{t}},J_{\mathrm{d}},J_{\mathrm{o}},J_{\mathrm{r}}]\cdot\boldsymbol{\lambda}_x \tag{8}$$

式中:J_x 为轨迹上的不同代价函数,$x=\{\mathrm{e},\mathrm{t},\mathrm{d},\mathrm{o},\mathrm{r}\}$ 为对应物理含义下标(其中每个下标分别表示轨迹的控制代价(e)、时间代价(t)、动力学惩罚代价(d)、障碍物碰撞惩罚代价(o),以及集群避碰惩罚代价(r));$\boldsymbol{\lambda}_x$ 为不同代价之间的权重向量。各维度上的代价函数 J_x 及其梯度均可分解到每个轨迹分段上。整条轨迹的总梯度 $\partial J/\partial \boldsymbol{c}$ 和 $\partial J/\partial \boldsymbol{T}$ 可以通过将所有分段的对应梯度进行拼接得到。

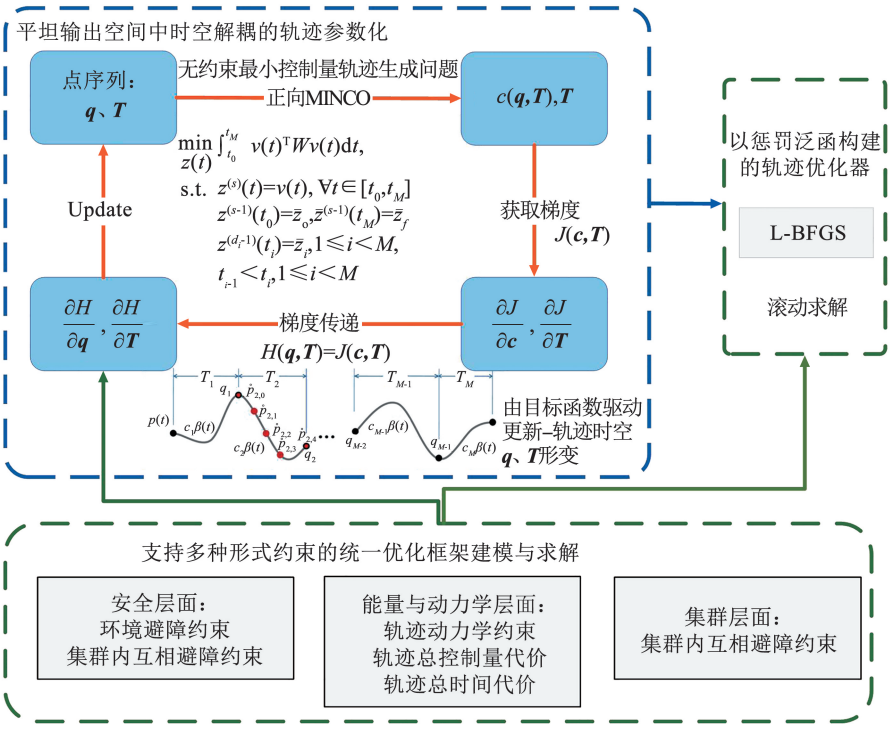


图 4 无人机轨迹参数化与时空优化逻辑结构图

Fig. 4 Logical structure diagram of unmanned aerial vehicle trajectory parameterization and spatiotemporal joint optimization

根据上文定义逐一介绍各项代价函数并求其梯度,以描述从 $J(\boldsymbol{c},\boldsymbol{T})$ 和 $\boldsymbol{c}$、$\boldsymbol{T}$ 的直接联系,为建立 ξ_{MINCO} 到 $\boldsymbol{q}$、$\boldsymbol{T}$ 的完整映射提供基础。

1)控制代价 J_{e}。控制代价的形式一般建模为该轨迹段各时刻的 Jerk 积分。其中,第 i 段轨迹的控制代价 J_{e}^i 及其梯度可分别表示为:

$$J_{\mathrm{e}}^i = \int_0^{T_i} \|p_i^{(3)}(t)\|^2 \mathrm{d}t \tag{9}$$

$$\begin{cases} \dfrac{\partial J_{\mathrm{e}}^i}{\partial c_i} = 2\left(\displaystyle\int_0^{T_i} \beta^{(3)}(t)\beta^{(3)}(t)^{\mathrm{T}}\mathrm{d}t\right)c_i \\ \dfrac{\partial J_{\mathrm{e}}^i}{\partial T_i} = \mathrm{tr}\left(c_i^{\mathrm{T}}\beta^{(3)}(T_i)\beta^{(3)}(T_i)^{\mathrm{T}}c_i\right) \end{cases} \tag{10}$$

式中,tr 为矩阵的轨迹。

2)时间代价 J_{t}。为缩短整条轨迹的用时,线性累加轨迹各段的时间作为整条轨迹的时间代价分别为:

$$J_{\mathrm{t}} = \sum_{i=1}^{M} T_i \tag{11}$$

$$\begin{cases} \dfrac{\partial J_{\mathrm{t}}}{\partial \boldsymbol{c}} = 0_{m\times n} \\ \dfrac{\partial J_{\mathrm{t}}}{\partial \boldsymbol{T}} = 1_{m\times n} \end{cases} \tag{12}$$

3)动力学惩罚代价 J_{d}。此处以速度可行性为例进行分析,加速度以及更高阶的动力学约束原理相同。设轨迹上允许的最大速度为 v_{m},轨迹上 t 时刻速度为 $\dot{p}(t)$ 可定义为:

$$\psi_v = \dot{p}(t)^2 - v_{\mathrm{m}}^2 \tag{13}$$

$$\begin{cases} \dfrac{\partial \psi_v}{\partial c_i} = 2\dot{\beta}(t)\dot{p}(t)^{\mathrm{T}} \\ \dfrac{\partial \psi_v}{\partial t} = 2\ddot{\beta}(t)^{\mathrm{T}}c_i\dot{p}(t)^{\mathrm{T}} \end{cases} \tag{14}$$

4)障碍物碰撞惩罚代价 J_{o}。评估障碍物碰撞

惩罚代价需要引入一种模型来表征环境信息,本文使用 ESDF 地图进行描述。对于地图上每一点 $p_{i,j}$,欧几里得符号距离场提供了这一点到最近障碍物的距离 $d(p_{i,j})$,以及这一距离的梯度信息 $\nabla d(p_{i,j})$,可用于高效的威胁评估。定义如下:

$$\psi_o(\hat{p}_{i,j})=\begin{cases} d_{\text{thr}}-d(\hat{p}_{i,j}), & d(\hat{p}_{i,j})<d_{\text{thr}} \\ 0, & d(\hat{p}_{i,j})\geqslant d_{\text{thr}} \end{cases} \tag{15}$$

其中,当 $\psi_o(\hat{p}_{i,j})$ 非零时,有

$$\begin{cases} \dfrac{\partial \psi_o}{\partial c_i}=-\beta(t)\nabla d^{\mathrm{T}} \\ \dfrac{\partial \psi_o}{\partial t}=-\nabla d^{\mathrm{T}}\dot{p}(t) \end{cases}$$

5)集群避碰惩罚代价 J_r。本文采用分布式的集群规划框架,即一个无人机接受其他无人机的轨迹作为约束,自主生成轨迹。考虑由多个无人机组成的集群,以其中两架无人机为例(图 5),当无人机 u 的轨迹 $p_u(t)$ 与其他无人机在同一全局时间戳上的所有轨迹保持大于安全阈值的距离时,可以保证群体避碰。然而,在优化中本文无人机使用相对时间 $t=\mathrm{j}T_i/\kappa i$,而其他无人机的轨迹采用绝对时间 $\tau=T_1+\cdots+T_{i-1}+\mathrm{j}T_i/\kappa i$(在其当次轨迹上),本质上就是各无人机时间戳在当前无人机规划过程中的对齐。在全局时间戳 τ 的尺度下惩罚与其他无人机轨迹距离小于安全阈值的约束点。因此,不同于上述代价函数,J_r 需要考虑之前的时间 $T_k(1\leqslant k\leqslant i)$,定义群体互避的代价函数为

$$J_r=\sum_{\Phi}\frac{T_i}{\kappa_i}\sum_{j=0}^{\kappa_i}\bar{\omega}_j\max\{\psi_r(p(t),\tau),0\}^3 \tag{16}$$

集群避碰惩罚代价 J_r 的梯度计算需要注意时间戳的对齐。各个无人机在自己的局部时间坐标系 t 时刻中规划,而避碰则发生在全局时间坐标系 τ 中,因此梯度 $\partial J_r/\partial T_k$ 通过链式法则,同时考虑 T_k 对局部轨迹形状($\partial t/\partial T_k$)和全局时间对齐($\partial\tau/\partial T_k$)。

对于具体的集群避碰惩罚代价,式(16)中 Φ 表示其他无人机的集合,$\psi_r(p(t),\tau)$ 表示当前计算的无人机和其他无人机在相同时间戳下的代价计算方式。$\psi_r(p(t),\tau)$ 的定义如下:

$$\begin{aligned} &\psi_r(p(t),\tau)=D_r^2-d\,(p(t),p_\phi(\tau))^2, \\ &d(p(t),p_\phi(\tau))=\|p(t)-p_\phi(\tau)\| \end{aligned} \tag{17}$$

式中:D_r 为机间避碰安全阈值,p_ϕ 为集合 Φ 中的具体的一个个体。上文已计算 J_r 对 c_i 和 T_k 的梯度的基本形式,针对具体的 ψ_r,当 $D_r^2\geqslant d(p(t),p_\phi(\tau))^2$ 时,有

$$\begin{cases} \dfrac{\partial \psi_r}{\partial c_i}=-2\beta(t)(p(t)-p_\phi(\tau))^{\mathrm{T}} \\ \dfrac{\partial \psi_r}{\partial t}=-2\,(p(t)-p_\phi(\tau))^{\mathrm{T}}\dot{p}(t) \\ \dfrac{\partial \psi_r}{\partial \tau}=2\,(p(t)-p_\phi(\tau))^{\mathrm{T}}\dot{p}_\phi(t) \end{cases} \tag{18}$$

至此,包含控制代价、时间代价、动力学惩罚代价、障碍物碰撞惩罚代价,以及集群避碰惩罚代价的可微代价函数构建完成。

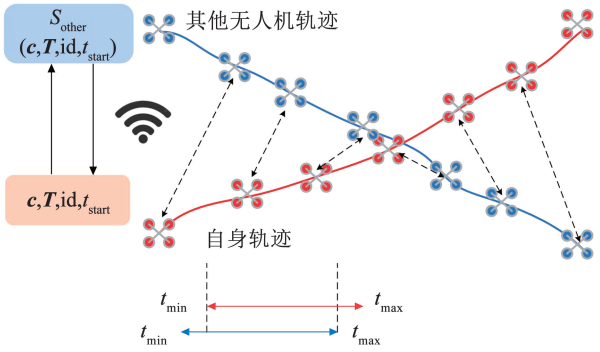


图 5　两机机间避碰示意

Fig. 5　Schematic diagram of inter-collision avoidance between two unmanned aerial vehicles

为保证时间分配 T 在优化过程中始终为正,本文引入无约束的代理变量 $\boldsymbol{\tau}=(\tau_1,\cdots,\tau_M)^{\mathrm{T}}$,并通过 Sigmoid 映射函数建立 $\tau_i\in(-\infty,+\infty)$ 与 $T_i\in(T_{\min},T_{\max})$ 的一一对应关系。优化器直接对($\boldsymbol{q}$, $\boldsymbol{\tau}$)进行优化。所需梯度 $\partial\boldsymbol{H}/\partial\boldsymbol{\tau}$ 可通过链式法则由已知的 $\partial\boldsymbol{H}/\partial\boldsymbol{T}$ 计算得到[24]。结合此结果可以构建梯度传递关系,使用基于梯度的优化器 L-BFGS[23],即可求出式(8)下最优的 $\boldsymbol{q}$ 与 $\boldsymbol{T}$,即完成轨迹的优化。

3　仿真实验

3.1　实验设置

3.1.1　仿真平台

为验证 AAM 规划器在复杂环境下的性能,本文基于机器人操作系统(robot operating system, ROS)框架搭建了支持多无人机集群协同的仿真环境,其核心由 Gazebo 场景引擎、PX4 飞控软件在环仿真(software-in-the-loop, SITL)和轻量化集群同步模拟(swarm-sync-sim, SSS)平台构成,并通过 Rviz 实现可视化。平台架构如图 6 所示,实验均在 SSS 平台上执行。

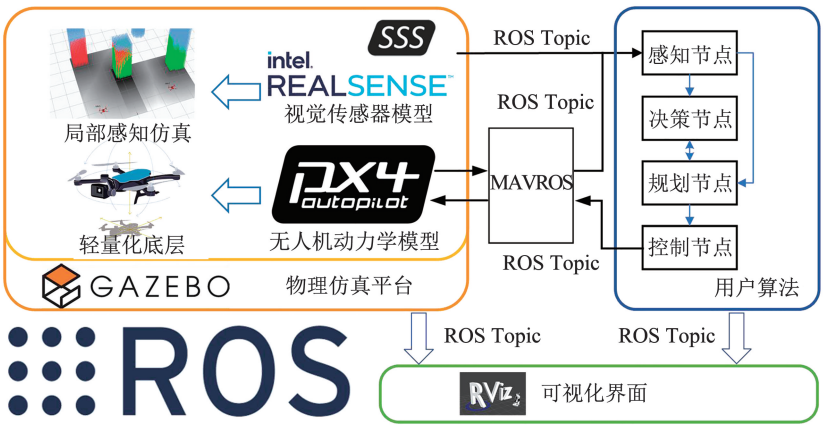
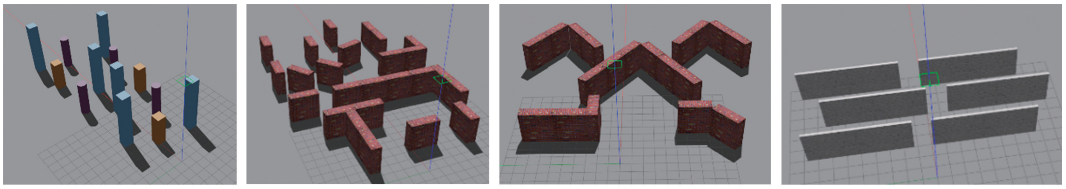


图 6　仿真环境软件架构

Fig. 6　Software architecture of simulation environment

3.1.2　仿真测试场景

为评估算法在不同复杂度环境下的鲁棒性与适应性,本文设计了立柱(poles)、砖墙(bricks)、V 形墙(V-shape)与窄墙道(narrow)4 种代表性仿真场景。这些场景覆盖了从稀疏到密集的障碍物分布形态,具体布局与参数详见图 7 及表 1。



(a) poles　(b) bricks　(c) V-shape　(d) narrow

图 7　4 个实际仿真场景

Fig. 7　Four practical simulation scenarios

表 1　4 个仿真场景中障碍物参数设定

Tab. 1　Parameter settings of obstacles in four simulation scenarios

场景名	障碍数量	场景范围/m	起点区域/m	终点区域/m	特征
poles	13	X [0, 30.0] Y [−10.0, +10.0]	X [0, 2.0] Y [−5.0, +5.0]	X [18.0, 20.0] Y [−5.0, +5.0]	柱形障碍
bricks	13	X [0, 30.0] Y [−10.0, +10.0]	X [0, 2.0] Y [−6.0, +6.0]	X [18.0, 20.0] Y [−6.0, +6.0]	长墙壁
V-shape	5	X [0,25.0] Y [−10.0, +10.0]	X [0, 2.0] Y [−5.0, +5.0]	X [18.0, 21.0] Y [−5.0, +5.0]	V 形墙壁
narrow	6	X [0, 20.0] Y [−10.0, +10.0]	X [0, 2.0] Y [−5.0, +5.0]	X [15.0, 17.0] Y [−5.0, +5.0]	窄缝

在实验场景中,统一设置通用飞行任务:无人机在 2 m 固定高度,从指定区域内的随机起始点出发,自主飞行至目标区域内的随机终点,并在整个飞行过程中持续进行重规划以动态避障。

为全面评估算法性能,本文在仿真实验中选取了 3 类典型的基线方法作为 AAM 规划器的对比对象。

1) Base MINCO[12]。代表仅采用 MINCO 轨迹类进行优化的纯后端方法,评估缺乏高质量前端引导时轨迹优化的性能上限和局限性。

2) LRAM(low-resolution A* + MINCO)。采用较低分辨率的 A* 算法作为前端路径搜索,代表现有工程实现中以提升效率为主的典型分层规划策略。

3) HRAM(high-resolution A* + MINCO)。使用与地图分辨率一致的高精度网格进行搜索,代表以追求路径精度为主的典型固定分辨率方法。其中

LRAM 的分辨率设置为 1.0 m,与无人机机体尺度处于同一量级,能减少搜索节点数量,体现出低分辨率搜索在效率上的优势,同时暴露其在狭窄通道中易丢失几何细节的问题;而 HRAM 的分辨率则设置为 0.1 m,与 ESDF 地图分辨率一致,可作为精细路径搜索的上限基线。评估依据以下 3 个核心指标:

①平均规划时间。指单次重规划的平均耗时,用于衡量算法的实时性。

②轨迹代价。综合轨迹长度、碰撞风险与动力学可行性的加权指标,评估轨迹质量。每次规划时,记录一个无人机状态序列 $S_{\mathrm{UAV}}=((p_1,v_1),(p_2,v_2),\cdots,(p_L,v_L))$,$L$ 为这一序列的长度。轨迹代价为

$$J_{\mathrm{traj}}=\omega_{\mathrm{l}}J_{\mathrm{length}}+\omega_{\mathrm{c}}J_{\mathrm{collision}}+\omega_{\mathrm{d}}J_{\mathrm{dynamics}} \tag{19}$$

式中,各代价的权重设置为 $\omega_{\mathrm{l}}=1$,$\omega_{\mathrm{c}}=100$,$\omega_{\mathrm{d}}=1$,主要基于不同代价项在数值量级及物理意义上的差异性,遵循安全性优先、可行性次之、优化目标再次的原则。碰撞惩罚项原始量级小,因此赋予相对较高权重,以确保轨迹在各种场景中均满足安全距离约束;动力学惩罚项原始量级相对较大,赋予较小权重即可有效抑制速度过大而不使其主导整体优化;长度代价主要用于改善轨迹质量,其量级较低,通过多次预实验调节确定。

实际飞行轨迹的长度为

$$J_{\mathrm{length}}=\sum_{i=1}^{L-1}\|p_{i+1}-p_i\|_2 \tag{20}$$

障碍物冲突代价为

$$J_{\mathrm{collision}}=\sum_{i=1}^{L}\max(d_{\mathrm{thr}}-d(p_i),0)^3 \tag{21}$$

动力学违反代价为

$$J_{\mathrm{dynamics}}=\sum_{i=1}^{L}\max(\|v_i\|_2^2-v_m^2,0)^3 \tag{22}$$

③规划成功率。指在多次实验中,无人机无碰撞到达目标点的次数与总次数之比,用以评估算法的可靠性与鲁棒性。

3.2　结果与分析

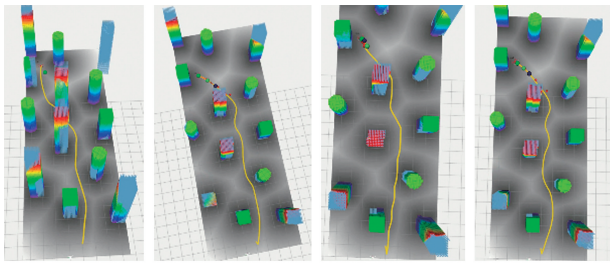
本文首先对单机规划任务进行定量与定性分析,以验证 AAM 规划器在复杂环境下的优势,随后展示其在多机协同任务中的有效性。为此,按照实验设置描述的实验方案在 poles、bricks,以及 narrow 3 个场景中设置了指标对比实验,各场景下对相应算法分别进行了 10 次仿真实验,统计规划成功率,并对规划成功的实验数据计算平均轨迹代价和平均规划时间。实验数据汇总于表 2。

表 2　4 种轨迹规划器算法仿真结果对比

Tab. 2　Comparison of simulation results of four trajectory planning algorithms

场景	规划器算法	评价指标		
		规划成功率/%	平均规划时间/s	平均轨迹代价
1(poles)	Base MINCO	100	0.086	20.9
	LRAM	100	0.053	17.4
	HRAM	90	0.124	18.3
	AAM	100	0.055	17.4
2(bricks)	Base MINCO	20	0.073	48.1
	LRAM	100	0.060	31.4
	HRAM	70	0.093	25.6
	AAM	100	0.068	25.5
3(narrow)	Base MINCO	40	0.102	17.5
	LRAM	100	0.066	30.2
	HRAM	90	0.121	17.1
	AAM	100	0.073	16.8

同时,图 8 ~ 10 给出了代表性的实际飞行轨迹,以直观对比各算法的规划效果。



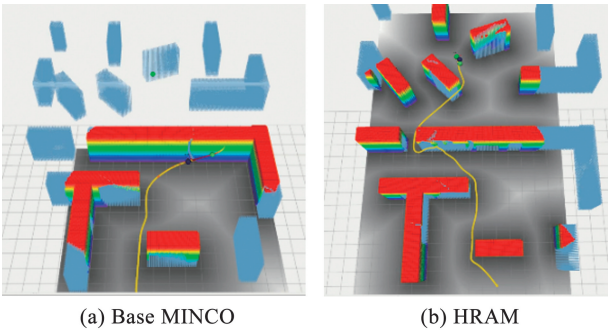
(a) Base MINCO　(b) LRAM　(c) HRAM　(d) AAM

图 8　poles 场景下 4 种规划器算法的可视化对比

Fig. 8　Visualization comparison of four planners algorithms in poles scenario

在成功率与效率方面,AAM、LRAM 规划器算法凭借前端路径引导的高成功率,优于在复杂场景中频繁失败的纯后端 Base MINCO 规划器算法。同时,AAM 规划器算法的计算效率与 LRAM 规划器算法相当,远快于因初始解不佳而收敛缓慢的 Base MINCO 规划器算法。在轨迹质量方面,AAM 规划器算法表现最佳。其自适应分辨率机制能在开阔区高效搜索,在狭窄区精细规划,最终获得类似于高分辨率 HRAM 规划器算法的平滑、安全路径。这克服了 LRAM 规划器算法因分辨率过低导致的轨迹质量不佳,以及 HRAM 规划器算法因分辨率过高而规划失败的风险。综上所述,在复杂场景中,综合考虑

规划成功率、规划时间,以及轨迹代价,AAM 规划器算法的整体性能优于其他算法,具备在各类复杂场景中求解较高质量轨迹的能力。

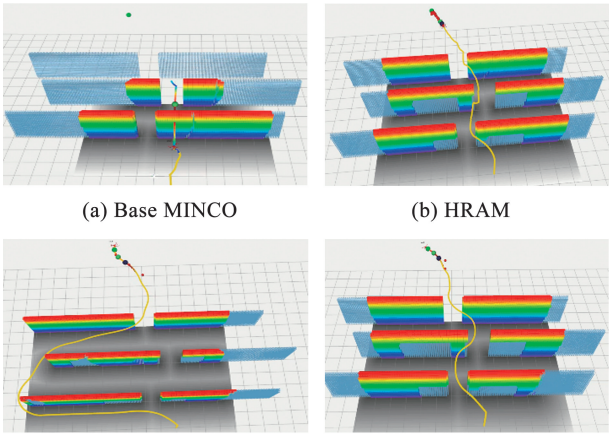


(a) Base MINCO (b) HRAM

(c) LRAM (d) AAM

图 9 bricks 场景下 4 种规划器算法的可视化对比

Fig. 9 Visualization comparison of four planners algorithms in bricks scenario



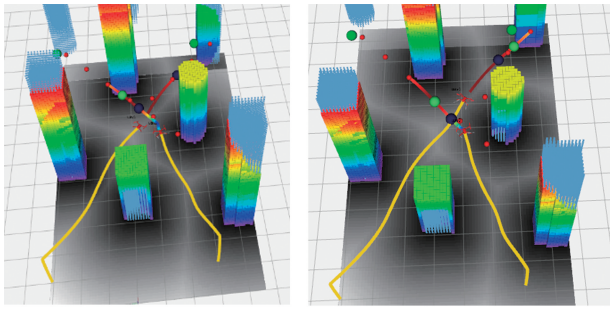
(a) Base MINCO (b) HRAM

(c) LRAM (d) AAM

图 10 narrow 场景下 4 种规划器算法的可视化对比

Fig. 10 Visualization comparison of four planners algorithms in narrow scenario

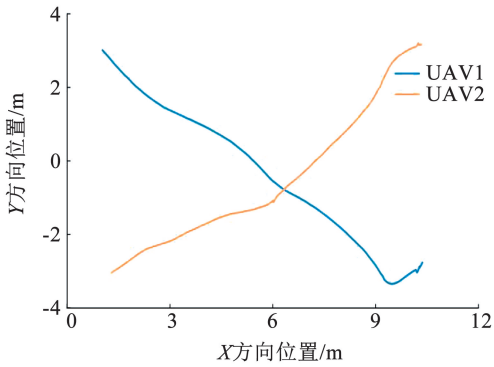
在集群规划验证中,本文模拟了 2 机航线与 4 机航线冲突场景,以测试其关键的个体间避碰能力。仿真结果如图 11 ~ 14 所示,表明无人机能在规避环境障碍的同时,完成安全的航线冲突避让,机间距离始终大于安全阈值。得益于时空优化,其避让轨迹表现为时间维度上的平滑调整,避免了大幅度的空间绕行,从而保证了高质量的协同轨迹。



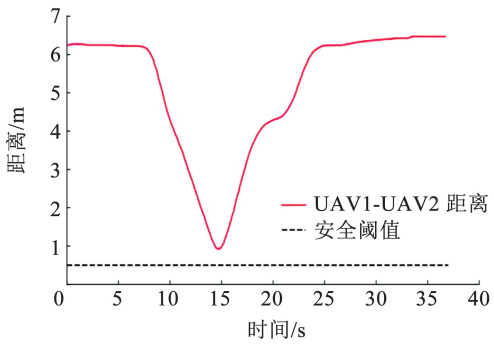
(a) 2机相遇前 (b) 2机相遇后

图 11 2 机协同规划仿真结果

Fig. 11 Simulation results of cooperative planning for two unmanned aerial vehicles



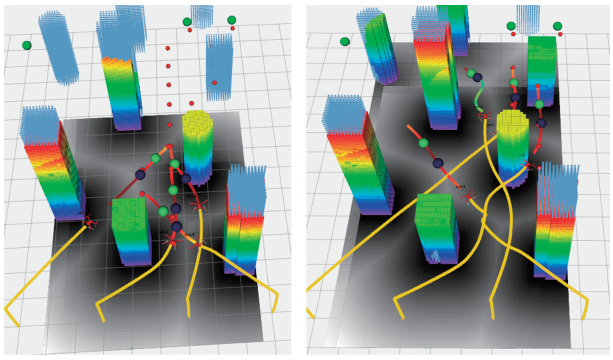
(a) 无人机轨迹



(b) 无人机之间的距离

图 12 2 机仿真过程无人机位置和相对距离记录

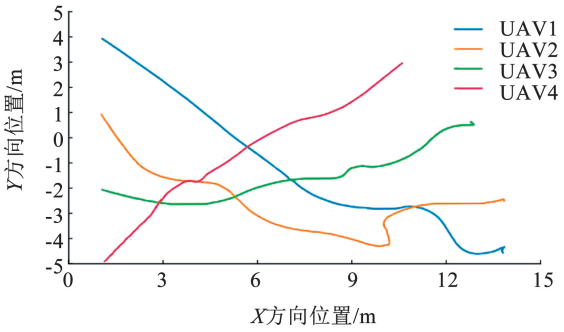
Fig. 12 Records of unmanned aerial vehicle positions and relative distances in simulation process of two unmanned aerial vehicles



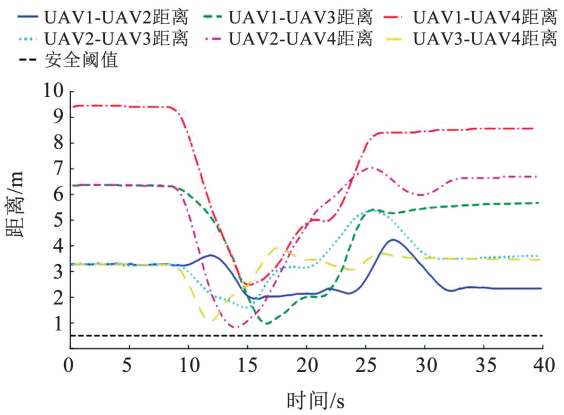
(a) 4机相遇前 (b) 4机相遇后

图 13 4 机协同规划仿真结果

Fig. 13 Simulation results of cooperative planning for four unmanned aerial vehicles



(a) 无人机轨迹



(b) 无人机之间的距离

图 14　4 机仿真过程无人机位置和相对距离记录

Fig. 14　Records of unmanned aerial vehicle positions and relative distances in simulation process of four unmanned aerial vehicles

4　结　论

1)本文设计了一种自适应 A* 算法前端路径搜索算法,通过动态调整搜索分辨率,有效解决了传统 A* 算法在效率与精度上的固有矛盾;构建了一个基于 MINCO 参数化的时空优化后端,将多个目标整合进一个统一可微的代价函数中,实现了高效求解;通过将自适应前端与统一后端高效协同,形成了一种在线规划框架。

2)仿真实验表明,该框架在规划成功率、计算效率,以及轨迹质量上均优于基线方法,其平均规划时间较基线方法减少 18.4%,轨迹总代价平均降低 15.3%,有效平衡了规划的实时性、鲁棒性与最优性。

3)未来研究将考虑在动态环境下引入障碍物运动预测模型,使其能够应对动态障碍物的协同避障问题;另外将考虑姿态动力学模型纳入规划层优化过程,以进一步提升轨迹在强机动和高动态场景下的可执行性。

参考文献

[1] Zhang Jun, Xing Jiahao. Cooperative task assignment of multi-UAV system[J]. Chinese Journal of Aeronautics, 2020, 33(11): 2825. DOI: 10.1016/j.cja.2020.02.009

[2] Floreano D, Lipson H. From individual robots to robot societies[J]. Science Robotics, 2021, 6(56): eabk2787. DOI: 10.1126/scirobotics.abk2787

[3] Chung S J, Paranjape A A, Dames P, et al. A survey on aerial swarm robotics[J]. IEEE Transactions on Robotics, 2018, 34(4): 837. DOI: 10.1109/TRO.2018.2857475

[4] Jones M, Djahel S, Welsh K. Path-planning for unmanned aerial vehicles with environment complexity considerations: a survey[J]. ACM Computing Surveys, 2023, 55(11): 1. DOI: 10.1145/3570723

[5] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100. DOI: 10.1109/TSSC.1968.300136

[6] Dolgov D, Thrun S, Montemerlo M, et al. Path planning for autonomous vehicles in unknown semi-structured environments[J]. International Journal of Robotics Research, 2010, 29(5): 485. DOI: 10.1177/0278364909359210

[7] Chen Fang, Zheng Yu, Wang Zheng, et al. RBI-RRT*: efficient sampling-based path planning for high-dimensional state space[C]// 2024 IEEE International Conference on Robotics and Automation (ICRA). Yokohama: IEEE, 2024: 8721. DOI: 10.1109/ICRA57147.2024.10610975

[8] Wang Pengyu, Tang Jiawei, Wang Lin H, et al. MINER-RRT*: a hierarchical and fast trajectory planning framework in 3D cluttered environments[J]. IEEE Transactions on Automation Science and Engineering, 2025, 22: 10973. DOI: 10.1109/TASE.2025.3531504

[9] Mellinger D, Kumar V. Minimum snap trajectory generation and control for quadrotors[C]//2011 IEEE International Conference on Robotics and Automation. Shanghai: IEEE, 2011: 2520. DOI: 10.1109/ICRA.2011.5980409

[10] 于振中, 闫继宏, 赵杰, 等. 改进人工势场法的移动机器人路径规划[J]. 哈尔滨工业大学学报, 2011, 43(1): 50
Yu Zhenzhong, Yan Jihong, Zhao Jie, et al. Mobile robot path planning based on improved artificial potential field method[J]. Journal of Harbin Institute of Technology, 2011, 43(1): 50. DOI: 10.11918/j.issn.0367-6234.2011.01.011

[11] 肖友刚, 朱铖臻, 李蔚, 等. 一种新型高阶连续的点对点运动轨迹规划算法[J]. 哈尔滨工业大学学报, 2021, 53(9): 135
Xiao Yougang, Zhu Chengzhen, Li Wei, et al. A novel high-order continuous point-to-point motion trajectory planning algorithm[J]. Journal of Harbin Institute of Technology, 2021, 53(9): 135. DOI: 10.11918/202002089

[12] Wang Zhepei, Zhou Xin, Xu Chao, et al. Geometrically constrained trajectory optimization for multicopters[J]. IEEE Transactions on Robotics, 2022, 38(5): 3259. DOI: 10.1109/TRO.2022.3160022

[13] Fan Mingyue, Xie Fangfang, Ji Tingwei, et al. Three-dimensional

trajectory optimization for quadrotor tail-sitter AAVs: traversing through given waypoints[J]. IEEE Transactions on Aerospace and Electronic Systems, 2025, 61(3): 6987. DOI: 10.1109/TAES.2025.3531845

[14] Tordesillas J, Lopez B T, Everett M, et al. FASTER: fast and safe trajectory planner for navigation in unknown environments[J]. IEEE Transactions on Robotics, 2022, 38(2): 922. DOI: 10.1109/TRO.2021.3100142

[15] Arul S H, Manocha D. DCAD: decentralized collision avoidance with dynamics constraints for agile quadrotor swarms[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 1191. DOI: 10.1109/LRA.2020.2967281

[16] Zhou Xin, Zhu Jiangchao, Zhou Hongyu, et al. EGO-swarm: a fully autonomous and decentralized quadrotor swarm system in cluttered environments[C]//2021 IEEE International Conference on Robotics and Automation(ICRA). Xi'an: IEEE, 2021: 4101. DOI: 10.1109/ICRA48506.2021.9561902

[17] Tordesillas J, How J P. MADER: trajectory planner in multiagent and dynamic environments[J]. IEEE Transactions on Robotics, 2022, 38(1): 463. DOI: 10.1109/TRO.2021. 3080235

[18] Kondo K, Figueroa R, Rached J, et al. Robust MADER: decentralized multiagent trajectory planner robust to communication delay in dynamic environments[J]. IEEE Robotics and Automation Letters, 2024, 9(2): 1476. DOI: 10.1109/LRA.2023.3342561

[19] Adajania V K, Zhou Siqi, Singh A K, et al. AMSwarmX: safe swarm coordination in complex environments via implicit non-convex decomposition of the obstacle-free space [C]//2024 IEEE International Conference on Robotics and Automation (ICRA). Yokohama: IEEE, 2024: 14555. DOI: 10.1109/ICRA57147.2024.10610428

[20] Hou Jialiang, Zhou Xin, Gan Zhongxue, et al. Enhanced decentralized autonomous aerial robot teams with group planning [J]. IEEE Robotics and Automation Letters, 2022, 7(4): 9240. DOI: 10.1109/LRA.2022.3191037

[21] Huang Zhehui, Yang Zhaojing, Krupani R, et al. Collision avoidance and navigation for a quadrotor swarm using end-to-end deep reinforcement learning[C]//2024 IEEE International Conference on Robotics and Automation (ICRA). Yokohama: IEEE, 2024: 300. DOI: 10.1109/ICRA57147.2024.10611499

[22] Quan Lun, Yin Longji, Xu Chao, et al. Distributed swarm trajectory optimization for formation flight in dense environments [C]//2022 International Conference on Robotics and Automation. (ICRA). Philadelphia: IEEE, 2022: 4979. DOI: 10.1109/ICRA46639.2022.9812050

[23] Liu D C, Nocedal J. On the limited memory BFGS method for large scale optimization[J]. Mathematical Programming, 1989,45(1): 503. DOI: 10.1007/BF01589116

[24] Fainekos G E, Kress-gazit H, Pappas G J. Temporal logic motion planning for mobile robots[C]//Proceedings of the 2005 IEEE International Conference on Robotics and Automation. Barcelona: IEEE, 2006: 2020. DOI: 10.1109/ROBOT.2005.1570410

(**编辑** 张 红)

封面图片说明

封面图片来自本期论文“分层强化学习驱动无人作战飞机空战自主决策”,是空军工程大学装备管理与无人机工程学院茹乐课题组在开展无人机智能空战决策研究时,提出的一种结合分层强化学习与两阶段训练范式的自主决策新框架。本文针对复杂高动态空战环境下传统方法决策效率低、策略鲁棒性差的问题,首先设计了一种分层决策架构,将端到端的决策空间解耦为上层战术指令学习与下层连续动作执行,以降低学习复杂度并增强可解释性。其次,提出“专家模仿—自博弈进化”两阶段训练方法,利用基于 SECA 规则架构的专家对手引导智能体获得高质量初始策略,再通过虚拟自博弈与历史最优版本持续对抗,推动策略突破专家知识局限。最后,搭建高保真空战仿真环境对方法性能进行验证。研究结果表明,相比传统 DQN 及 PPO 等基准算法,本文方法在奖励收敛速度、收敛水平,以及对抗胜率方面均显著提升,消融实验进一步验证了两阶段训练机制的有效性,有力提升了无人机在动态空战环境中的自主决策与对抗能力。

(图文提供:王文飞,茹乐,吕茂隆,莫雳。空军工程大学装备管理与无人机工程学院;无人飞行器技术全国重点实验室(西北工业大学);北京理工大学空天科学与技术学院)