

中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生
数学建模竞赛

学 校 华东师范大学

参赛队号 20102690110

队员姓名 1. 纪莉娜
2. 刘禹含
3. 虞小颖

中国研究生创新实践系列大赛

“华为杯”第十七届中国研究生

数学建模竞赛

题 目

无人机集群协同对抗

摘 要：

本文对无人机群集群对抗任务进行了分析研究，针对每个具体问题建立模型并采用合适的算法进行求解。

问题一第一小问为定性微分对策问题，在完全信息以及最优策略的基础下求解蓝方的可逃脱区域等同于求解定性微分对策的界栅问题。根据题目首先建立了地面坐标系和航迹坐标系，根据题目建立运动与位置约束条件。对无人机可行队形进行分析，并计算出近似最大覆盖面积用于确定边界。基于双机编队的队形确定了坐标系的转换关系，把微分对策看作变分问题，进而利用极大值极小值原理构建哈密顿函数解最优控制，椭圆控制集将引起奇异解，将初始值以及求解的最优控制代入运动方程倒向积分可得最优轨线，最优轨线构成界栅。由 BUP（目标集的可用部分边界）为起点所构成的界栅与 NUP 所围成的区域构成了对策空间的躲避区。

问题一第二小问在第一小问的基础上求解蓝方最优突防策略，即该微分对策问题的纳什均衡。博弈理论中的动态博弈与矩阵对策相结合的方法，选取目标规避策略中有限种机动形式的组合作为策略集，根据红蓝双方距离以及追逃时间为量化指标建立支付函数，由哈密顿函数求取动态博弈模型的纳什均衡解，得出了受初始值位置影响动态突防策略。

问题二为生存型定量微分对策问题，在问题一的基础上建立双机编队下时间最优的无人机突防模型，其中支付函数以逃脱时间为量化指标。当通道带宽取下限时，蓝方出发点（AB 中点）位于界栅上。由于问题解析求解过于复杂，将突防区域进行栅格化划分并根据条件约束进行航迹优化，运用蚁群算法进行不同初始值下的纳什均衡解，并反推出通道宽度极限值。根据计算结果分析红方出发位置对蓝方初始局部区域哈密顿函数的影响，得到红方最优的发射点分布。仿真分析不同通道宽度下蓝方出发点的哈密顿函数值，得到通道宽度取值为 92.2Km。

问题三，红方运载机可以发射两个波次的无人机，即基于微分对策理论研究二维空间内多对一无人机突防模型，求取突防模型中拦截过程的发射策略（数量分配、路径规划）。分为三步求解：基于博弈论的飞行器微分制导律方法建立多对一拦截模型，运用零脱靶量建立支付函数，此处，以飞行器拦截目标为背景，借鉴滚动时域控制的思想，构建动态博弈模型；第二步运用快速随机数算法进行动态最优策略的搜索，规划追捕过程的最优路径；第三步为了简化动态搜索过程、提高搜索精度，采用 Dubins 方法对栅格化路径中转弯最优角度进行判断，RRT 算法评价对路径选择及计算效率的影响。求解可得最优拦截策略（表 6-2）以及任意情况成功拦截的通道带宽上限值为 108.8Km。

问题三第二问通道带宽上限是否存在的问题，即生存型微分对策的鞍点存在性问题。

根据微分对策最优性条件推导该问题对策鞍点满足的必要条件，哈密顿函数中双方极值存在，并利用差分进化算法进行求解。

问题四为多对多无人机对抗最优策略问题。分四步进行模型的求解：第一步建立无人机集群的协同对抗模型；第二步建立突防区的空战态势优势函数，根据红蓝双方的态势信息如几何位置计算对抗区域边界，对空战态势进行评估；第三步运用分布式一致性拍卖算法（CBAA）对目标进行分配决策，对整个集群的目标函数进行优化；第四步基于 SAC-OD 规则进行集群运动决策，基于集群动态拓扑交互机制，建立有限感知条件下的集群运动模型，使集群涌现出宏观的作战行为。对建立的模型进行仿真检验（表 8-1、8-2），结果显示模型良好。

最后对以上四个问题的模型和算法进行了评价，指出了模型的优点，如问题二、三针对微分对策问题难以求得准确解析解时，本问通过运用合适的算法对边界值进行正向计算，求得结果；同时也指出了如在队形变化时边界处理有误差等不足。

综上，无人机在未来空战中可发挥至关重要的作用，本文在无人机集群对抗与突防策略制定上有很强的现实指导意义。

关键词：微分对策、界栅、协同对抗策略、Dubins 方法、拍卖算法

目 录

1 问题重述	5
1.1 问题背景	5
1.2.1 问题一：逃逸区与最优突防策略求解	5
1.2.2 问题二：通道带宽下限与最短突防策略求解	6
1.2.3 问题三：拦截策略制定与通道带宽上限计算	6
1.2.4 问题四：无人机群协同对抗博弈策略制定	6
2. 模型假设	7
3. 符号说明	7
4. 问题一	8
4.1 问题分析	8
4.2 模型准备	8
4.2.1 坐标系建立及运动约束	8
4.2.2 微分对策问题	10
4.3 二对一无人机突防界栅模型	11
4.3.1 基于队形的无人机集群覆盖区域	11
4.3.2 双机编队下的坐标系转换	13
4.3.3 界栅构造	15
4.4 模型求解与分析（第一小问）	16
4.5 最优突防策略模型	19
4.6 模型求解与分析（第二小问）	21
5. 问题二	21
5.1 问题分析	21
5.2 双机编队下时间最优的无人机突防模型	22
5.3 突防策略分析	23
5.3.1 突防通道区域栅格离散化处理	23
5.3.2 UAV 突防航迹优化规划	24
5.4 模型求解	25
5.5 结果分析	27
6. 问题三	29
6.1 问题分析	29
6.2 模型建立	29
6.2.1 微分对策制导方法	30
6.2.2 多对一拦截模型	31
6.2.3 Dubins 与 RRT 结合的突防算法	32
6.3 模型求解分析（第一小问）	38
6.4 鞍点存在性判断与计算（第二小问）	39
6.4.1 鞍点的必要条件	39
6.4.2 差分进化算法求解	40
6.4.3 通道带宽上限求解分析	41
7. 问题四	41
7.1 问题分析	41

7.2 无人机集群协同对抗模型	42
7.2.1 协同对抗任务规划流程	42
7.2.2 空战态势评估	43
7.2.3 目标分配优化模型决策	44
7.2.4 基于扩展 ICBA 算法的目标分配模型求解	45
7.2.5 基于 SAC-OD 规则的集群运动决策	48
7.3 模型检验	50
7.4 结果分析	54
8. 模型的评价、改进与推广	54
8.1 模型优点	54
8.1.1 原理的完备性	54
8.1.1 求解的可行性	55
8.2 不足和改进	55
参考文献	55
附 录	57

1 问题重述

1.1 问题背景

新一代人工智能技术和自主技术快速走向战场，无人机集群作战作为智能作战的重要形式，通过多架无人机协同侦察、协同探测、协同跟踪、协同攻击、协同拦截等，共同完成较复杂的作战任务。现考虑红、蓝双方的无人机集群在平面区域内的协同对抗问题。

蓝方作为进攻方，希望突破红方无人机的拦截，成功抵达目的地遂行军事行动；红方则希望在给定的区域内完成对蓝方无人机的拦截，阻止蓝方的突防。

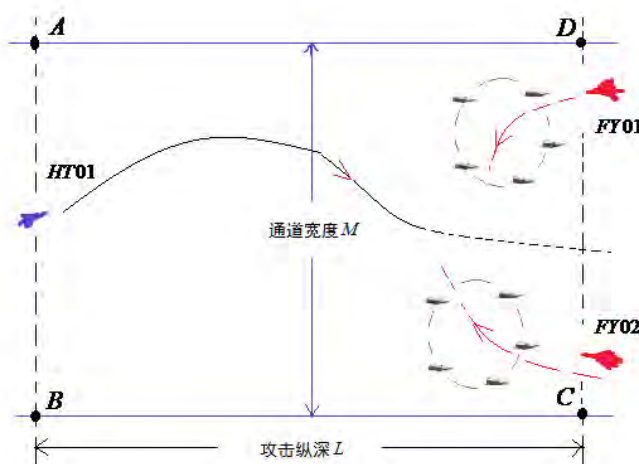


图 1-1 红方两个无人机集群拦截一架蓝方突防无人机示意图

对抗区域约定为图 1-1 所示的矩形区域，攻击纵深即 之间的距离为，蓝方无人机的飞行轨迹不能越过、两边，即考虑的是攻击通道（突防走廊）带宽有一个限定约束的情形，通道带宽即 之间的距离记为。蓝方无人机的速度为，最小转弯半径为；红方无人机的速度为，最小转弯半径为；红蓝双方无人机的速度保持不变，运动的方向可根据机动策略的需要随时改变，但受转弯半径的限制。

当蓝方突防无人机与红方至少 2 架无人机的距离均小于 $R = 300m$ 时，就认为红方成功拦截了蓝方突防无人机。同时，根据任务要求，蓝方无人机需在 360s 内越过边界 CD，否则视为突防失败。红方无人机集群可以根据实际需要进行编队飞行，其中的一种队形 5 架无人机位置近似分布在一个圆周上，任何相邻两架无人机的间距相同。为控制、通信以及相互避撞的需要，要求红方任何两架无人机的间距需大于 30m，每一架无人机与本集群中至少两架无人机的距离不超过 200m。

红方运载机与所属无人机集群中至少一架无人机的距离不超过 10km，与任何一架无人机的距离需大于 100m，同时为安全需要，与蓝方的突防无人机的距离需大于 5km。红方运载机的速度为 $V_{\text{红}} = 300m/s$ ，转弯半径不小于 1000m。红方运载机和无人机的飞行轨迹不受 ABCD 边界的限制。

1.2.1 问题一：逃逸区与最优突防策略求解

如图 1-2 所示，对抗伊始红方 2 个无人机集群的圆周中心分别位于 $G1$ 和 $G2$ ，圆周半径为 100m，其中 $DG1 = 20km$ ， $G1G2 = 30km$ ， $CG2 = 20km$ 。试建模分析蓝方无人机处于矩形区域 ABCD 内哪些位置时，无论红方无人机采用什么样的追击策略，蓝方无人机总能采用合适的策略以躲避红方的拦截，实现成功突防；讨论蓝方无人机相应的最优突防策略。

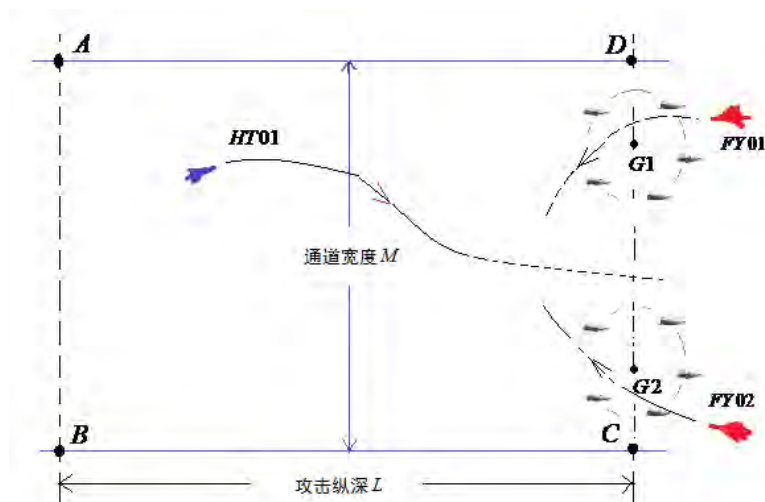


图 1-2 红方无人机集群初始中心位置示意图

1.2.2 问题二：通道带宽下限与最短突防策略求解

对抗伊始蓝方突防无人机位于边界 AB 的中心点，红方 2 个无人机集群的圆周中心分别位于 $G1$ 和 $G2$ ，圆周半径为 $100m$ ，其中 $G1$ 和 $G2$ 位于边界 CD 上，具体位置根据需要确定。试建模分析是否存在一个通道带宽 M 的下限 M_{\min} ，当实际通道带宽 M 比 M_{\min} 大时，蓝方无人机一定能突破红方无人机集群的拦截；给出此种情形下蓝方无人机时间最短的突防策略。

1.2.3 问题三：拦截策略制定与通道带宽上限计算

红方每架运载机可分两个波次共发射 10 架无人机，组成两个无人机集群遂行拦截任务，每个无人机集群的无人机数量不少于 3 架。每一波次发射时无人机集群初始队形如图 2 所示的圆周构型，运载机与圆周中心的距离为 $2km$ ，随后无人机集群的队形可根据需要调整，但要求满足相应的间距约束。如图 5 所示，对抗伊始，蓝方无人机位于边界 AB 的中心，通道带宽 $M = 70km$ ；红方两架运载机分别位于边界 CD 上 $G1$ 点和 $G2$ 点，并开始发射第一波次的无人机集群，运载机和无人机集群中心具体位置根据需要确定。运载机第二波次发射无人机集群时，必须保证运载机与第一波次发射的无人机集群满足间距上的约束。讨论红方两架运载机两个波次发射的无人机数量、每架运载机第二波次发射的时刻和位置以及第二波次发射的无人机集群的中心位置，以实现最优的拦截效果；进一步具体建模分析是否存在一个通道带宽 M 的上限 M_{\max} ，当实际通道带宽 M 小于 M_{\max} 时，无论蓝方无人机采用什么样的突防策略，红方无人机集群均存在相应的拦截策略，在区域 $ABCD$ 内成功阻止蓝方无人机的突防。

1.2.4 问题四：无人机群协同对抗博弈策略制定

通道带宽 $M = 100km$ ，蓝方 3 架突防无人机组成突防集群从矩形边界 AB 一侧开始突防（任 2 架突防无人机的间距需大于 $30m$ ），红方 5 架运载机各携带 10 架无人机，从边界 CD 一侧同时开始遂行协同拦截任务。红方每架运载机分两个波次发射无人机，分别组成两个无人机集群，每个集群的无人机数量不少于 3 架；每架运载机第一波次发射无人机的时刻为初始对抗时刻，与所属无人机集群几何构型圆周中心的距离为 $2km$ 。红方运载机初始位置、红方运载机发射的第一个波次的无人机集群中心位置、红方运载机发射第二波次无人机集群的时刻和位置、第二波次发射的无人机集群中心位置、两个波次无人机数量以及蓝方突防无人机初始位置根据需要确定。蓝方希望尽可能多的无人机突防成功，红方则希望成功拦截尽可能多的蓝方无人机。试讨论红方最优拦截策略和蓝方最优突防策略。

2. 模型假设

- (1) 只讨论平面上的红蓝双方对抗问题，即红蓝双方的无人机均在同一高度上机动飞行；
- (2) 运载机可以瞬间布设好无人机集群，忽略布设时间
- (3) 无人机和运载机在空中飞行时均视为质点，不考虑其体积对对战的影响
- (4) 对抗双方的无人机能顾完全根据指挥执行战术决策，将无人机集群理解为完全理性的局中人；
- (5) 战场态势和对方的完全理性均为共识，即双方彼此了解对方实力及可能选取的所有策略集合，且彼此知道对方是完全理性的；
- (6) 双方无人机均能够准确探测对方无人机和己方无人机的实时位置，完全信息博弈；
- (7) 双方能够正确识别敌我战机，且能够通过对实时态势的感知以及与邻居友机的交互进行在线自主决策；
- (8) 无人机运动轨迹位置约束，拦截与突防成功判断条件遵照题目；
- (9) 其他假设在问题中具体说明。

3. 符号说明

符号	符号说明
N	UAV 数量
$ABCD$	对抗的矩形区域
$FY0N \times \times$	运载机 $FY0N$ 发射的无人机编号
$OX_g Y_g$	地面坐标系
$OX_k Y_k$	航迹坐标系
$(x_{F1}, y_{F1}),$ (x_{F2}, y_{F2})	红方无人机集群中心位置坐标
L_A, L_B	红方无人机集群最大攻击距离
(x_H, y_H)	蓝方无人机位置坐标
χ_C	蓝方无人机和无人机集群坐标系的横轴方向 夹角
$H(x, u, v, \lambda)$	局中人 P 和 E 在对弈策略 v 和 u 下的 Hamilton 函数
Pr_i	i 处概率代价
$A=\{i, j\}$	UAV 航迹的集合
D_{ij}	i, j 双机距离
t_{ij}	UAV 的沿 D_{ij} 路径的飞行时间

符号	符号说明
a_{ij}	我方 UAV 对敌机的方位角
Sa_{ij}	角度优势指数
Sr_{ij}	距离优势指数
p_j^Y	Y 方第 j 架 UAV 的生存概率
v_j^Y	Y 方第 j 架 UAV 的生存价值

4. 问题一

4.1 问题分析

问题一第一小问要求分析蓝方无人机处于矩形区域内哪些位置时，无论红方追击策略如何，蓝方无人机总有机会成功突防。该问题属于完全信息下的定性微分对策问题，而求解蓝方的安全区域等同于求解博弈对策中的躲避区，也相当于求解躲避区与捕获区的界栅问题。第二小问中蓝方最优突防策略，其中蓝方最优策略下该微分对策模型的哈密顿函数对蓝方策略的偏导数为 0。

4.2 模型准备

4.2.1 坐标系建立及运动约束

坐标系的选取和定义及其坐标系之间的变换是构建无人机模型在二维空间中运动的基础和框架。为了方便本文的后续研究，现介绍本文所使用的二维空间坐标以及一些角度做一个说明，本文所使用的坐标系和角度经过一定的简化以方便后续研究。

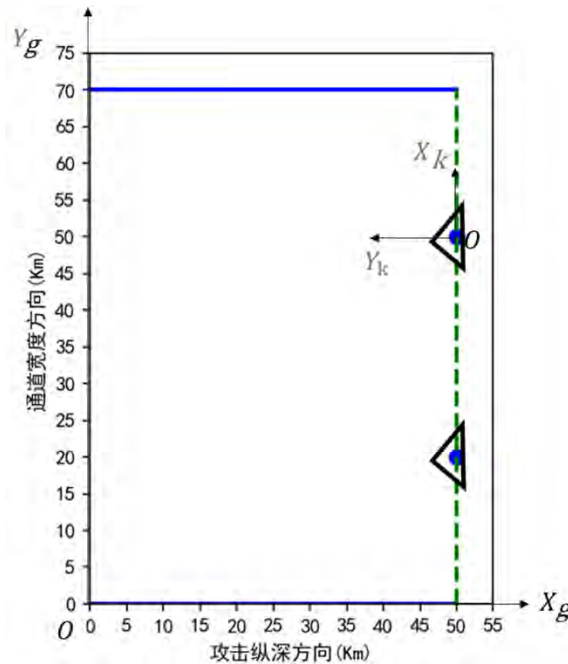


图 4-1 地面与航迹坐标系

地面坐标系 $OX_g Y_g$ ：原点为对抗矩形区域的 B 点，x 坐标轴 OX_g 沿着 BC 方向与攻击通道平行，y 坐标轴 OY_g 沿着 BA 方向通道垂直。

航迹坐标系 $OX_k Y_k$ ：对在地面坐标系运动的无人机，都会有自己的航迹坐标系，随着飞行器在地面坐标系的位置改变。其航迹坐标系的原点相对地面坐标系的位置就会发生

改变，原点与飞行器质心重合。 OX_k 轴为飞行器的地速方向， OY_k 轴垂直于 OX_k 轴，航迹坐标系符合右手原则。

航迹偏角 χ :也叫航向角，为航迹坐标系中 OX_k 轴与地面坐标系 OX_g 轴的夹角。并定义如下：当飞行器向左偏转飞行时航迹偏角角度数值为负，当飞行器向右偏转飞行时航迹偏角角度数值为正，当飞行器不进行偏转飞行时航迹偏角角度数值为 0。

地面与航迹两个坐标系由于构造方式相同，可以相互转换，其转换矩阵如下：

$$L_{kg} = \begin{pmatrix} \cos \chi & \sin \chi \\ \sin \chi & \cos \chi \end{pmatrix} \quad (4.1)$$

本文中所采用二维平面中两自由度刚体质点动力学模型，不考虑飞机的姿态也就不会有飞行器的姿态角方程和力矩方程。主要是基于运用以上介绍的航迹坐标系和地面坐标系基础上探究飞机的运动情况。

\vec{i}_g 地面坐标系 x 轴的单位向量， \vec{j}_g 为地面坐标系 y 轴的单位向量， \vec{i}_k 为航迹坐标系 x 轴的单位向量， \vec{j}_k 为航迹坐标系 y 轴的单位向量。由航迹坐标系与地面坐标系的转换关系得到如下公式：

$$\begin{bmatrix} \vec{i}_k \\ \vec{j}_k \end{bmatrix} = \begin{bmatrix} \cos \chi & \sin \chi \\ \sin \chi & \cos \chi \end{bmatrix} \begin{bmatrix} \vec{i}_g \\ \vec{j}_g \end{bmatrix} \quad (4.2)$$

将速度转换到地面坐标系的两个方向上得到如下两个坐标变换公式：

$$\dot{x} = V \cos \chi \quad (4.3)$$

$$\dot{y} = V \sin \chi \quad (4.4)$$

综上所述，公式(4.2)-(4.4)即为本文后续研究所采用平面刚体质点动力学模型。从模型上可以看出，只要给出出发点及航偏角 χ 随时间改变的变换规律函数，通过数值带入，然后积分求解得到当前时刻飞行器（刚体质点）在地面坐标系(航迹坐标系)中的坐标，经过一段时间后得到一系列的坐标的集合就是飞行器的运动轨迹。

对无人机动作进行分析，每个机动动作是飞行器在平面中能够做出的基本飞行器机动动作。平飞时， $\chi=0$ ，航迹偏角为 0；左转弯时， $\chi < 0$ ，航迹偏角小于 0；右转弯时， $\chi > 0$ ，航迹偏角大于 0；

根据题目红蓝双方无人机的速度保持不变，蓝方无人机的速度为 $V_E = 250m/s$ ，最小转弯半径为 $R_E = 500m$ ；红方无人机的速度为 $V_P = 200m/s$ ，最小转弯半径为 $R_P = 350m$ ；红方运载机的速度为 $V_{\text{红}} = 300m/s$ ，转弯半径不小于 $1000m$ 。则受转弯半径的限制。

$$V = \omega r = \chi r \quad (4.5)$$

则蓝方无人机的航偏角范围为 $\chi_E \in \left[-\frac{1}{2}, \frac{1}{2}\right] rad$ ；红方无人机的航偏角范围为 $\chi_P \in \left[-\frac{4}{7}, \frac{4}{7}\right] rad$ ；红方运载机的航偏角范围为 $\chi_M \in \left[-\frac{3}{10}, \frac{3}{10}\right] rad$ 。

$$\begin{cases} v_E = 250 \\ x_e \geq 0 \\ y_e \in (0, 70) \\ \chi_E \in \left[-\frac{1}{2}, \frac{1}{2}\right] \end{cases} \quad (4.6)$$

约束条件分别表示为：蓝方无人机的速度为 $v_E = 250m/s$ ；蓝方无人机横向纵向位置不得超过边界；最小转弯半径为 $R_E = 500m$ ；

$$\begin{cases} v_p = 200 \\ V_{\text{红}} = 300 \\ \chi_P \in \left[-\frac{4}{7}, \frac{4}{7}\right] \\ \chi_M \in \left[-\frac{3}{10}, \frac{3}{10}\right] \end{cases} \quad (4.7)$$

约束条件分别表示为：红方无人机的速度为 $v_p = 200m/s$ ；最小转弯半径为 $R_p = 350m$ ；红方运载机的速度为 $V_{\text{红}} = 300m/s$ ，转弯半径不小于 $1000m$ ；

4.2.2 微分对策问题

航天器突防本质上是一个多方控制的连续动态对抗问题，微分对策理论是研究和分析该问题的有力工具。

对策理论是研究利益相互制约的双方策略选择的理性行为及相应结局的学科。尽管对策论的模型形式上可以差别很大，但每一个对策论模型都包含以下三个基本元素：局中人（player）、策略集（strategy set）、支付函数（cost function）。在一个对策模型中，为争取得到对自己有利的结局、达到各自的目的，并且对于自己的行动方案有决策权的参与者称为“局中人（player）”。一个对策模型中的局中人不得少于两个。一个对策模型中，可供决策者采纳的一个完整的、实际可行的、可用于实现决策者目的的行动方案称为一个“策略（strategy）”。策略集（strategy set）是对策模型中的每一个局中人都应该具有的。在空战中，策略集就是无人飞机能采用的各种机动动作的总和，在微分对策模型中，通过输入控制向量可以得到各种机动动作，故可以用无人战机的控制向量替代策略集 [4]。

微分对策是指局中人在进行对策活动时，用微分方程（组）来描述对策现象或规律的一种对策。属于对策论的范畴。它是处理双方或多方连续动态冲突、竞争或合作问题的一种数学工具。微分对策实质上是一种双（多）方的最优控制问题，它将现代控制理论与对策论相融合，从而比控制理论具有更强的竞争性和对抗性。

微分对策理论是在考虑一定的环境下参与对抗或者竞争活动的对手之间关系问题时，通过借助数学上的微分方程或方程组等工具以数据化的方式描述整个关系问题的现象和内在规律的一种以时间为主轴的动态对策。

微分对策种类繁多，根据划分标准的不同，分类也就不同。根据有无支付函数可分为定量与定性微分对策；对于定量微分对策而言，若其支付之和为零，则称为零和微分对策，若不为零，则称为非零微分对策；按照信息结构的不同可划分为完全信息、不完

全信息和无信息微分对策；按照对策问题中动态系统类型的不同，可分为偏微分对策和随机微分对策，偏微分对策用偏微分方程来描述动态系统，随机微分对策存在随机的干扰或观测误差；按照局中人的数目进行分类，可分为二人和多人微分对策；按照局中人是否有合作的动机，又可分为合作与非合作微分对策。

①定量微分对策

定量微分对策是指结局的得失在连续范围内变化的微分对策，即其支付函数 J 可以在一定的范围内连续变化， J 是一个连续函数。

②定性微分对策

定性微分对策没有支付函数，不研究某一性能指标的极值，而是研究某种结局能否实现（如击中或捕获）。它通过界栅分隔出躲避区和捕获区，界栅就是定性微分对策的解。

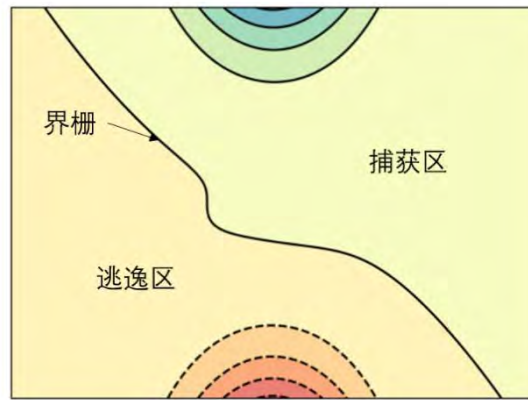


图 4-2 界栅示意图

4.3 二对一无人机突防界栅模型

假设：

- (1) 红方无人机集群中心分别为 F_1, F_2 , 位置坐标分别为 (x_{F1}, y_{F1}) , (x_{F2}, y_{F2})
- (2) 蓝方无人机位置坐标为 (x_H, y_H)
- (3) β 为红方无人机集群坐标系相对于地面坐标逆时针转动的角度
- (4) 红方无人机集群中心 F_1, F_2 和无人机集群坐标系的横轴方向夹角分别为 χ_A , χ_B
- (5) 蓝方无人机和无人机集群坐标系的横轴方向夹角为 χ_C
- (6) 红方无人机集群 F_1 , F_2 最大攻击距离 L_A, L_B

4.3.1 基于队形的无人机集群覆盖区域

无人机编队集中式控制是指集群中的每一架无人机都需要与所有的无人机进行通信。相当于集群中存在一个虚拟的掌握全局信息的中央调度器，由它与集群中的无人机进行通信。分布式控制则不需要集群中的所有成员进行信息交互，每架无人机只需要与有限范围内的相邻无人机进行通信，通过邻居无人机的影响进行下一步的运动。分散式控制的结构最为简单，每架无人机只与编队中的特定点保持相对关系，不与其他无人机进行通信。这种方式既没有控制中心，也不受相邻无人机影响。

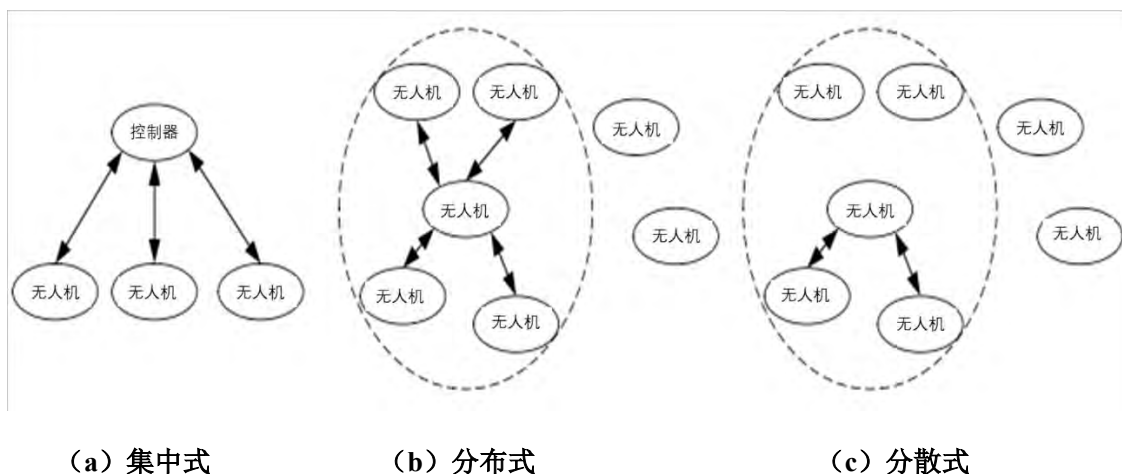


图 4-3 无人机群类型

根据题目，红队五人机为分布式编队。红方任何两架无人机的间距需大于 $30m$ ；每一架无人机与本集群中至少两架无人机的距离不超过 $200m$ ；红方运载机与所属无人机集群中至少一架无人机的距离不超过 $10km$ ；运载机与任何一架无人机的距离需大于 $100m$ ；运载机与蓝方的突防无人机的距离需大于 $5km$ 。

根据题目约束条件随机生成红队无人机队形，队形可以是不规则的多边形或者线性排布。由于无人机之间距离要求数值较小（相比于同运载机之间的最大距离约束），采用格雷厄姆算法求解不同队形多边形区域的最小凸包，得出区域覆盖面积，可以得出面积相差不大。且题目中考虑红方采取任意策略，如果考虑无人机不同队形转换对蓝方拦截的影响，给计算造成很大的不便。在这种情况下，为方便分析与求解，将红方无人机队形采用圆周设计代入计算。

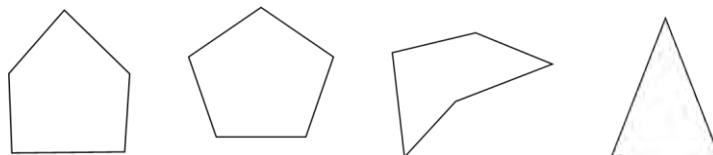


图 4-4 红方无人机队形

考虑安全区域为红方无人机无论采取什么追击策略都无法拦截的，针对此类多 UAV 协同抓捕问题，为降低求解的复杂度，我们假设：（1）红方无人机集群基于圆周队形的凸包构造；（2）在突防过程中，红队的拦截策略可以实现在满足追踪模型约束条件下时刻处于拦截区域内最接近蓝队无人机的位置。

该问题也是多约束条件下的动态组合优化问题，则安全区域内的任意一点应该满足：

一是从该点出发，蓝方一定存在一条可行路径（带宽为 600 米的曲线）成功突防；则问题等同于任意 t 时刻红方无人机的巡查覆盖区域外仍存在蓝方可通过的可行路径（带宽为 600 米的曲线）。

根据假设与题目条件带入计算，可以得到基于队形的无人机集群覆盖区域。

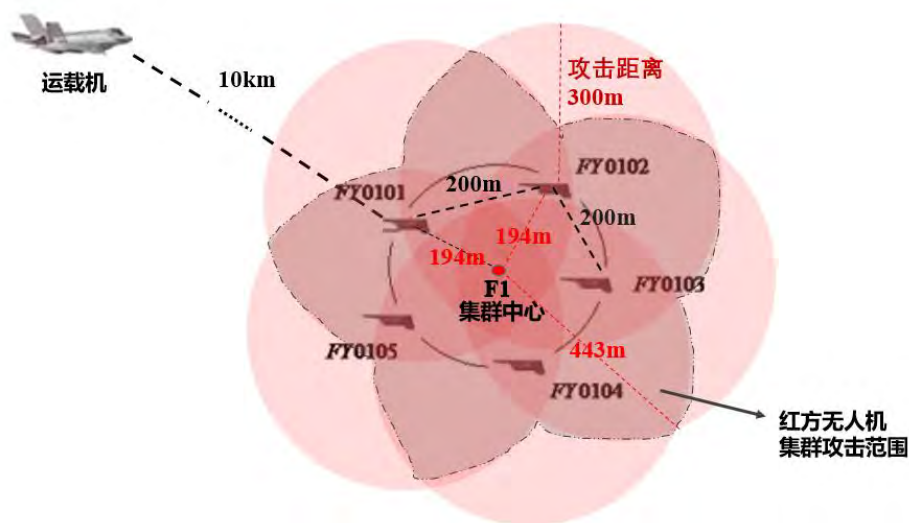


图 4-5 无人机 dt 时间内扫过的区域

4.3.2 双机编队下的坐标系转换

根据题意，红蓝双方无人机都可以实现对方飞机位置的侦察，红方无人机可以进行内部通讯，通过多机协调机动决策和分配机动空域展开蓝方与红方双机空中对抗抓捕。红方执行拦截任务的无人机组成两个无人机集群，可考虑为双机问题，以下所有计算都是在二维空间环境中考虑的参数设置，并以水平平面 XOY 坐标系展开模型说明。

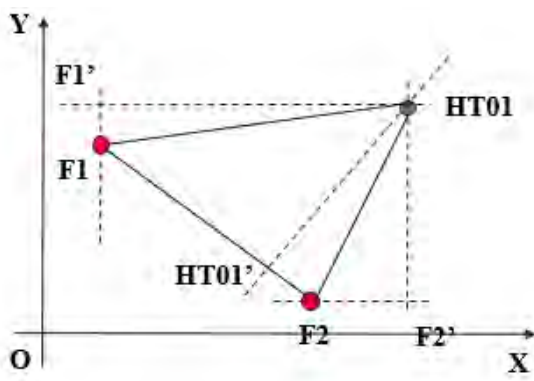


图 4-6 红蓝双方相对坐标系示意图

如上图所示，红方无人机集群中心 F_1 ， F_2 ，蓝方无人机为 $HT01$ ，直线 $F_1 F_2$ 为红方无人机集群中心的连线，同理直线 $F_1 HT01$ 、 $F_2 HT01$ 为红蓝无人机的连线，直线 $HT01 HT01'$ 与 $F_1 F_2$ 相互垂直并构成了三个无人机中心点之间的坐标系 $HT01 HT01' F_1 F_2$ 。由该坐标系可知红蓝双方无人机的相对运动轨迹是由红方一个无人机集群中心 F_1 与蓝方无人机 $HT01$ 之间基于平面坐标系 XOY 的位置关系形成的角度 $\angle F_1 HT01 F_1'$ 来进行计算轨迹。由于红方另一个无人机群 F_2 加入对抗博弈中，针对新加入的变量进行修改以发挥无人机群 F_2 的优势， F_1 ， F_2 形成对蓝方无人机可飞行区域的挤压。在无人机之间相对坐标系 $HT01 HT01' F_1 F_2$ 中，本文以 $\angle HT01' HT01 F_1$ 作为无人机集群 F_1 飞行轨迹变化的关键角度，同理，以 $\angle HT01' HT01 F_2$ 作为无人机集群 F_2 飞行轨迹变化的关键角度。由于无人机集群 F_1 ， F_2 攻击效能一致，所以蓝方无人机 $HT01$ 的运动轨迹由红方无人机集群中心连线的中点坐标确定。最终红蓝双方无人机集群之间的相对坐标系建立以及关键航迹角度改变，使得红方无人机集群之间通过编队坐标系形成相互配合成为了可能。

红蓝方无人机相对坐标系 $HT01$ $HT01'$ F_1 F_2 与无人机空间位置坐标系（水平平面 XOY 坐标系）的相互转换有着固定规律，转换关系如下图：

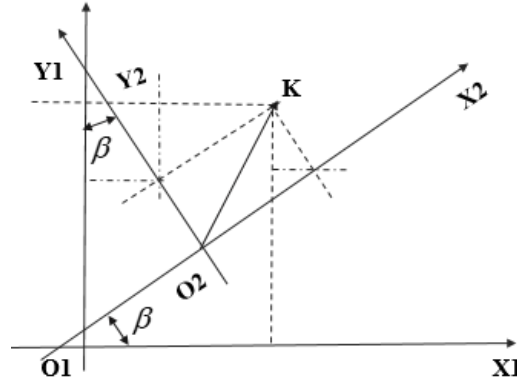


图 4-7 无人机集群相对坐标系与地面坐标系转换

由上图可知坐标系 $HT01$ $HT01'$ F_1 为无人机集群相对坐标系，假设 β 为无人机集群坐标系相对于地面坐标逆时针转动的角度，可得转换关系如下：

$$\begin{aligned} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} &= \begin{bmatrix} \cos(x_1, HT01) & \cos(x_1, F_1) \\ \cos(y_1, HT01) & \cos(y_1, F_1) \end{bmatrix} \begin{bmatrix} HT01 \\ F_1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} HT01 \\ F_1 \end{bmatrix} \\ \begin{bmatrix} HT01 \\ F_1 \end{bmatrix} &= \begin{bmatrix} \cos(HT01, x_1) & \cos(HT01, y_1) \\ \cos(F_1, x_1) & \cos(F_1, y_1) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \end{aligned} \quad (4.8)$$

在二维空间中，根据图形学坐标系转化可得到如下坐标变换公式， (x', y') 为转换后坐标。

$$\begin{cases} x' = x \cos \beta - y \sin \beta \\ y' = x \sin \beta + y \cos \beta \end{cases} \quad (4.9)$$

当无人机集群之间进行坐标系变化时，按照上述变换公式就可以通过旋转变换，得到无人机集群相对坐标系和相关坐标。基于二维水平面坐标系基础，红方无人机集群 F_1 ， F_2 与蓝方无人机 $HT01$ 在水平面微分突防对策，已知蓝方无人机的速度为 $V_E = 250m/s$ ，最小转弯半径为 $R_E = 500m$ ；红方无人机的速度为 $V_p = 200m/s$ ，最小转弯半径为 $R_p = 350m$ 。红方无人机集群中心 F_1 ， F_2 和蓝方无人机与无人机集群坐标系的横轴方向夹角分别为 χ_A ， χ_B ， χ_C 。目标集合 F 是以红方无人机集群中心 F_1 ， F_2 为圆心、以红方无人机集群 F_1 ， F_2 最大攻击距离 L_A, L_B 为半径的圆域，由此可以得到终端集合 F 如下：

$$F = \left\{ \begin{aligned} &(t, x_{F1}, y_{F1}, x_H, y_H) \mid \varphi(x_{F1}, y_{F1}, x_H, y_H) = (x_{F1} - x_H)^2 + (y_{F1} - y_H)^2 - L_A^2 \leq 0 \\ &(t, x_{F2}, y_{F2}, x_H, y_H) \mid \varphi(x_{F2}, y_{F2}, x_H, y_H) = (x_{F2} - x_H)^2 + (y_{F2} - y_H)^2 - L_B^2 \leq 0 \end{aligned} \right\} \quad (4.10)$$

该问题的绝对坐标运动方程如下：

$$X = \begin{bmatrix} x_{F1} = 200 \cos \chi_A & x_{F1}(0) = x_{F1}^0 \\ y_{F1} = 200 \cos \chi_A & y_{F1}(0) = y_{F1}^0 \\ x_{F2} = 200 \cos \chi_B & x_{F2}(0) = x_{F2}^0 \\ y_{F2} = 200 \cos \chi_B & y_{F2}(0) = y_{F2}^0 \\ x_H = 250 \cos \chi_C & x_H(0) = x_C^0 \\ y_H = 250 \cos \chi_C & y_H(0) = y_C^0 \end{bmatrix} \quad (4.11)$$

4.3.3 界栅构造

则根据题目构建界栅, 哈密顿函数 $H(x, u, v_1, v_2, \lambda)$ 表示状态的变化率。如果轨迹 X 在 ∂D 上或者充分接近 ∂D , 并且 $H(x, u, v_1, v_2, \lambda) > 0$, 那么轨迹 X 不能穿过 ∂D ; 反之, 即 $H(x, u, v_1, v_2, \gamma) < 0$ 则轨迹 X 穿过 ∂D , 此时对局结束。因此, 从 P 与 E 各自的目的出发, P 欲使 $H < 0$, 而 E 欲使 $H > 0$ 。哈密顿函数 $H(x, u, v_1, v_2, \gamma)$ 中含有控制变量 u, v , 因此 P 应该选择最优策略 v , 使 $H(x, u, v_1, v_2, \gamma)|_{\partial D}$ 在目标边界集合 ∂D 上取得尽可能小的负值以保证对局结束, 而 E 应选择最优策略 u 使 $H(x, u, v_1, v_2, \gamma)|_{\partial D}$ 取得尽可能大正值以实现逃脱, 即

$$\max_{u \in U} \min_{v_1, v_2 \in V} H(x, u, v_1, v_2, \gamma) = \min_{v_1 \in V} H_p(v_1) + \min_{v_2 \in V} H_p(v_2) + \max_{u \in U} H_e(v) \quad (4.12)$$

满足 $H(x_p^*, x_e^*, u^*, v_1^*, v_2^*, \gamma)|_{\partial D} > 0$ 的所有点, 称为目标集的不可用部分(NUP)。在 NUP 上, 无论 P 采取何种策略, E 总能躲避捕获, 从而对局结束。满足 $H(x_p^*, x_e^*, u^*, v_1^*, v_2^*, \gamma)|_{\partial D} = 0$ 的所有点, 称为目标集的部分可用边界(BUP)。由 BUP 为起点所形成的界栅与与 NUP 围成的区域构成躲避区。

梯度迭代法的基本思想是: 任取突防双方初值并设定初始轨迹, 从时间零点开始正向积分得到初始点坐标, 然后按照追讨微分对策模型积分计算。

首先创建哈密顿函数:

$$\begin{aligned} H((x_{F1}, y_{F1}, x_{F2}, y_{F2}, x_H, y_H, \chi_A, \chi_B, \chi_C, \lambda_A, \lambda_B, \lambda_C, \lambda) = \\ 1 + 200\lambda_1 \cos \chi_A \cos \lambda_A + 200\lambda_2 \sin \chi_A \cos \lambda_A + 200\lambda_3 \sin \lambda_A \\ + 200\lambda_4 \cos \chi_B \cos \lambda_B + 200\lambda_5 \sin \chi_B \cos \lambda_B + 200\lambda_6 \sin \lambda_B \\ + 250\lambda_7 \cos \chi_C \cos \lambda_C + 250\lambda_8 \sin \chi_C \cos \lambda_C + 250\lambda_9 \sin \lambda_C \end{aligned} \quad (4.13)$$

其中 $\lambda = (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9)^T \in R^9$ 为任意向量
根据题目与假设的顶点的坐标。那么在平面上对局结束条件可表示为,

$$(x_{F1} - x_H)^2 + (y_{F1} - y_H)^2 - L^2 \leq 0 \cup (x_{F2} - x_H)^2 + (y_{F2} - y_H)^2 - L^2 \leq 0 \quad (4.14)$$

$$\text{即 } F_1 = (x_{F1} - x_H)^2 + (y_{F1} - y_H)^2 - L^2 = 0 \quad (4.15)$$

$$\text{或 } F_2 = (x_{F2} - x_H)^2 + (y_{F2} - y_H)^2 - L^2 = 0 \quad (4.16)$$

两个条件任意一个得到满足时对局终止, 可得终端集

$$\begin{aligned}
x_H &= \varphi_1(\chi) = \chi_7 + x_{F1} + x_{F2} \\
y_H &= \varphi_2(\chi) = \chi_8 + y_{F1} + y_{F2} \\
x_{F1} &= \varphi_3(\chi) = \frac{\chi_7 \chi_8 \chi_9}{2} - \chi_1 \chi_2 + \chi_3 \pm \sqrt{\cos^2 \chi_3 + \frac{\chi_2^2}{4} - \chi_1 \chi_2 \chi_3} \\
y_{F1} &= \varphi_4(\chi) = -\frac{\chi_7 \chi_8 \chi_9}{2} - \chi_1 \chi_2 + \chi_3 \pm \sqrt{\cos^2 \chi_3 + \frac{\chi_2^2}{4} - \chi_1 \chi_2 \chi_3} \\
x_{F2} &= \varphi_5(\chi) = \frac{\chi_7 \chi_8 \chi_9}{2} - \chi_4 \chi_5 + \chi_6 \pm \sqrt{\sin^2 \chi_6 + \frac{\chi_5^2}{4} - \chi_4 \chi_5 \chi_6} \\
y_{F2} &= \varphi_6(\chi) = -\frac{\chi_7 \chi_8 \chi_9}{2} - \chi_4 \chi_5 + \chi_6 \pm \sqrt{\sin^2 \chi_6 + \frac{\chi_5^2}{4} - \chi_4 \chi_5 \chi_6}
\end{aligned} \tag{4.17}$$

其中 $\chi = (\chi_1, \chi_2, \chi_3, \chi_4, \chi_5, \chi_6, \chi_7, \chi_8, \chi_9)^T$ ，在界栅上法向量 λ 是任意的，故可取 λ 为单位法线向量，在上同样有

$$\sum_{i=1}^9 \lambda_i \frac{\partial \varphi_i(\chi_1, \chi_2, \chi_3, \chi_4, \chi_5, \chi_6, \chi_7, \chi_8, \chi_9)}{\partial \chi_j} \quad j=1,2,3,4,5,6,7,8,9 \tag{4.18}$$

4.4 模型求解与分析（第一小问）

对于微分对策的解主要分为解析解和数值解。解析解的求解在理论上比较成熟，就是把微分对策看作变分问题，进而利用极大值极小值原理求解所谓的哈密顿—雅克比方程。

以 BUP 的点作为界栅的初始点倒向积分状态微分方程组，状态微分方程组的初始值为对应于 BUP 的状态变量值。积分出来的(轨迹)状态 x^* 就是所求的界栅。假设蓝方五无人机的出发点为 $x_H(0) = x_{H0}, y_H(0) = y_{H0}$ ，倒向积分状态微分方程组及初始条件为：

$$\begin{cases}
x_{F1}^*(0) = 50 \\
y_{F1}^*(0) = 20 \\
x_{F2}^*(0) = 50 \\
y_{F2}^*(0) = 50 \\
x_H^*(0) = x_{H0} + \chi_1 \\
y_H^*(0) = y_{H0} + \chi_2 \\
\frac{dx_{F1}^*}{dt} = -200 \cos \chi_A \\
\frac{dy_{F1}^*}{dt} = -200 \sin \chi_A \\
\frac{dx_{F2}^*}{dt} = -200 \cos \chi_B \\
\frac{dy_{F2}^*}{dt} = -200 \sin \chi_B \\
\frac{dx_H^*}{dt} = 250 \\
\frac{dy_H^*}{dt} = 0
\end{cases} \quad (4.19)$$

积分可得

$$\begin{cases}
x_H^*(t) = x_{H0} + \chi_1 + 250t \cos \chi_c \\
y_H^*(t) = y_{H0} + \chi_2 + 250t \cos \chi_c \\
x_{F1}^* = -200t + \frac{\chi_7 \chi_8 \chi_9}{2} - \chi_1 \chi_2 + \chi_3 \pm \sqrt{\cos^2 \chi_3 + \frac{\chi_2^2}{4} - \chi_1 \chi_2 \chi_3} \\
y_{F1}^* = -\frac{\chi_7 \chi_8 \chi_9}{2} - \chi_1 \chi_2 + \chi_3 \pm \sqrt{\cos^2 \chi_3 + \frac{\chi_2^2}{4} - \chi_1 \chi_2 \chi_3} \\
x_{F2}^* = -200t + \frac{\chi_7 \chi_8 \chi_9}{2} - \chi_4 \chi_5 + \chi_6 \pm \sqrt{\sin^2 \chi_6 + \frac{\chi_5^2}{4} - \chi_4 \chi_5 \chi_6} \\
y_{F2}^* = \frac{\chi_7 \chi_8 \chi_9}{2} - \chi_4 \chi_5 + \chi_6 \pm \sqrt{\sin^2 \chi_6 + \frac{\chi_5^2}{4} - \chi_4 \chi_5 \chi_6}
\end{cases} \quad (4.20)$$

目标集边界值上满足公式 4.10 的所有点，称为目标集的可用部分边界（Boundary of the Usable Part, BUP）。在 BUP 上，追击者 P 与逃逸者 E 采取各自的最优策略，使得对策状态既不能进入目标集内部，又不能脱离目标集边界进入对策空间。显然，BUP 把目标集边界划分为 NUP 和 UP 两部分。目标集可用部分边界 BUP 是界栅的起点，同时也是微分对策结束的终点，由 BUP 为起点所构成的界栅与 UP 所围成的区域构成了对策空间的捕获区，而其余区域则为逃逸区。

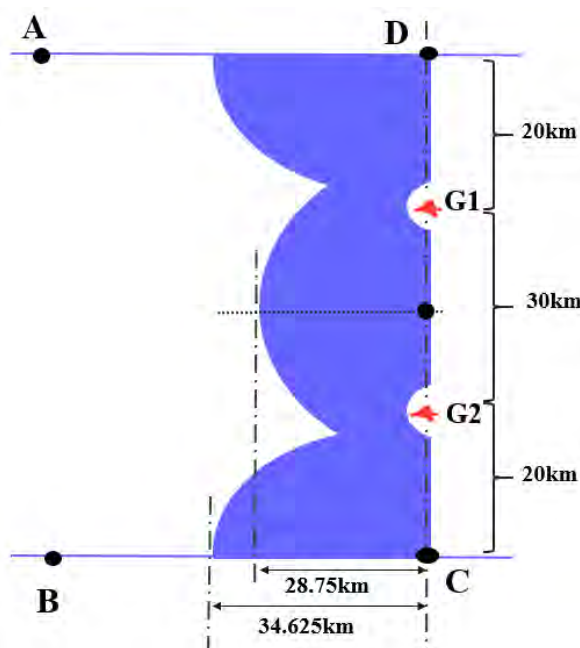


图 4-8 蓝方无人机逃脱区域示意图

$$200 \div \sin(72^\circ \div 2) = 194m, 194 \cos 31^\circ + 300 \cos 23^\circ = 443m \quad (4.21)$$

根据公式 4.21 作图，界栅 D 对应的是穿过 BUP 的半透曲面，为方便观察将图 3a 朝 xoy 平面投影得到图 3b。图 3b 中圆和双曲圆弧代表目标集在 xoy 平面的投影，蓝色阴影部分是由界栅 D 和目标集围成的逃脱区，在逃脱区内运动，轨迹将到达目标集的边界并穿过。在此区域内，只要 E 进入该区，则无论 P 采取何种策略，在 E 选择适当的策略下一定能逃出该区。捕获区之外为捕获区，在此区域上，只要 E 初始位置在该区中，不管 E 采用何种策略，P 总能采取适当的策略对 E 进行的捕获。

表 4-1 蓝方突防博弈最优策略分析表

序号	x (km)	y (km)	\dot{x} (km)	\dot{y} (km)	τ (s)
1	23.5	42.4	35.98	40.36	65.32
2	23.5	53.6	36.45	68.91	78.92
3	18.5	36.7	0	0	-
4	5.1	53.6	24.65	64.32	127.28

表 4-1 中分别给出了四种不同对策初始状态下捕获逃逸区域的计算结果，从表中数据可以看出，第 1、4 种情况下对策状态处于捕获区内，而另外两种情况下的对策状态处于逃逸区内，其中第 3 种情况中在给定时间范围内不存在最小微分对策的解。

对比表中第 1 和第 2 行数据可知，两种情况下的对策初始位置横坐标分量相同，速度大小也相同，不同的仅是初始位置纵坐标分量，而其结果一个是处于捕获区内另一个却处于逃逸区内，这表明对策状态处于捕获区还是逃逸区内受初始纵坐标分量影响。

对比表中第 2 和第 4 行数据可知，两种情况下的对策初始位置纵坐标分量相同，速度大小也相同，而其结果一个是处于捕获区内另一个却处于逃逸区内，这表明对策状态处于捕获区还是逃逸区内受初始横坐标分量影响。

虽然初始位置坐标对可逃脱性的影响非线性，但是存在一定的相关关系。其中纵坐标与红方初始位置的纵坐标越接近，其可逃脱性越小。横坐标与逃脱边界 CD 的距离越

大，可逃脱性越小。

对比表中第 2 和第 3 行数据可知，两种情况下对策的初始速度均相同，其中第 2 种情况下的对策初始相对距离更小，但其反而处于捕获区中，这表明初始相对距离小并不意味着逃逸航天器更容易被捕获，捕获是否能够发生主要取决于两航天器的相对速度方向和大小。

4.5 最优突防策略模型

在问题一的基础上求解蓝方无人机相应的最优突防策略，不同情况下的量化指标是根据对抗双方各自不同的任务目标进行确定。在飞行器与飞行器的空战对抗的定量突防问题中，所以该问题的支付函数通常为对抗双方可以距离以及追讨时间等，量化指标过多会增加计算难度。本题目采用突防时间大小和两机距离远近作为量化指标。支付函数构建如下：

$$J(\chi_A, \chi_B, \chi_C) = L_1^2 + L_2^2 + \tilde{t} = \int_0^{\tilde{t}} dt + (x_{F1} - x_H)^2 + (y_{F1} - y_H)^2 + (x_{F2} - x_H)^2 + (y_{F2} - y_H)^2 \quad (4.22)$$

$\tilde{t} = t(\chi_A, \chi_B, \chi_C)$ 是蓝方无人机坐标第一次出现红方无人机集群攻击范围圆域的时间，

即为捕获时间。根据哈密顿函数定义在最优路径上， $\frac{dH_p(v)}{dv} = 0$ ，即

$$V_p(-\lambda_1 \sin v_1 - \lambda_2 \sin v_2 + \lambda_3 \cos v_1 + \lambda_4 \cos v_2) = 0 \quad (4.23)$$

$$\begin{cases} \sin v_1^* = \frac{-\lambda_2}{\sqrt{\lambda_1^2 + \lambda_2^2}} \\ \cos v_1^* = \frac{-\lambda_1}{\sqrt{\lambda_1^2 + \lambda_2^2}} \end{cases} \quad (4.24)$$

显然，有 $\frac{d^2 H_p(v)}{dv^2} = V_p(-\gamma_1 \cos v_1^* - \gamma_2 \cos v_2^* + \gamma_3 \sin v_1^* + \gamma_4 \sin v_2^*) = 0$ ，故由该式确定的 v^* 是无人机的最优策略。要使 $H_e(v)$ 取得极大值，应该选取的策略为：

$$u^* = \text{sgn } \gamma_5 \quad (4.25)$$

所以可得：

$$H(x_p^*, x_e^*, u^*, v_1^*, v_2^*, \gamma) = \gamma_5 u^* v_e - v_p \sqrt{\gamma_1^2 + \gamma_2^2 + \gamma_3^2 + \gamma_4^2} \quad (4.26)$$

令 $\frac{\partial H}{\partial \chi_C} = 0$ ，则哈密顿函数公式通过计算后可得：

$$\lambda_5 \cos \chi_C^* - \lambda_4 \sin \chi_C^* = 0 \quad (4.27)$$

同理

$$\begin{cases} \sin \chi_B^* = \frac{-\lambda_5}{\sqrt{\lambda_4^2 + \lambda_5^2}} \\ \cos \chi_B^* = \frac{-\lambda_4}{\sqrt{\lambda_4^2 + \lambda_5^2}} \end{cases} \quad (4.28)$$

显然有：

$$\left. \frac{\partial^2 H}{\partial \chi_B^2} \right|_{\chi_B = \chi_B^*} = -\lambda_5 \sin \chi_B^* - \lambda_4 \cos \chi_B^* \leq 0 \quad (4.29)$$

故 χ_C^* 为红方无人机集群 F_2 的最优策略，其中*表示未来策略。

同理，分别令 $\frac{\partial H}{\partial \chi_C} = 0$ ，分别可得：

$$\begin{cases} \sin \chi_C^* = \frac{-\lambda_8}{\sqrt{\lambda_7^2 + \lambda_8^2}} \\ \cos \chi_C^* = \frac{-\lambda_7}{\sqrt{\lambda_7^2 + \lambda_8^2}} \end{cases} \quad (4.30)$$

针对该问题的伴随方程组及其末值条件如下：

$$\begin{cases} \lambda_i = 0 \quad (i = 1, 2, 3, 4, 5, 6) \\ \lambda_1(\tilde{t}) = 3\mu(x_{F1}^*(\tilde{t}) - x_H^*(\tilde{t})) \\ \lambda_2(\tilde{t}) = 3\mu(y_{F1}^*(\tilde{t}) - y_H^*(\tilde{t})) \\ \lambda_3(\tilde{t}) = 3\mu(x_{F2}^*(\tilde{t}) - x_H^*(\tilde{t})) \\ \lambda_4(\tilde{t}) = 3\mu(y_{F2}^*(\tilde{t}) - y_H^*(\tilde{t})) \\ \lambda_5(\tilde{t}) = -3\mu(x_H^*(\tilde{t}) - \frac{x_{F1}^*(\tilde{t}) + x_{F2}^*(\tilde{t})}{2}) \\ \lambda_6(\tilde{t}) = -3\mu(y_H^*(\tilde{t}) - \frac{y_{F1}^*(\tilde{t}) + y_{F2}^*(\tilde{t})}{2}) \end{cases} \quad (4.31)$$

伴随方程组及其末值条件的设定，主要是根据无人机集群的态势信息和任务属性确定的，蓝方无人机为突防方，由于红方无人机性能相同，随意将选取红方无人机集群中点坐标作为规避基础计算坐标点，通过积分可得：

$$\lambda_1(t) = \lambda_1(\tilde{t}), \lambda_2(t) = \lambda_2(\tilde{t}), \lambda_3(t) = \lambda_3(\tilde{t}), \lambda_4(t) = \lambda_4(\tilde{t}), \lambda_5(t) = \lambda_5(\tilde{t}), \lambda_6(t) = \lambda_6(\tilde{t}) \quad (4.32)$$

结合绝对运动方程可以得到蓝方无人机博弈后的运动轨迹方程如下：

$$\begin{cases} x_H^*(t) = x_H^0 + 250t \cos \chi_C \\ y_H^*(t) = y_H^0 + 250t \sin \chi_C \end{cases} \quad (4.33)$$

$$\begin{cases} x_H''(t) = x_H^* \cos \beta - y_H^* \sin \beta \\ y_H''(t) = x_H^* \sin \beta + y_H^* \cos \beta \end{cases} \quad (4.34)$$

4.6 模型求解与分析（第二小问）

该最优突防策略基于哈密顿函数求得，其最优性体现在保证逃脱可能性的基础上尽可能快的突防。由于解析较为复杂，无法通过公式形象说明蓝方最优突防策略。

由第一小问可知，蓝方初始位置的横纵坐标决定了蓝方是否具有突防的可能。但需要说明的是蓝方在最优突防策略下并不一定能突防成功，也就是蓝方初始坐标值不位于逃脱区。同时，当蓝方位于逃脱区域，有突防成功的可能时，也不一定存在最优突防策略。最优策略的存在与该问题下的哈密顿函数的微分解存在性相关。

该问题假设红蓝双方速度大小不变，且红方的初始位置固定。则由界栅模型也可以看出，蓝方的可逃脱行受初始位置横纵坐标分量影响，且逃脱的可能性为连续性变化。通过带入数值到最优策略的公式 4.38-4.42 试算，可以大致估计蓝方最优策略的情形。

蓝方的最优策略为根据红方的当前位置与速度信息，预判相遇（X 坐标一致）点的位置，确保相遇点在突防区域 ABCD 内，且在使得逃脱距离红方两个无人机集群的中心点 Y 轴相对距离和尽可能的远。采用分段函数的方法对不同取值下蓝方的大致突防策略进行说明。

当蓝方初始位置靠近 CD 边界时，蓝方趋向于沿着 X 轴正方向运动尽快突防，除非初始位置与红方初始点的纵坐标位置过于靠近，此时运动方向一边向 CD 边界移动一边保证与红方两个无人机集群的中心点的安全距离；

当蓝方初始位置靠近 AB 边界时，蓝方不一定能够逃脱红方的追捕，此时蓝方的最优策略为先沿着 X 轴正方向前近，当与红方无人机群具有一定距离时（到达对抗区域时）再进行调整速度的方向，使得距离红方两个无人机集群的中心点 Y 轴相对距离尽可能的远；

当蓝方初始位置纵坐标大于红方初始纵坐标一定值时，蓝方最优策略倾向于向 Y 轴正方向偏转。当蓝方初始位置纵坐标小于红方初始纵坐标一定值时，蓝方最优策略倾向于向 Y 轴负方向偏转。当蓝方初始位置纵坐标位于红方两初始纵坐标之间时，蓝方速度的偏角倾向于使得距离红方两个无人机集群的中心点 Y 轴相对距离和尽可能的远。

5. 问题二

5.1 问题分析

假设同问题一，问题二为生存型定量微分对策问题，即突防双方都想要尽快达到捕获或逃脱目标。假设存在一个通道带宽下限 M_{\min} ，则带宽取值大于该数值时，蓝方无人机采用最优策略一定能突破红方无人机集群的拦截，即蓝方处于逃脱区。则带宽取值为该数值时，蓝方出发点（AB 中点）位于界栅上，即哈密顿函数对蓝方策略取极值。

定量突防问题会针对适用的不同情况会有不同的具体量化标准，由于定量突防问题针对的适用情况不同就会有不同的支付函数，也就会不同的支付标准和量化标准，为求

解最快突防策略，将支付函数设定为追逃时间相关。

5.2 双机编队下时间最优的无人机突防模型

支付函数构建如下：

$$J(\chi_A, \chi_B, \chi_C) = \tilde{t} = \int_0^{\tilde{t}} dt \quad (5.1)$$

$\tilde{t} = t(\chi_A, \chi_B, \chi_C)$ 是蓝方无人机坐标第一次出现红方无人机集群攻击范围圆域的时间，即为捕获时间。

创建哈密顿函数：

$$\begin{aligned} H((x_{F1}, y_{F1}, x_{F2}, y_{F2}, x_H, y_H, \chi_A, \chi_B, \chi_C, \lambda_A, \lambda_B, \lambda_C, \lambda) = \\ 1 + 200\lambda_1 \cos \chi_A \cos \lambda_A + 200\lambda_2 \sin \chi_A \cos \lambda_A + 200\lambda_3 \sin \lambda_A \\ + 200\lambda_4 \cos \chi_B \cos \lambda_B + 200\lambda_5 \sin \chi_B \cos \lambda_B + 200\lambda_6 \sin \lambda_B \\ + 250\lambda_7 \cos \chi_C \cos \lambda_C + 250\lambda_8 \sin \chi_C \cos \lambda_C + 250\lambda_9 \sin \lambda_C \end{aligned} \quad (5.2)$$

其中 $\lambda = (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9)^T \in R^9$ 为任意向量，令 $\frac{\partial H}{\partial \chi_B} = 0$ ，则哈密

顿函数公式通过计算后可得：

$$\lambda_5 \cos \chi_B^* - \lambda_4 \sin \chi_B^* = 0 \quad (5.3)$$

同理

$$\begin{cases} \sin \chi_B^* = \frac{-\lambda_5}{\sqrt{\lambda_4^2 + \lambda_5^2}} \\ \cos \chi_B^* = \frac{-\lambda_4}{\sqrt{\lambda_4^2 + \lambda_5^2}} \end{cases} \quad (5.4)$$

显然有：

$$\left. \frac{\partial^2 H}{\partial \chi_B^2} \right|_{\chi_B = \chi_B^*} = -\lambda_5 \sin \chi_B^* - \lambda_4 \cos \chi_B^* \leq 0 \quad (5.5)$$

故 χ_B^* 为红方无人机集群 F_2 的最优策略，其中*表示未来策略。

同理，分别令 $\frac{\partial H}{\partial \chi_A} = 0$ ， $\frac{\partial H}{\partial \chi_C} = 0$ ，分别可得：

$$\begin{cases} \sin \chi_A^* = \frac{-\lambda_2}{\sqrt{\lambda_1^2 + \lambda_2^2}} \\ \cos \chi_A^* = \frac{-\lambda_3}{\sqrt{\lambda_1^2 + \lambda_2^2}} \end{cases} \quad (5.6)$$

$$\begin{cases} \sin \chi_C^* = \frac{-\lambda_8}{\sqrt{\lambda_7^2 + \lambda_8^2}} \\ \cos \chi_C^* = \frac{-\lambda_7}{\sqrt{\lambda_7^2 + \lambda_8^2}} \end{cases} \quad (5.7)$$

针对该问题的伴随方程组及其末值条件代入公式，伴随方程组及其末值条件的设定，主要是根据无人机集群的态势信息和任务属性确定的，蓝方无人机为突防方，由于红方无人机性能相同，随意将选取红方无人机集群中点坐标作为规避基础计算坐标点，通过积分可得：

$$\lambda_1(t) = \lambda_1(\tilde{t}), \lambda_2(t) = \lambda_2(\tilde{t}), \lambda_3(t) = \lambda_3(\tilde{t}), \lambda_4(t) = \lambda_4(\tilde{t}), \lambda_5(t) = \lambda_5(\tilde{t}), \lambda_6(t) = \lambda_6(\tilde{t}) \quad (5.8)$$

结合绝对运动方程可以得到红蓝方无人机博弈后的运动轨迹方程如下：

$$\begin{cases} x_{F1}^*(t) = x_{F1}^0 + 200t \cos \chi_A \\ y_{F1}^*(t) = y_{F1}^0 + 200t \sin \chi_A \\ x_{F2}^*(t) = x_{F2}^0 + 200t \cos \chi_B \\ y_{F2}^*(t) = y_{F2}^0 + 200t \sin \chi_B \\ x_H^*(t) = x_H^0 + 250t \cos \chi_C \\ y_H^*(t) = y_H^0 + 250t \sin \chi_C \end{cases} \quad (5.9)$$

$$\begin{cases} x_{F1}''(t) = x_{F1}^* \cos \beta + y_{F1}^* \sin \beta \\ y_{F1}''(t) = x_{F2}^* \cos \beta - y_{F2}^* \sin \beta \\ x_{F2}''(t) = x_{F2}^* \cos \beta - y_{F2}^* \sin \beta \\ y_{F2}''(t) = x_{F2}^* \sin \beta + y_{F2}^* \cos \beta \\ x_H''(t) = x_H^* \cos \beta - y_H^* \sin \beta \\ y_H''(t) = x_H^* \sin \beta + y_H^* \cos \beta \end{cases} \quad (5.10)$$

5.3 突防策略分析

5.3.1 突防通道区域栅格离散化处理

这里将巡查区域离散化为 5000*7000 的栅格，选取每个单元栅格的几何中心点作为飞行航路规划的节点，在后面的问题中，涉及搜索路径的优化时，将基于栅格的所有元素在其中心赋予不同类型的概率权重。

在处理时，由于上一步划分后的多边形区域是不规则的，因此某些栅格几何中心不在

多边形内部，为保证尽可能覆盖多的巡查区域，可采用整体坐标平移或插入航路点方法，将遗漏区域添加进入需搜索栅格中，提高覆盖精度。

离散化处理后，任务巡查区域被划分为栅格，栅格中的元素称为节点，将 UAV 视为质点，假设每个步长无人机运动一步，即从当前单元格运动到相邻的某个单元格。由于受转弯特性的限制，无人机在 $t+1$ 时刻的航向可以有 3 个选择，即沿 t 时刻航向直行，或者向左转 45° ，或者右转 45° 。

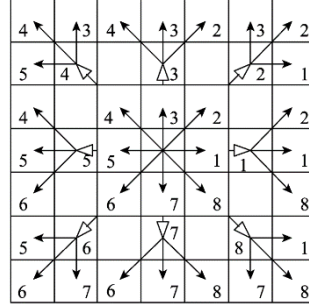


图 5-1 UAV 在任一时刻航行方向示意图

5.3.2 UAV 突防航迹优化规划

对红方 UAV 分配的拦截区域离散化处理后，将本题转化为单 UAV 突防航迹优化规划问题，站在蓝方的决策角度，共有 2 个优化目标：

- (a) 尽可能快的越过红方边界；
- (b) 执行本次任务代价最小（蓝方无人机无损失）。

决策变量为 X , $X=\{x_{pij}|p=1,\dots,N,i,j=N_T\}$ ，当 $x_{pij}=1$ 时，表示该 UAV 沿着节点 ij 的方向飞过，反之。

根据实际巡查过程的分析，可以得出，当红方无人机集群按照无人机之间最大距离限制安排队形时拦截覆盖面积最大，因此考虑在红方拦截覆盖范围最大的情况下蓝方突防时间最短。我们认为，航行区域中距离红方无人机集群越近的点被拦截的概率越大，将每个元素按照距离分配大小不等的概率权重 Pr_i 。

针对优化目标 (a)，首先实现越过红方边界的终极目标，使无人机突防的时间最短。考虑栅格元素概率 Pr_i 的航行目标量化为：

$$N_t(X) = \sum_{p=1}^N \sum_{i=0}^{N_T} \sum_{j=0}^{N_T} Pr_i x_{pij} \quad (5.11)$$

显然， $0 < N_t(X) \leq 1$ ，当取值接近于 1 时，表明此时蓝方无人机以 200s 的时间直线抵达边界，没有受到任何干扰，此时需结合条件 b 进一步优化：

针对优化目标 (b)，执行本次任务代价最小（蓝方无人机无损失），即要求蓝方的突防航线与红方拦截范围不重合。

执行任务过程中需要满足的约束包括：(1) 突防总时间不超过 360s；(2) UAV 自身性能约束，如飞行速度、无人机间距、转弯半径等。在本题中，主要考虑巡查时间的代价函数，由于飞行速度固定，因此可表示为突防路径的代价函数：

$$f_2(X) = Dis(X) = \sum_{p=1}^N \sum_{i=0}^{N_T} \sum_{j=0}^{N_T} Dis_{ij} X_{pij} \quad (5.12)$$

因此，最终的单目标多 UAV 协同对抗任务规划模型表示为：

$$\min \vec{f}(X) = (f_1(X), f_2(X)) \quad (5.13)$$

$$f_1(X) = \varepsilon(N + 2 - N_t(X)) + (1 - \varepsilon)N_v(X) \quad (5.14)$$

$$f_2(X) = Dis(X) = \sum_{p=1}^N \sum_{i=0}^{N_T} \sum_{j=0}^{N_T} Dis_{ij} X_{pij} \quad (5.15)$$

s.t.

$$\left\{ \begin{array}{l} \forall j = 1, \dots, N_T : \sum_{p=1}^N \sum_{i=0}^{N_T} xp_{ij} \leq 1 \\ \sum_{i=0}^N \sum_{j=1}^{N_T} xp_{ij} (t_{ij} + w_j + st_j) \leq TL \\ \sum_{l=1}^{N_T} x_{pol} = 1 \\ \sum_{l=1}^{N_T} x_{pol} = 1 \\ w_j = \max(0, ej - (T_i + st_i + t_{ij})) \\ 0 < \sum T_i < 360 \\ \forall i = 1, \dots, N, 30 < Dis_{ii} \leq 200 \\ \forall i = 1, \dots, N, Dis_{ij} \leq 300 \end{array} \right. \quad (5.16)$$

约束条件分别表示为：每个无人机拦截范围有限；每个 UAV 的总飞行时间不能超过其任务时间约束；UAV v_p 对目标 j 进行拦截的约束条件；UAV v_p 与运载机之间的条件约束；UAV 的无人机集群之间条件约束；总突防时间不超过 360s。

5.4 模型求解

表 5-1 初始状态设置

名称	x (Km)	y (Km)	v (m/s)
蓝方初值	0	35	250
红方一初值	50	y_{G1}	200
红方二初值	50	y_{G2}	200

针对上述模型，我们采用量子蚁群算法，即量子计算与蚁群算法相结合的一种概率优化方法。

(1) 蚁群算法介绍

蚁群算法的基本思想和基本原理来源于自然界蚂蚁觅食的最短路径原理。

自然界的蚂蚁虽然视觉不发达，但它可以在没有任何提示的情况下找到食物源到巢穴的最短路径，并且能够在环境发生变化后自适应地搜索新的最佳路径。蚂蚁种群觅食的过程为：蚂蚁在寻找食物源时，能在其走过的路径上释放一种蚂蚁特有的分泌物——信息素，使得一定范围内的蚂蚁能察觉到并由此影响他们以后的行为。当一些路径上通过的蚂蚁越来越多时，其留下的信息素也越多，以致信息素的强度增强，所以蚂蚁选择该路径的概率也越高，从而增加了改路径的信息素强度。

蚁群算法是一种正反馈机制，即增强型学习系统。它的原理可以理解为：对于单个蚂蚁而言，它并没有要寻找最短路的主观上的故意，但对于整个蚁群系统来说，它们又确实达到了寻找最短路径的客观效果。

(2) 量子算法介绍

量子计算是一门基于量子力学的基本原理实现信息处理的计算技术。量子计算中的信息存储单元是量子比特，信息状态是不确定的，且信息状态可以叠加，区别于信息存储单元中只能是比特 1 或 0 的经典计算。量子计算中表示信息的基本单元是量子比特 $|\varphi\rangle$ ，量子比特的“0”的状态为 $|0\rangle$ ，量子比特的“1”的状态为 $|1\rangle$ ，量子比特 $|\varphi\rangle$ 由它们按一定的概率叠加而成，量子态 $|\varphi\rangle$ 可以表示为：

$$|\varphi\rangle = |0\rangle + |1\rangle \quad (5.17)$$

其中， α 、 β 分别是量子态 $|\varphi\rangle$ 的基础态 $|0\rangle$ 和 $|1\rangle$ 的概率幅， α 、 β 都为复数，且满足 $|\alpha|^2 + |\beta|^2 = 1$ 。量子态通过观察过程，使得量子态坍塌成一种确定的状态，其坍塌的确定状态是由概率幅所对应的，相应概率幅的平方就是向改状态坍塌的概率。

(3) 量子蚁群算法原理

量子蚁群算法是量子计算与蚁群算法相结合的一种概率优化方法。其主要工作原理是：将蚁群算法中的各路径上的信息素进行量子比特编码，得到一种新的信息素——量子信息素，量子蚁群算法的实质就是在蚁群算法中融合进量子特性，相应的信息素的更新规则变为通过量子旋转门策略与最优路径相结合来实现，旋转门策略可以通过一定的旋转门调整方式实现。通过研究发现，与单纯的蚁群算法相比，量子蚁群算法通过利用量子信息素及距离信息来创建路径，使得算法的收敛性得到很大的提高，从而提高了算法的性能。主要过程为：量子蚂蚁编码，量子信息素更新。

(4) 基于量子蚁群算法的多无人机突防任务规划

利用量子蚁群算法为无人机寻找协同突防最优航迹，首先主要需明确一下几个问题：

- 蚂蚁起始位置的确定。一般离散优化问题中蚂蚁可以随机的选取任意节点作为起始位置，对于航迹规划问题，在本题中，无人机将以探测边界与搜索边界交点和搜索边界中点为起始点。
- 蚂蚁终止位置的确定。对于离散优化问题，处于这个环境中的蚂蚁走过突防通道到达红方边界即为终止位置。
- 产生多组解，在这一阶段，红蓝双方无人机都需要由系统提供备选航迹，选择合理的参数，通过取多次循环的并集，能够得到不同的几组解。

用 N_{max} 表示每一次执行量子蚁群算法系统需要迭代的次数，每架无人机执行 N 次量子蚁群算法，量子蚁群算法提供待搜索的节点集为 T ，设节点构成 m 条边。下面给出了利用量子蚁群算法解决无人机 N 条航迹规划问题的具体步骤：

Step1: 初始化各参数。设蚂蚁个数为 n 个，量子编码的量子位个数为 m ， t 时刻量子蚁群可表示为： $A(t) = (\tau_1^t, \tau_2^t, \dots, \tau_n^t)$ ， $\tau_i^t (i=1, 2, \dots, n)$ 为第 t 次迭代中的第 i 个量子蚂蚁个体的量子信息素矩阵：

$$\tau_i^t = \begin{bmatrix} \alpha_{i1}^t & \alpha_{i2}^t & \dots & \alpha_{im}^t \\ \beta_{i1}^t & \beta_{i2}^t & \dots & \beta_{im}^t \end{bmatrix} \quad (5.18)$$

当前迭代次数 $t=0$ ，量子信息素矩阵各元素都初始设置为 $1/\sqrt{2}$ ，即

$$A(O) = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & \dots & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} & \dots & 1/\sqrt{2} \end{bmatrix}_{2 \times m} \quad (5.19)$$

Step2: 首先把每个蚂蚁放置于明确的起始位置，然后红蓝两方按照伪随机概率公式选择路径，直至所有蚂蚁遍历完所有边，记录蚂蚁对各边的选择信息，生成候选解。

参数 q_0 的设定同样可以影响该算法的收敛及寻优性能。本文对参数 q_0 设置如下， $q_0 = 0.94 - 0.3t/t_{all}$ ，体现了算法的自适应特性，即在算法执行初期， q_0 较大，使得蚂蚁能更好地发挥群体智能，但随着迭代次数增加 q_0 ，也逐渐减小，使得 q 相对的容易满足 $q > q_0$ ，则蚂蚁随机搜索的能力就增强了，从而大大减少了算法陷入局部最优的可能性。

Step3: 记录每只量子蚂蚁的可行解。并计算其相应的突防航迹代价。

Step4: 若 n 只蚂蚁都组织完成了各自的解，则转向 Step5，否则转向 Step3。

Step5: 记录本次迭代后 n 只量子蚂蚁构造出来的全局最优解。采用量子自适应的旋转门调整策略来更新量子蚁群信息素 $A(t)$ ，依据当前解得到全局最优解。

Step6: 若满足结束条件，即输出得到的最优解，否则 $t=t+1$ 。转向 Step3。

Step7: 若 $t = N_{max}$ ，则转向 Step8，否则，转向 Step3。

Step8: 输出得到的最优解及其在雷达探测范围内的相应距离。

Step9: $N = N - 1$ ，转至 Step10。

Step10: 若满足结束条件，算法结束。即 $N=0$ ，则输出 N 组最优解，即得到无人机 UAV V^{P1} 的 N 条最优备选突防航迹。否则，转至 Step1。

根据以上步骤，得到 UAV 突防最优航迹。

讨论了在基本蚁群优化理论下的改进蚁群算法。首先,介绍了基本蚁群算法的优化思想、算法模型,分析了蚁群算法的优化特点。其次,介绍了典型蚁群算法中的信息素更新机制。针对现有信息素更新机制中蚁群信息利用不充分的问题,引入了信息素的再励学习更新机制。机制通过对蚁群搜索行为的评价,产生再励信号,依据再励信号实施信息素“奖惩”更新,发扬优质搜索行为而回避劣质行为,实现蚁群行为的快速自学习对比实验表明引入再励学习更新机制的蚁群算法具有更强的寻优能力。最后,讨论了多子群蚁群优化算法,给出了基于问题划分的异质多子群蚁群优化框架,讨论了框架中“子问题”结合策略对算法复杂度的影响提出了预编序号策略和控制配队策略,将问题复杂度从指数级降到了可控的特定规模同时针对这两种策略可能存在的空间覆盖率较低的问题,给出了基于正交实验设计的子问题解结合策略。

5.5 结果分析

上式 5.7, 5.9, 5.10 显示, 当问题二的条件下, 蓝方无人机集群总是沿着红方无人机集群中心与蓝方无人机形成的三角形 F1, F2, HT01, F1F2 的中点与 HT01 的连线的延长线的某一点的连线飞行和逐渐靠近, 以此来压缩蓝方无人机的可飞行机动空域。

根据以上算法对红方初始值设定对蓝方出发点区域哈密顿函数值以及通道宽度对蓝方出发点区域哈密顿函数值的影响进行分析, 并求出通道宽度下限值。

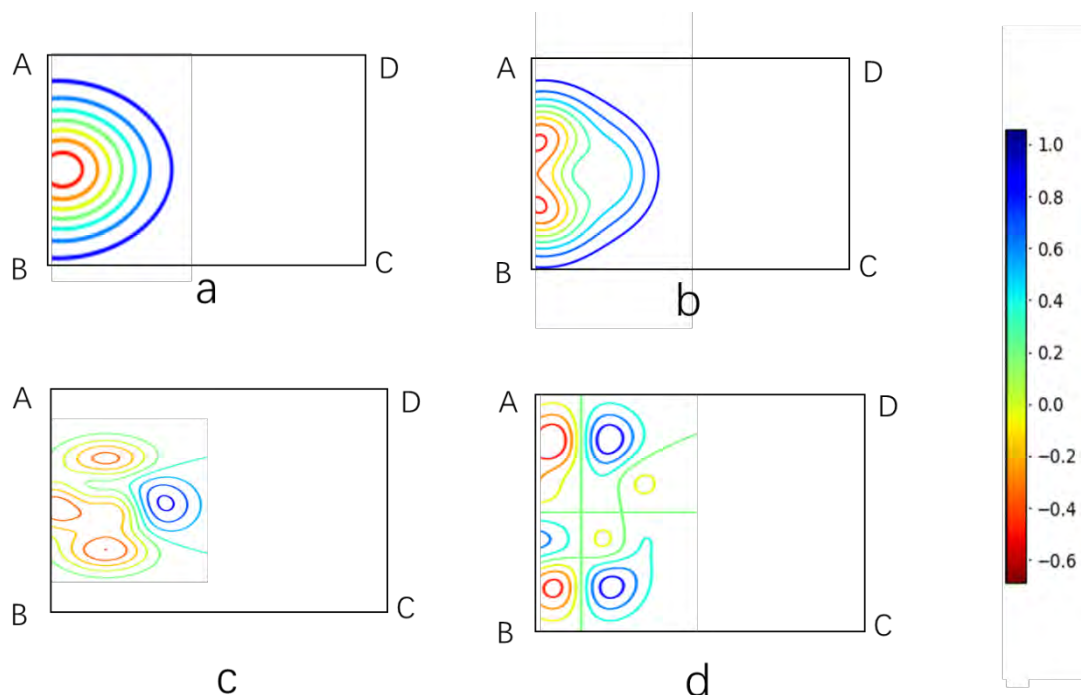


图 5-2 红方、位置对 AB 中点局部区域哈密顿函数的影响图

如图为在 ABCD 突防区域内，随着红方 G_1 、 G_2 位置变化蓝方出发点（AB 中点）局部区域内的哈密顿函数数值变化图。其中，哈密顿函数数值越大越易逃脱。将通道宽度取值为 85Km，假设红蓝双方均按照最优策略突防。其中 a 情况为 G_1 、 G_2 沿着通道中心对称分布，其起始 Y 坐标分别为（90,30）、（90,55）；b 情况为 G_1 、 G_2 沿着通道中心对称分布，其起始 Y 坐标分别为（90,20）、（90,65）；c 情况 G_1 、 G_2 不对称分布，其起始 Y 坐标分别为（90,32）、（90,65）；d 情况 G_1 、 G_2 不对称分布，其起始 Y 坐标分别为（90,14）、（90,72）。

由图可以看出，当红方出发点对称分布时，出发点靠近通道中线时，蓝方初始点的哈密顿函数数值较小，即更易被捕获。

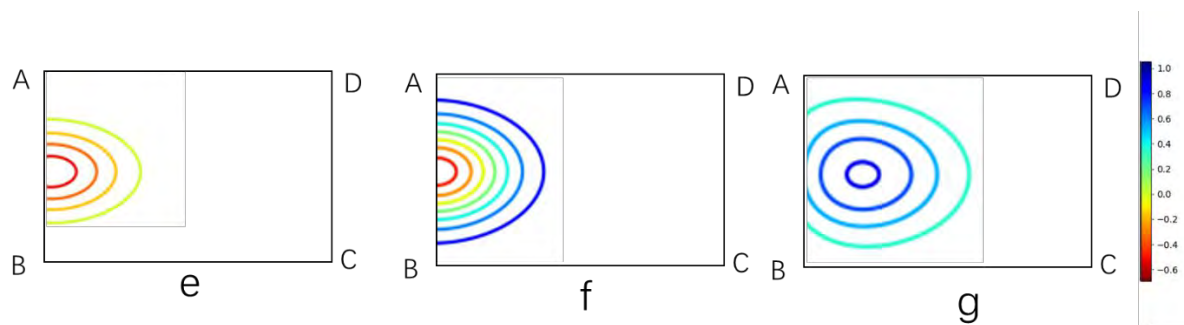


图 5-3 通道宽度对 AB 中点局部区域哈密顿函数的影响图

如图为在 ABCD 突防区域内，随着通道宽度变化蓝方出发点（AB 中点）局部区域内的哈密顿函数数值变化图。假设红蓝双方均按照最优策略突防，红方按照最优位置设定出发点。其中，哈密顿函数数值越大越易逃脱，数值大于零，点位于逃脱区；数值小于零，点位于追捕区。其中 e,f,g 对应的通道宽度分别为 75，85，100。

当蓝方出发点（AB 中点）的哈密顿函数值恰好为 0 时，对应的通道宽度下限 M_{\min} 为 92.2Km。

6. 问题三

6.1 问题分析

问题三红方运载机可以发射两个波次的无人机，即基于微分对策理论研究二维空间内多对一以及多对多无人机突防问题。突防模型中拦截过程的发射策略（数量分配、路径规划）问题。

在追踪无人机试图接近逃逸无人机的过程中，两无人机通常距离较近，互相可测，此时对抗双方均根据博弈状态确定最优控制策略，从而达到各自的目标。基于定量微分对策理论研究无人机轨道追逃博弈问题时，追逃双方的最优控制策略即为微分对策的纳什均衡或鞍点微分对策制导率是真正的双边机动下的最优制导率。

建立拦截飞机末端问题微分对策模型，简化模型。取导弹飞机的相对位移分量作为状态量建立运动方程。捕获是否成功只关心飞机对策结束时相对距离，令对策结束时相对距离作为支付函数。为方便求解，进一步引入状态转移矩阵将状态量降至二维。

无人机集群作战是由多架具备部分自主能力的无人机，在高速机间链路和外部信息平台的情报支援下遂行共同战斗任务的过程。此过程涉及两个方面的问题：(1) 双方对抗策略选择。无人机集群交战前对抗双方策略选择多样化。(2) 双方决策收益估计。对抗双方不同的策略选择对应着不同的战场收益。针对对抗双方选择的策略进行对应的收益值评估，然后双方根据收益值的大小比较判定所选策略的优劣，最终做出最优决策。

避障突防的基本算法一般分为完备的路径规划算法，如 A* 算法、Dijkstra 算法，以及基于采样的路径算法，采样算法、快速随机树算法(Rapidly-exploring Random Tree, RRT)和概率图法等。还有一些间接的路径规划算法，基于经验的路径算法，如仿生算法，遗传算法，人工势场法等。完备的路径规划算法能够产生路径最优解，但要面对较大的数据量；基于采样的路径算法，计算速度快，也能接近最优解；基于经验的路径算法则各有优劣。本章结合两种基本算法优势，然后进行混合算法的求解。

编队飞行即多架无人机为适应任务要求而进行的某种队形排队和任务分配的组织模式，包括编队飞行的队形产生、保持和变化，以及飞行任务的规划和组织。多架无人机编队飞行可以完成单机不能完成的任务，但也带来了许多复杂问题，如任务配合、航迹规划、队形的产生和保持等。

J.J.D'Azzo[6]基于古典控制理论设计了二维平面的无人机编队队形保持控制器，并采用序列二次规划(SQP)方法优化了具有固定控制结构的 PID 控制器的增益系数，改善了控制效果。文献[7]中研究了无人机在自由空间和障碍空间的编队飞行，定义了“安全”和“危险”两种工作模式，设计了双重控制策略[8-12]。在安全模式下，由于编队控制器考虑了所有无人机的动特性，达到了全局最优；在危险模式下，基于改进的神经网络设计了一种新的控制方法。这种双重控制方法在二维空间使用几何方法得到了最优或次优轨迹，并在二维空间中拓展了躲避障碍物的方法；为了解决实际飞行中的约束，使用了基于跟踪控制的模型预测控制方法追踪参考目标，通过数字仿真验证了此法的有效性。

本章主要设计了主拦截机集群及辅拦截机集群的编队控制律，首先给出了航迹系内目标和拦截机的质心动力学和运动学方程，在此基础上给出了主拦截机的比例导引律和过载稳定控制系统；在辅拦截机运动学方程的基础上，建立了发射点惯性系内编队指令跟踪误差的状态方程，设计了编队跟踪指令；最后，将辅拦截机跟踪过载指令的动力学过程近似为二阶动态环节，对中制导段进行了仿真，仿真结果验证了设计的编队控制律的有效性。

6.2 模型建立

随着突防技术手段与策略的多元化发展，飞行器存在多种突防方式，单枚导弹可能无

法完成对目标的有效打击。在提高导弹个体拦截性能的同时，考虑作战策略与拦截手段进行互通信、多层次、立体化的拦截是防空导弹未来发展的趋势所在。体系化防御作战的过程中也离不开导弹之间的通信合作，导弹协同制导律作为导弹协同的关键所在。

多弹编队协同指的是：多弹在拦截目标过程中相互通信并保持着一定的相对空间位置关系，适当时候还进行一定队列变换进行打击。

6.2.1 微分对策制导方法

拦截无人机与目标微分对策的系统方程为：

$$\dot{Z}(t) = \theta_P^{\max} f_P(t_f, t) u_P(t) + \theta_E^{\max} f_E(t_f, t) u_E(t) \quad (6.1)$$

在拦截过程中，拦截无人机通过采用控制策略 $u_P(t)$ 使拦截无人机与目标的零控脱靶量 $Z(t)$ 减小，并最终收敛到零；目标则通过采用控制策略 $u_E(t)$ 使零控脱靶量 $Z(t)$ 增大，以避免被拦截无人机拦截。因此，以终端时刻 t_f 拦截无人机与目标的零控脱靶量 $Z(t)$ 为指标，将微分对策的支付函数取为：

$$J = |Z(t_f)| \quad (6.2)$$

当拦截无人机与目标的初始拦截状态为 (t_0, Z_0) ，如果拦截无人机和目标的控制策略对 (u_P^*, u_E^*) 对任意可用策略对 (u_P, u_E) 满足

$$J(t_0, Z_0; u_P^*, u_E) \leq J(t_0, Z_0; u_P^*, u_E^*) \leq J(t_0, Z_0; u_P, u_E^*) \quad (6.3)$$

则称策略对 (u_P^*, u_E^*) 为微分对策的鞍点。鞍点策略 u_P^* 和 u_E^* 分别为拦截无人机的最优拦截策略和目标的最优逃逸策略。将拦截无人机和目标采用最优控制策略 (u_P^*, u_E^*) 时支付函数的值定义为微分对策的值，即：

$$V(t_0, Z_0) = J(t_0, Z_0; u_P^*, u_E^*) \quad (6.4)$$

为求解拦截无人机和目标的最优控制策略 (u_P^*, u_E^*) ，根据式(4-12)构造微分对策的哈密顿函数：

$$H = \lambda \left[a_P^{\max} f_P(t_f, t) u_P(t) + a_E^{\max} f_E(t_f, t) u_E(t) \right] \quad (6.5)$$

其中， λ 为伴随变量，满足以下伴随方程及边界条件

$$\frac{d\lambda}{dt} = -\frac{\partial H}{\partial Z} = 0 \quad (6.6)$$

$$\lambda(t_f) = \frac{\partial |Z(t_f)|}{\partial Z(t_f)} = \text{sgn}\{Z(t_f)\} \quad (6.7)$$

由伴随变量 λ 的连续性得到 λ 的表达式为 $\lambda(t) = \text{sgn}\{Z(t_f)\}$

根据微分对策的最大值和最小值原理，拦截无人机和目标的最优控制策略 (u_P^*, u_E^*) 满足：

$$u_P^* = \arg \min H = -\text{sgn}\{Z(t_f) f_P(t_f, t)\} u_P^{\max} \quad (6.8)$$

$$u_E^* = \arg \max_{u_E} H = -\text{sgn}\{Z(t_f) f_E(t_f, t)\} u_E^{\max} \quad (6.9)$$

其中， u_P^{\max} 和 u_E^{\max} 为拦截无人机和目标的最大控制输入。如果拦截无人机和目标控制输入的分量均有界，即对 $\forall_i \in \{1, \dots, n_c\}$ 满足 $|u_i^j| \leq \bar{u}_i^j$ 。定义

$$\bar{u}_j = \left[\bar{u}_1^j \cdots \bar{u}_{n_c}^j \right]^T, j \in \{P, E\} \quad (6.10)$$

则 u_P^{\max} 和 u_E^{\max} 可表示为

$$u_j^{\max} = \frac{\bar{u}_j}{\|\bar{u}_j\|}, j \in \{P, E\} \quad (6.11)$$

拦截无人机和目标的最大控制输入的模为1，即 $\|u_j^{\max}\| = 1$ 。

将拦截无人机与目标的最优控制策略(u_p^*, u_E^*)代入系统方程式(6-1), 从任意时刻 t 积分到 t_f , 得到系统的最优轨迹为:

$$Z^*(t) = Z(\tau_f) - \text{sgn}\{Z(\tau_f)\} \int_t^{\tau_f} a_E^{\max} |f_E(t_f, \tau)| u_E^{\max} - a_P^{\max} |f_P(t_f, \tau)| u_P^{\max} d\tau \quad (6.12)$$

6.2.2 多对一拦截模型

多枚拦截无人机对同一目标进行拦截时, 拦截无人机与目标的相对运动关系如图 6-1 所示。

在拦截过程中, 拦截无人机 $i \in \{1, \dots, n\}$ 的加速度方向与视线相垂直, 目标的加速度与速度方向相垂直。因此, 拦截无人机 i 和目标的速度和速度倾角对时间的导数可表示为:

$$\dot{V}_{P_i} = a_{P_i} \sin(\theta_{P_i} - \lambda_i), \dot{\theta}_{P_i} = a_{P_i} \cos(\theta_{P_i} - \lambda_i) / V_{P_i} \quad (6.13)$$

$$\dot{V}_E = 0, \dot{\theta}_E = a_E / V_E \quad (6.14)$$

其中, a_{P_i} 为拦截无人机 i 的机动加速度; θ_{P_i} 为拦截无人机 i 的速度倾角; λ_i 为拦截无人机 i 与目标的视线角。拦截无人机 i 与目标沿垂直于视线方向的加速度分量为:

$$a_{P_i}^\perp = a_{P_i}, a_E^\perp = a_E \cos(\theta_E - \lambda_i) \quad (6.15)$$

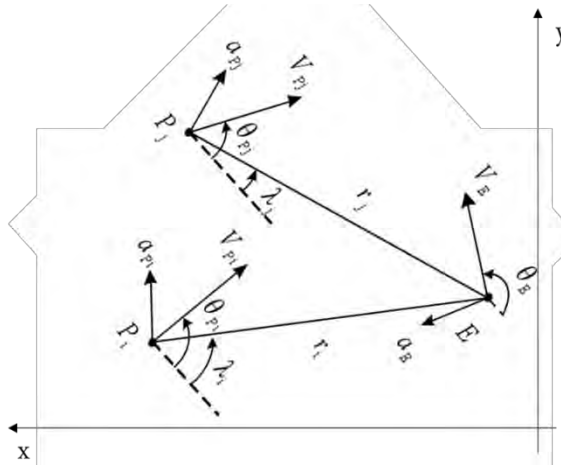


图 6-1 平面拦截几何关系

假设各枚拦截无人机与目标的相对运动满足线性化条件。拦截无人机 i 与目标在垂直于视线方向的位置偏差 y_i 满足:

$$\ddot{y}_i = k_{Ei} a_E - a_{P_i}, i \in \{1, \dots, n\} \quad (6.16)$$

其中

$$k_{Ei} = \cos(\theta_T^0 - \lambda_i^0) \quad (6.17)$$

拦截无人机 i 与目标的相对运动方程可表示为

$$\dot{Z}_i(t) = a_{P_i}^{\max} f_{P_i}^y(t_{fi}, t) u_{P_i}(t) + a_E^{\max} f_{Ei}^y(t_{fi}, t) u_E(t), i \in \{1, \dots, n\} \quad (6.18)$$

其中, $Z_i(t)$ 为 t 时刻拦截无人机 i 与目标的零控脱靶量, 并且:

$$f_{P_i}^y(t_f, t) = f_{P_i}(t_{fi}, t) \cdot 1(t_{fi} - t) \quad (6.19)$$

$$f_{Ei}^y(t_f, t) = k_{Ei} f_E(t_{fi}, t) \cdot 1(t_{fi} - t) \quad (6.20)$$

$$1(\tau) = \begin{cases} 0, & \tau < 0 \\ 1, & \tau \geq 0 \end{cases} \quad (6.21)$$

由于当 $t > t_{fi}$ 时拦截无人机 i 结束运行, 其状态参数失去意义。因此, 采用单位阶跃函数 $1(t_{fi} - t)$ 对函数进行修正。

定义拦截无人机*i*对目标的拦截角为：

$$\mathcal{G}_i(t) = \pi + \theta_{Pi} - \theta_E \quad (6.22)$$

对式 6.22 求导，根据式(6-1)和式(6-2)得到：

$$\dot{\mathcal{G}}_i(t) = k_i \frac{a_{Pi}}{V_{Pi}} - \frac{a_E}{V_E} \quad (6.23)$$

其中：

$$k_i = \cos(\theta_{Pi}^0 - \lambda_i^0) \quad (6.24)$$

如果拦截无人机*i*对目标的拦截角 $\mathcal{G}_i(t)$ 的绝对值小于 90° ，则拦截无人机*i*迎击目标；如果 $\mathcal{G}_i(t)$ 的绝对值大于 90° ，拦截无人机*i*追击或阻击目标。如果从*t*时刻开始，拦截无人机*i*和目标均不进行机动，在拦截时刻 t_{fi} 得到的拦截角即为拦截无人机*i*对目标的零控拦截角，记为 $Z_{n+i}(t)$ 。根据式(5-10)， $Z_{n+i}(t)$ 对时间的导数可表示为：

$$\dot{Z}_{n+i}(t) = a_{Pi}^{\max} f_{Pi}^{\mathcal{G}}(t_f, t) u_{Pi}(t) + a_{Ei}^{\max} f_{Ei}^{\mathcal{G}}(t_f, t) u_E(t) \quad (6.25)$$

其中：

$$f_{Pi}^{\mathcal{G}}(t_f, t) = -\frac{k_i}{V_{Pi}} f_{Pi}(t_{fi}, t) \cdot 1(t_{fi} - t) \quad (6.26)$$

$$f_{Ei}^{\mathcal{G}}(t_f, t) = -\frac{k_i}{V_E} f_E(t_{fi}, t) \cdot 1(t_{fi} - t) \quad (6.27)$$

将拦截无人机*i*与相邻拦截无人机*i+1*对目标的拦截角之差定义为二者的协同拦截夹角 $\delta\mathcal{G}_i(t)$ ，即：

$$\delta\mathcal{G}_i(t) = \mathcal{G}_i(t) - \mathcal{G}_{i+1}(t), i \in \{1, \dots, n-1\} \quad (6.28)$$

6.2.3 Dubins 与 RRT 结合的突防算法

在拦截突防问题中，由于拦截-目标相对运动方程一般是非线性的，因此很难通过解析方法求解微分博弈过程的纳什均衡。因此考虑运用 RRT 算法对系统状态变量分离层后对非线性方程进行求解，但求解过程相当复杂。

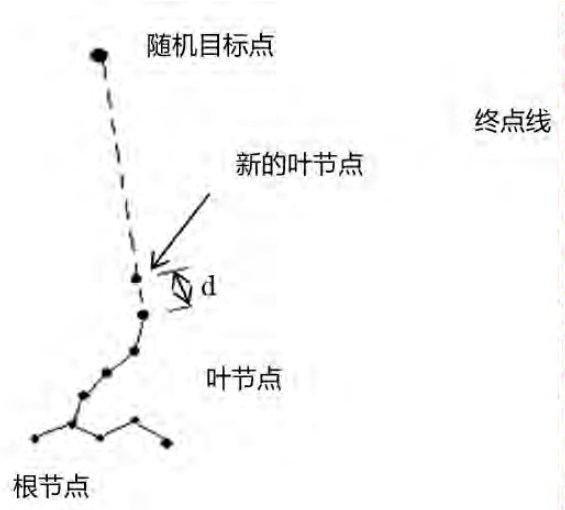


图 6-2 RRT 叶节点构造示意图

RRT 算法能够根据当前环境快速有效地搜索空间，通过在搜索空间中随机采样，将搜索导向空白区域，适用于解决包含几何约束和动力学约束的航迹规划问题。目前 RRT 已应用于飞行器运动规划和移动机器人路径规划。RRT 算法是一种智能的可实施的算法，其最大优点是可行、智能以及轨迹不易捕捉。其基本原理是利用地图上生成的点的随机性，转

化为搜索树的扩展方向的随机性，搜索所有可能扩展的区域，最终可形成从原点到达目标点附近的搜索树。然后通过利用搜索树从目标点进行回溯，检出从原点到达目标点路径的叶节点[10]。

如图 6-2 所示，RRT 通过逐渐添加近距离的随机叶节点而蔓延开来，以初始点作为根节点，首先通过特定的随机采样确定下一步的目标点，然后从当前树节点中寻找距离下一步目标点最近的叶节点，并以一定的步长 d (通常为满足动力学约束的距离常量或时间常量) 计算新的叶节点，如果新的叶节点行进的过程中遇到障碍，则返回空节点，并重新选择任意点继续计算。当随机树的叶节点中包含了终点或者终点区域的点，便可在随机树中通过父节点回溯，找到一条以树节点组成的从初始点到终点的规划路径。RRT 方法基本步骤如下：

1. 生成随机点；
2. 找到扩展树上距随机点最近的点，为父节点；
3. 生成叶节点(在父节点和随机点的连线上)；
4. 判断叶节点是否在障碍内，是则舍去回到步骤 1，否则进行下一步；
5. 判断叶节点是否在目标附近，是则完成搜索，否则继续步骤 1；
6. 从最后节点不断取父节点回溯，得到路径。

RRT 由于盲目搜索产生的无效搜索路径太多，影响路径规划的速度。引入文章中目标导向的启发概率 γ ，在选取任意点时，一部分时间以概率 γ 选取终点作为目标点扩展，其余时间则以概率 $(1-\gamma)$ 在搜索空间内选取一个随机点作为目标点进行扩展。使其有更大概率导向目标点产生有效路径，减少无效搜索路径。但启发概率 γ 对路径选择的影响也要进行统计分析。

RRT 算法搜索能力强，能在全局可行域中搜索并累积叶节点，不会面临局部收敛问题；适用范围广，可适用于各种复杂的障碍环境。RRT 规划出来的是折线路径，不适合用来实际飞行，所以通常需要和其它曲线平滑方式组合改进，应用带有曲线优化的基本模板改进原算法。

栅格化处理后，以上问题可描述为，在需要巡查的区域 S' (海拔 $H < 4000\text{m}$) 内，分布有栅格内元素集合，集合中各元素的地理位置均已知，有 N 架无人机完成对目标的巡查。

针对机器人路径规划的平滑连续性问题，前人提出了一些基于不同几何原理的路径规划方案，如 Dubins 曲线[11]。Clothoid 曲线，Pythagorean hodograph 曲线等。这一小节将简单介绍 Dubins 路径规划的概念。详细算法将发展设计可飞行的安全的 Dubins 路径转移，使得无人机能够在两航路点之间实现状态转移。Dubins 曲线被认为是这种转移的最短路线，并且拥有简单的几何构造，有效的计算方法。

在本节中，路径规划定义是自由空间里两航路点状态间的曲线的几何演变过程。在二维平面内，航路点状态包括坐标 (x, y) 和方向角 θ 。首先考虑的是两种姿态间的路径产生情况，路径都可以用一系列的中间航路点和姿态表示，如下公式。

$$P_s(x_s, y_s, \theta_s) \xrightarrow{r(q)} P_f(x_f, y_f, \theta_f), r(q) \in C_{free}, |\kappa(t)| \leq \kappa_{\max} \quad (6.29)$$

其中 P_s 和 P_f 表示起点和终点的状态量。 $\kappa(t)$ 则表示路径曲线 $r(q)$ 的参数 q ， κ_{\max} 表示各项无人机的机动性能约束，比如最大侧向加速度约束。飞行运动包括直线运动、转弯和有一定角度的运动。并且，圆航线提供一个固定的转弯半径，它满足最大侧向加速度约束。因此，转弯半径是速度和最大侧向加速度的函数。Dubins 曲线是两个向量平面内的最短路径，该路径符合转弯半径的最小约束[12-14]。Dubins 路径是一种混合路径，它由两段圆航线和它们的公切线组成。

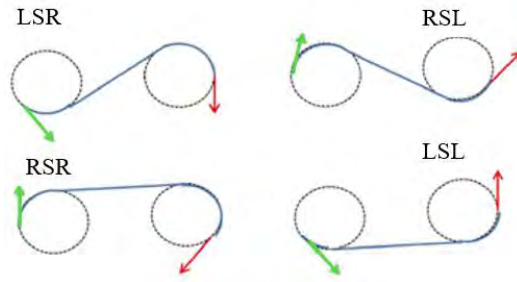


图 6-3 Dubins 曲线的四条路径

由于 Dubins 曲线简单的几何构造，计算效率很高。因而，Dubins 曲线通常用几何分析原理构造。有两种可能的圆弧线间的切线，一种是外切，即开始段和结束段方向相同，如图 6-3 中的下面两种情况(RSR 和 LSL)[15]。一种是内切，开始段和结束段方向相反，如图 6-3 中的上两种情况(LSR 和 RSL)。

Dubins 路径产生有四个要素：

开始状态： $P_s(x_s, y_s, \theta_s)$ ，结束状态： $P_f(x_f, y_f, \theta_f)$

开始转弯半径： ρ_s ，结束转弯半径： ρ_f

其计算步骤包括：

找到开始转弯中心 $o_s(x_{cs}, y_{cs})$ 和结束转弯中心 $o_f(x_{cf}, y_{cf})$

$$(x_{cs}, y_{cs}) = (x_s \pm \rho_s \cos(\theta_s \pm \pi/2), y_s \pm \rho_s \sin(\theta_s \pm \pi/2)) \quad (6.30)$$

$$(x_{cf}, y_{cf}) = (x_f \pm \rho_f \cos(\theta_f \pm \pi/2), y_f \pm \rho_f \sin(\theta_f \pm \pi/2)) \quad (6.31)$$

当 $\rho_f \geq \rho_s$ 时，在 o_f 处画辅助圆，半径为 $|\rho_f - \rho_s|$ 。

连接圆心 o_s 和 o_f 形成线段 c ，线段长度 $|c| = \sqrt{(x_{cs} - x_{cf})^2 + (y_{cs} - y_{cf})^2}$ ，称中心线。

画一条线段 $o_s T'$ 与辅助圆相切于点 T' 。过圆心 o_f 作关于 $o_s T'$ 的垂线，交结束转弯圆 C_f 于点 T_N 。 T_N 将是 Dubins 曲线中切线段的结束点。

画一条线 $T_X T_N$ 平行于直线 $o_s T'$ ， T_X 将是 Dubins 曲线中切线段的起始点。

通过半径 ρ_s 的圆弧连接点 ρ_s 和 T_X ；同理，通过半径 ρ_f 的圆弧连接点 ρ_f 和 T_N 。

Dubins 混合路径就形成了：开始圆弧段 $P_s T_X$ ，切线段 $T_X T_N$ ，结束圆弧段 $T_N \rho_f$ 。

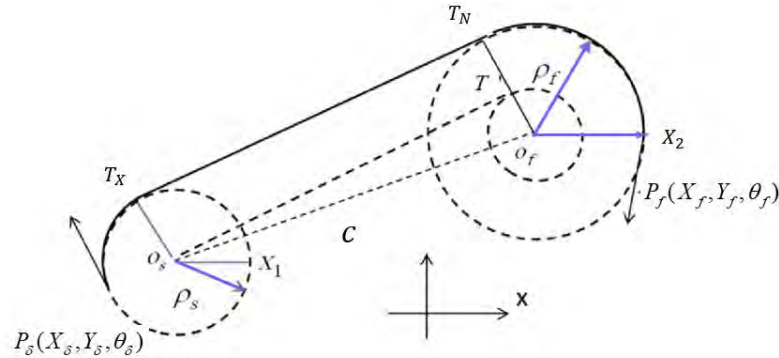


图 6-4 Dubins 曲线构造解析

三角形 $\Delta o_s o_f T'$ 是一个以 $o_s o_f$ 为斜边的直角三角形，其它两直角边是 $o_s T'$ 和 $o_f T'$ ，其中 $\|o_f T'\| = |\rho_f - \rho_s|$ ，它们的夹角 $\angle T' o_s o_f$ 用 ϕ_e 表示。

$$\phi_e = \arcsin\left(\frac{\rho_f - \rho_s}{|c|}\right) \quad (6.32)$$

线段 c 的斜率是 ψ ，表示为

$$\psi = \arctan \left(\frac{y_{cf} - y_{cs}}{x_{cf} - x_{cs}} \right) \quad (6.33)$$

表 6-1 外切方式的起始点

开始处转弯	结束处转弯	ϕ_{ex}	ϕ_{en}
顺时针	顺时针	$\phi_c + \frac{\pi}{2} + \psi$	$\phi_c + \frac{\pi}{2} + \psi$
逆时针	逆时针	$\phi_c - \frac{\pi}{2} + \psi$	$\phi_c - \frac{\pi}{2} + \psi$

所以切线的起始点可以归纳为表 6-1, $\angle X_1 o_s T_x$ 的角表示为 ϕ_{ex} , $\angle X_2 o_f T_N$ 的角表示为 ϕ_{en} 。则切线起始点的坐标是:

$$\begin{aligned} T_X &= (x_{cs} + \rho_s \cos(\phi_{ex}), y_{cs} + \rho_s \sin(\phi_{ex})) \\ T_N &= (x_{cf} + \rho_f \cos(\phi_{en}), y_{cf} + \rho_f \sin(\phi_{en})) \end{aligned} \quad (6.34)$$

图中的转弯机动都是顺时针转动外切圆弧。相似的,使用内切方式可以通过变换辅助圆使其半径为 $|\rho_f + \rho_s|$ 。 T_x 和 T_N 的坐标是生成 Dubins 曲线的关键[16]。

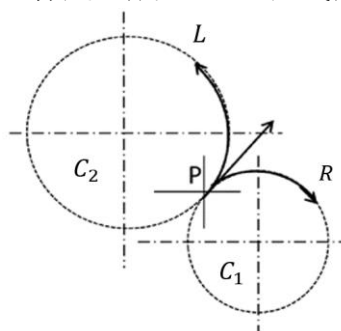


图 6-5 Dubins 曲线出发点

对于一个给定的姿态，有两个圆切于它。如图 3.6，姿态 P 右转在 C_1 圆上，左转在 C_2 圆上。如果 θ_s 和 θ_j 是固定的，在两种状态 P_s 和 P_j 间转移有 4 种可能，即 $\{RSR, RSL, LSR, LSL\}$ 。Dubins 曲线的长度表示为：

$$S_{Dubins} = S_s + S_t + S_f \quad (6.35)$$

$$= |\rho_s \alpha_s| + \|T_X T_N\| + |\rho_f \alpha_f|$$

$$S_{Dubins} = f(\rho_s, \rho_f) \quad (6.36)$$

其中, S_{Dubins} 是 Dubins 曲线的总长度, α_s 和 α_f 是开始与结束段的转弯角。并且

$$\|T_X T_N\| = \sqrt{(y_N - y_X)^2 + (x_N - x_X)^2} \quad (6.37)$$

通过以上长度公式, 每架无人机的相关速度可以计算出来, 从而控制合作任务计划的时间代价。

假定无人机存在最小转弯半径约束, 在当前的位置和方向, 要求到达一个离

当前位置较远的目标位置和方向。则依靠 Dubins 曲线的规划可以找到最佳的路线转移方式。Dubins 曲线在近距离下不一定有结果，且如果转移过程中有障碍，需考虑其它避障

方法与之结合，形成一个符合无人机行动约束的无人机路线规划，并且路线规划同时满足目标地点的位置和方向要求(如果有规划任务需求的话)。

Dubins 方法的几何机理可以广泛应用到各种任务的无人机航迹规划中，比如目标跟踪，目标覆盖搜索。Dubins 方法也常用于与各种算法的结合，以使得算法所规划的路径平滑可飞。Dubins 方法能较好的适应无人机机动能力的规划需求，而 RRT 算法能够适应复杂的环境，并且规划快速。本节通过分析 RRT 算法参数对路径选择及计算效率的影响，同时应用 Dubins 方法的四个转弯圆设为 RRT 算法中的虚拟障碍的方式，并且使用中值滤波优化曲线的方法，实现满足无人机机动性能的跨多重障碍的状态转移。

3.2.1 改进的理论与模型

1) Dubins 虚拟障碍的增加

考虑 RRT 的规划特性，利用前文中的连续模型，与 Dubins 曲线结合，基本思路增加约束无人机机动性能的虚拟障碍物[18-19]。在无人机的当前位置和目标位置的左右两边，都增加以无人机最小转弯半径为半径，以垂直于无人机方向上距无人机位置距离为 r_0 的位置为圆心的圆形障碍，限制无人机的初始转弯和最后转弯。

即对四个圆的圆心 $(x_{mi}, y_{mi}), i=1, 2, 3, 4$,

$$\begin{aligned} D_{pm} &= \text{isture}((x, y) \notin S_{pm}) \\ &= \prod_{i=1}^4 \text{isture}((x - x_{mi})^2 + (y - y_{mi})^2 > r_0^2) \end{aligned} \quad (6.38)$$

这样将 Dubins 曲线的结论，无人机状态转移的最短路线必定是四条曲线中的某条(如图 6-3)，融入 RRT 算法之中。同时在无人机穿过的障碍区建模时，将所有曲率半径小于 r_0 的障碍边尖角，都加上曲率半径为 r_0 的倒角。即对障碍尖角位置 (x_d, y_d) ，产生的伪障碍区 S_{pd} 。

$$\begin{aligned} D_{pd} &= \text{isture}((x, y) \notin S_{pd}) \\ &= \text{isture}((x - x_p)^2 + (y - y_p)^2 > r_0^2) \end{aligned} \quad (6.39)$$

再利用公式 6.38，将以上公式 6.39 纳入障碍区。另外，RRT 方法在开始和结束夹缝中，不利于随机取点的计算。先用几何方式将出发点和目标点引出，用引出后的点代替原出发点和目标点。得到 RRT 方法的折线后进行曲线优化。

2) 相应的曲线优化方法

路径平滑是固定翼无人机任务规划的重点之一。RRT 算法生成的某条路径是折线形的，转弯角度要求难以符合要求，不满足无人机机动性能。而对无规则的折线航迹的曲线优化，曲线优化可以通过在两航路点之间添加等距的较为密集的航路点对该方法进行适当拓展。但是，该类优化方法虽然最大限度地在不改变路径选择方向的情况下，优化了路径，使路程最短，但优化的路径后是具有明显大转弯的折线，且没有趋势缓冲的中间路线，无法直接用于无人机飞行航线。本节提出一种相邻粒子信息融合的迭代方法，通过相邻位置信息的不断同化迭代，使单个粒子与周围粒子形成的路径转折不再突兀，从而满足无人机的机动性能要求。

具体过程为，将 RRT 生成航路点为初始粒子，使用进化思想，下一代粒子在上一代相邻粒子的位置中间产生。对于存储有顺序路径的粒子点数组 M_j ， $(M_j = [x_j^1; x_j^2; x_j^3; \dots; x_j^n])$ ， M_j 表示第 j 代进化粒子数据，数据结构为 $n \times 2$ ，不包括起点 x_{init} 和终点 x_{aim} ；令：

$$\begin{cases} M_{aj} = [M_j; x_{aim}] \\ M_{bj} = [X_{init}; M_j] \end{cases} \quad (6.40)$$

那么下一代的粒子点数组为：

$$M_{j+1} = (M_{aj} + M_{bj})/2 \quad (6.41)$$

具体节点计算公式为：

$$\begin{cases} x_{j-1}^0 = x_{init} \\ x_{j-1}^{n+1} = x_{aim} \\ x_j^i = (x_{j-1}^{i-1} + x_{j-1}^i)/2 \end{cases} \quad (6.42)$$

经过多次重复式 6.41、6.42，可使在前后粒子点产生的路径(不包括在障碍边沿，或在出发点 and 目标点边上的粒子点)满足无人机的机动性能要求。

由于航线中存在障碍，每次迭代后，如果存在粒子点 x_j^i 有 $D_c(x_j^i) = 0$ ，即表示优化点 x_j^i 在障碍区内，则需进行处理。取上层两相邻粒子指向向量 d_{yc} ，设取垂直于 d_{yc} 的向量为 y_j^i 。其如下式：

$$\begin{cases} d_{yc} = (x_{j-1}^i - x_{j-1}^{i-1})/2 \\ y_j^i = d_{yc} \begin{bmatrix} \cos \frac{\pi}{2} & \sin \frac{\pi}{2} \\ -\sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{bmatrix} \end{cases} \quad (6.43)$$

则优化修改点 x_j^{i*} 为：

$$x_j^{i*} = x_j^i \pm \lambda y_j^i \quad (6.44)$$

其中， $\lambda \in [0.01 \ 0.05 \ 0.1 \ 0.5 \ 1]$ 。 λ 从小到大，将得到的结果代入公式 6.44，直到 $D_c(x_j^{i*}) = 1$ 。

对于拥有锐角的航迹，需要经过前处理，因为在公式 6.44 的实际处理中 λ 最大为 1，其垂直平方线上无法跨越锐角的障碍。为使粒子修改减少对转弯角度的影响，临近粒子点修改如下：

$$\begin{cases} x_j^{i-1*} = x_j^{i-1} \pm 0.5\lambda y_j^i \\ x_j^{i+1*} = x_j^{i+1} \pm 0.5\lambda y_j^i \end{cases} \quad (6.45)$$

优化几何过程中间向量表示 λy_j^i 的向量位移。同时随优化进行粒子数目增加。为加快计算效率，每优化若干次对前后距离不超过 0.3km。

粒子清除一次，保持粒子之间的距离相对稀疏，加快计算速率。最后留下的粒子，不改变原来的路径选择，优化了曲线，缩短了规划的飞行长度。该曲线优化方法本质上是一种附加约束的中值滤波方法。

Dubins 与 RRT 结合的混合方法分析：

RRT 方法能快速提供一条突防或拦截航迹，通过对其路径选择因素进行分析，分析了包括启发概率、步长与搜索空间等参数对路径选取的影响，针对不同的规划环境，选择合适的参数有利于 RRT 方法选择最短的路径。并且，本文算法针对无人机的初始和结束任务要求，使用 RRT 方法与 Dubins 结合，并进行曲线优化后，使原有折线状航线转化为可以实际飞行的平滑曲线，满足无人机最小转弯半径的航线要求，能够实现无人机拦截和突防的状态转移。

6.3 模型求解分析（第一小问）

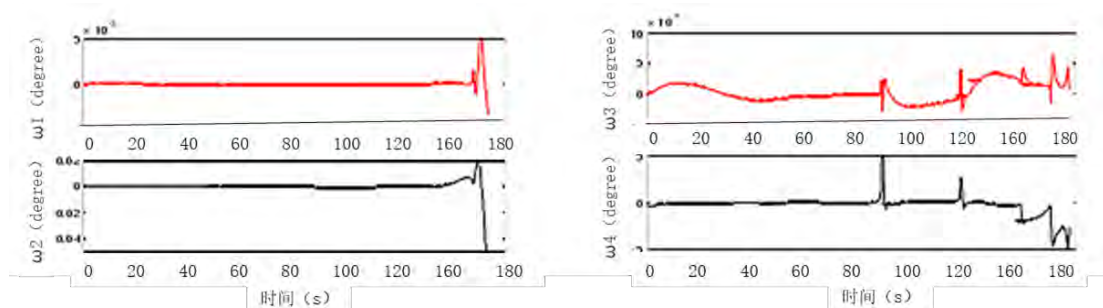
表 6-2 初始状态设置

名称	x (Km)	y (Km)	v (m/s)
蓝方初值	0	35	250
红方一第一波初值	50	y_{G1}	200
红方一运载机（第二波）初值	50	y_{G1}	300
红方二第一波初值	50	y_{G2}	200
红方二运载机（第二波）初值	50	y_{G2}	300

根据问题二以及上述模型方法，得到红方无人机第一波投放初始值 $G1$ ， $G2$ 坐标分别为 $(50,55.56)$ ， $(50,14.44)$ 。当红蓝方双方都采用微分制导策略进行博弈时，红方根据突防算法计算抛洒时刻，在两方并未交汇（红蓝双方的横坐标相同）之前，红蓝双方到达对抗区域同时满足的无人机集群间距上的约束和运载机与蓝方距离约束，运载机在尽可能靠近蓝方的位置进行二次抛洒。表 6-3 对其中一组仿真结果进行展示。

表 6-3 红方无人机最优发射策略

编号	第一波投放初始坐标	第一波无人机个数	第二波发射时刻	第二波发射时运载机位置	第二波发射时无人机群位置	拦截时刻
FY01	$(50,55.9)$	3	130.23	$(21.09, 62.27)$	$(20.72, 68.56)$	-
FY02	$(50,14.1)$	3	124.62	$(22.94, 50.35)$	$(21.09, 49.73)$	180.36



FY01 两无人机群中心偏角

FY02 两无人机群中心偏角

图 6-6 最优发射状态下红方两无人机群方向变化

对仿真结果进行分析，在红方无人机的第一波投放点坐标给定的条件下，红方运载机在初始位置先对三台无人机进行小规模抛洒，此时将未抛洒的无人机群中心看做运载机中心重合，由于运载机速度较快，未抛洒的无人机很快与第一批拉开距离。此时，红方无人机群运动方向不与 X 轴平行，而是以较小的速度偏角向蓝方夹进，红蓝双方相向而行，待到了对抗区域范围，即红蓝双方的相对距离到达一定值时，红方无人机群速度方向有了明显变化，为了缩小 Y 轴相对距离，且速度变化随着蓝方运动信息变化，直至 FY02 编号的无人机群对蓝方进行了捕获。

图 6-6 速度方向的变化曲线，在拦截开始时为了尽快缩短相对距离，速度变化较小。在拦截过程中为了尽可能的靠近目标点有着连续的、较为激烈的方向变化，说明红蓝双方进行了一定的交锋。在拦截末段将速度转率迅速减小到零附近，说明其追捕决策能较为准时准确的协调红方运动。

6.4 鞍点存在性判断与计算（第二小问）

6.4.1 鞍点的必要条件

与固定逗留期和无限时域微分对策不同，生存型微分对策的支付函数一般不是对策轨迹的连续泛函，其鞍点的存在性通常也难以判断。本文追踪无人机的速度大于逃逸无人机，则可以保证追踪无人机能在有限时间内对逃逸无人机进行捕获，因而该无人机追逃生存型微分对策对策的鞍点存在。下面根据微分对策最优性条件推导上述对策鞍点满足的必要条件。与追逃界栅构造过程类似，构造哈密顿函数和终端约束函数如下：

$$H(\mathbf{X}, \mathbf{u}_p, \mathbf{u}_E, \boldsymbol{\lambda}) = \boldsymbol{\lambda}^T \dot{\mathbf{X}} = \boldsymbol{\lambda}^T [\mathbf{A}\mathbf{X} + \mathbf{B}(\mathbf{T}_E - \mathbf{T}_p)] \quad (6.46)$$

$$\begin{aligned} &= \lambda_1 \dot{x} + \lambda_2 \dot{y} + \lambda_3 \dot{z} \\ &+ \lambda_4 (3\omega^2 x + 2\omega \dot{y} + T_E \cos \alpha_E \cos \beta_E - T_p \cos \alpha_p \cos \beta_p) \\ &+ \lambda_5 (-2\omega \dot{x} + T_E \sin \alpha_E \cos \beta_E - T_p \sin \alpha_p \cos \beta_p) \\ &+ \lambda_6 (-\omega^2 z + T_E \sin \beta_E - T_p \sin \beta_p) \end{aligned} \quad (6.47)$$

$$\Phi = t_f + \mathbf{v}^T \boldsymbol{\varphi} = t_f + v_1 x(t_f) + v_2 y(t_f) + v_3 z(t_f) \quad (6.48)$$

式中 $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6]^T$ 为协态变量， $\mathbf{v} = [v_1, v_2, v_3]^T$ 为终端约束对应的拉格朗日乘子。微分对策协态方程为：

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{X}} = -\mathbf{A}^T \boldsymbol{\lambda} \quad (6.49)$$

由式(2.35)可得，协态变量的末值条件为：

$$\boldsymbol{\lambda}(t_f) = \frac{\partial \Phi(t_f, \mathbf{x}(t_f))}{\partial \mathbf{x}(t_f)} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.50)$$

由式(4.52)可知，协态变量的演化与状态量无关，可将其写成状态转移的形式如下：

$$\boldsymbol{\lambda}(t) = \Phi_{\lambda}(t, t_0) \boldsymbol{\lambda}(t_0) \quad (6.51)$$

$$\boldsymbol{\lambda}(t) = \Phi_{\lambda}^{-1}(t_f, t) \boldsymbol{\lambda}(t_f) \quad (6.52)$$

式中 $\Phi_{\lambda}(t, t_0)$ 为状态转移矩阵，其解析形式为公式 6.48。

对式(6.50)，分别令 $\frac{\partial H}{\partial \mathbf{u}_p} = 0$ ， $\frac{\partial^2 H}{\partial \mathbf{u}_p^2} \geq 0$ 和 $\frac{\partial H}{\partial \mathbf{u}_E} = 0$ ， $\frac{\partial^2 H}{\partial \mathbf{u}_E^2} \leq 0$ ，可得追踪无人机和逃逸无人机的最优控制策略分别为：

$$\left\{ \begin{array}{l} \sin \alpha^*_P = \sin \alpha^*_E = \frac{\lambda_5}{\sqrt{\lambda_4^2 + \lambda_5^2}} \\ \cos \alpha^*_P = \cos \alpha^*_E = \frac{\lambda_4}{\sqrt{\lambda_4^2 + \lambda_5^2}} \\ \sin \beta^*_P = \sin \beta^*_E = \frac{\lambda_6}{\sqrt{\lambda_4^2 + \lambda_5^2 + \lambda_6^2}} \end{array} \right. \quad (6.53)$$

上式表明，追踪无人机和逃逸无人机的最优控制策略是相同的。
可得微分对策的横截条件为

$$H(t_f) = -\frac{\partial}{\partial t} \Phi(t_f, \mathbf{x}(t_f)) = -1 \quad (6.54)$$

该横截条件可写为

$$v_1 \dot{x}(t_f) + v_2 \dot{y}(t_f) + v_3 \dot{z}(t_f) = -1 \quad (6.55)$$

由上述鞍点必要条件可知，式(4.47)、(4.52)和式(4.53)、(4.58)共同构成了一个 7 维的两点边值问题，通过求解该两点边值问题即可获得对策的鞍点，需要注意的是这样求解得到的鞍点是生存型微分对策的开环鞍点，下面给出求解方法。

6.4.2 差分进化算法求解

对上一小节中构造得到的两点边值问题，其本质上是求解一个 7 维的非线性方程组其中还包含状态微分方程和协态微分方程，因此直接求解将较为困难。但结合式(6.50)和(6.53)可以看出，该 7 维非线性方程组实际上只含有 4 个独立变量，分别为追逃捕获时间 t_f 和终端约束对应的拉格朗日乘子 $\mathbf{v} = [v_1, v_2, v_3]^T$ 。相应的 4 个等式约束条件则为式(6.48)表示的追逃双方终端相对位置约束和对策哈密顿函数终端横截条件。因此，原追逃微分对策开环鞍点的求解最终转化为求解一个 4 维的非线性方程组。常用牛顿迭代法对该类方程组进行求解，但由于该方法需要给出较好的初值，否则将难以获得可行解，因此本文提出差分进化算法和牛顿迭代法相结合的两步串行求解策略，来提高求解速度和求解精度。首先以差分进化算法优化得到的解作为初解，再通过牛顿法迭代得到严格满足终端约束的解，从而对开环鞍点进行快速高精度求解。

采用差分进化算法获得初始解：

由分析可知，当给定追逃终端时间 t_f 及终端约束对应的拉格朗日乘子 $\mathbf{v} = [v_1, v_2, v_3]^T$ 后，可先由式(6.57)和(6.59)求得对策的初始协态变量，从而可以进一步根据式(4.57)求得追逃双方的控制量，然后对式(6.47)进行积分，得到更新后的追逃双方状态量，并根据式(4.54)求得更新后的协态变量。不断更新计算至追逃终端时刻 t_f ，即可得到追踪器和逃逸器的终端相对位置和 Hamilton 量，由此可以计算出其终端约束的脱靶量，当脱靶量为零时，所得即为追逃微分对策的开环最优策略。

本文首先采用差分进化算法优化得到最优策略的粗解，优化模型的设计变量为：

$$X_{DE} = [t_f, v_1, v_2, v_3]^T \quad (6.56)$$

目标函数为：

$$J_{DE} = \sum_{i=1}^4 k_i c_i^2 \quad (6.57)$$

式中 c_i 为终端等式约束的脱靶量, k_i 为加权系数。

差分进化算法是一种基于群体差异的启发式随机搜索算法,最早是由 Storn 和 Price 为求解切比雪夫多项式而提出的[11]。DE 算法通过群体内个体间合作与竞争产生的群体智能指导优化搜索,受控参数少、鲁棒性强、算法收敛性好。同时,采用 DE 算法对协态变量进行初值求解时,由于该算法的具有良好的全局搜索性能,因此可以较好地避免求解过程陷入局部最优。但另一方面,该算法在计算时需要多次对群体中个体进行适应度的评价,这带来了一定的计算量。针对该问题,本文在使用 DE 算法优化求解时将采用较大的积分步长和较低的积分精度,以此来有效降低计算时间。

6.4.3 通道带宽上限求解分析

根据数值计算得到,鞍点对应的通道带宽 M 的上限 M_{\max} 为 108.8Km。当通道带宽大于 108.8Km 时,尽管红方采用最优拦截策略,蓝方也能够顺利逃脱。与前两问中的带宽值相比较,可以看出明显通道带宽越大,蓝方无人机越容易逃脱。与第二问中的通道下限值相比较可以看出,该通道带宽明显数值更大,因为本问中运载机采用的两次抛洒策略且无人机个数有所增加。

7. 问题四

7.1 问题分析

考虑红蓝双方的作战场景:红方发现蓝方 3 架无人机来袭后,红方派出 5 架运载机各携带 10 架无人机对蓝方无人机进行空中拦截,双方最终进入近距格斗。为分析集群动态对抗任务,定义如下图所示集群对抗场景,其中作战区域不考虑空中障碍物等任何外部特征。红蓝作战双方均拥有一定数量的无人机,红方具有一个由无人机集群保护的边界线,可视为高价值目标。红方边界线位于作战区域一端,除了各自出动的无人机集群外,不具有其它防御或进攻能力。同时,各方无人机为同一类型的固定翼无人机,具有完全相同的平台性能。拦截方红方无人机威胁范围为定值,双方无人机能够正确识别敌我战机,通过对实时态势的感知以及与邻居友机的交互进行在线自主决策,实现集群的自组织作战。

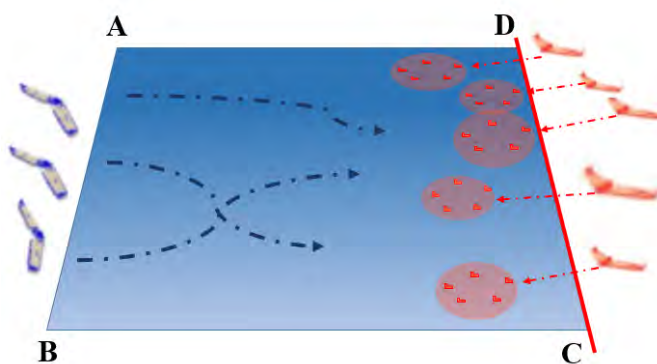


图 7-1 无人机集群协同对抗场景

当作战开始时,所有蓝方无人机构成一个有序的整体同时朝着红方边界线进攻,红方运载机第一波次发射无人机的时刻为初始对抗时刻,且发射的无人机集群几何构型圆周中心的距离为 443m。在进攻过程中,遇到敌方来袭无人机时,双方集群陷入混战。集群的作战目标包括:

- 1) 蓝方希望尽可能多的无人机突防成功。
- 2) 红方则希望成功拦截尽可能多的蓝方无人机。

这两个作战目标分别体现了集群的攻击和防御偏好。以此来制定红方最优拦截策略和蓝方最优突防策略。

7.2 无人机集群协同对抗模型

7.2.1 协同对抗任务规划流程

多 UAV 之间通过信息交换协同对目标实施作战任务分配,有效提高空战效率. 多 UAV 协同空战决策技术成为 UAV 应用领域的一项关键技术。无人机协同多任务规划为多无人机协同执行多战术任务提供合理高效的任务计划,是多机协同的基础和保证。协同多任务规划除了需要考虑任务本身的构成要素如任务兵力、目标分布、任务类型等的要求之外,还必须考虑多机之间协调一致共同执行任务的约束关系,根据战术任务需求和任务规划的优化指标为各无人机制定出协同的任务计划,使其整体作战效能最优或近似最优。多 UAV 协同空战决策主要涉及空战态势评估、多目标分配、协同作战机动决策三个方面 [16]。态势评估是多 UAV 协同空战的前提,目标分配的合理性直接关系到 UAV 的作战效率及整个空战结果,要求我方机群在最小的不利态势下获得对敌方机群尽可能大的攻击优势。完成态势评估和目标分配后,需要 UAV 采取如平飞、加力转弯、爬升等机动决策以避免敌机潜在威胁或取得更大的空战对抗优势 [17]。

本文考虑双方具有多架 UAV 空战对抗时的情形,采用影响图法 [18] 建立了多 UAV 的协同作战决策模型,使用贝叶斯推论 [19] 对空战态势进行评估,运用匈牙利算法 [20] 进行目标分配,将多对多空战转化为多个动态的一对一的对抗形式,通过仿真表明了本文给出的多 UAV 协同空战模型的有效性。

在集群动态对抗过程中,双方 UAV 的位置、数量等状态在不断变化,因此 UAV 应根据实时态势,进行在线任务决策,包括目标分配决策、运动决策和攻击决策。红方对蓝方的拦截决策会导致双方 UAV 的剩余无人机数量发生改变;而蓝方对红方的突防决策会导致红方 UAV 投放位置和拦截成功概率发生改变。因此,为研究集群动态对抗任务,首先应建立动态对抗模型,实现对 UAV 和位置状态的实时更新,得到实时态势作为 UAV 在线决策的基础。

无人机集群对抗是一个复杂的动态随机过程,空战态势随着时空不断演变,因此每架 UAV 必须根据实时态势进行在线任务决策,实现与己方个体之间的合作,以及与敌方的博弈。多无人机协同目标匹配决策最基础的是:(1)要完成对当前空战对抗态势信息的统计,(2)评估当前态势信息,(3)根据以往历史经验进行威胁评估,攻击排序,(4)根据攻击排列顺序和我方攻击数值对比信息进行匹配等。以上四个步骤是实现多无人机空中对抗目标匹配的核心步骤。其中对于一个机群内部的而言,每个无人机的单一自主优先权在多无人机协同作战体系中是首先要评估其在当前态势信息下的相对于每一个对手的对位匹配参数,只有匹配参数的确定,多无人机协同智能作战体系包含的决策算法才能针对对抗双方的自主优先权参数数值进行目标匹配,已达到在当前状态下的一个最优的目标匹配结果。

本章针对无人机集群动态对抗自主任务规划问题,以多智能体系统仿真建模技术为基础,建立大规模集群对抗系统。将集群中的每架 UAV 视为一个独立智能体,能够基于自身状态以及与邻居 UAV 的信息交互,进行自主任务决策。同时,通过借鉴生物群集行为的自组织机制,为 UAV 个体设计行为规则集,通过每个个体对其邻域环境的作用,实现大规模无人机集群的行为协调,使系统整体涌现出宏观的作战效果。每架 UAV 的作战流程如下图 7-2 所示。在每一决策步,UAV 首先进行态势评估,根据评估结果计算各目标的攻击收益,并以最大化攻击总收益为目标,进行目标分配决策。然后各 UAV 根据目标分配结果,选择相应的行为规则:若 UAV 所分配目标为蓝方无人机,即采取攻击行为;若目标为红方无人机,即采取防御躲避行为,根据所选择的行为规则,各 UAV 更新自身速度和位置,接近或躲避所分配目标。随后红方 UAV 针对所分配目标,判断是否满足攻击条件,若

满足，则攻击目标并更新蓝方 UAV 的生存概率和武器数量。最后判断作战是否结束，若未结束则进行下一轮决策，结束的条件是蓝方的 UAV 全部被摧毁或者蓝方剩余 UAV 都以突破红方边界线或者时间超过任务要求的 $360s$ 。通过对上述目标分配决策、集群运动决策以及目标攻击决策这几个关键步骤的不断循环迭代，可有效实现无人机集群之间的动态对抗。

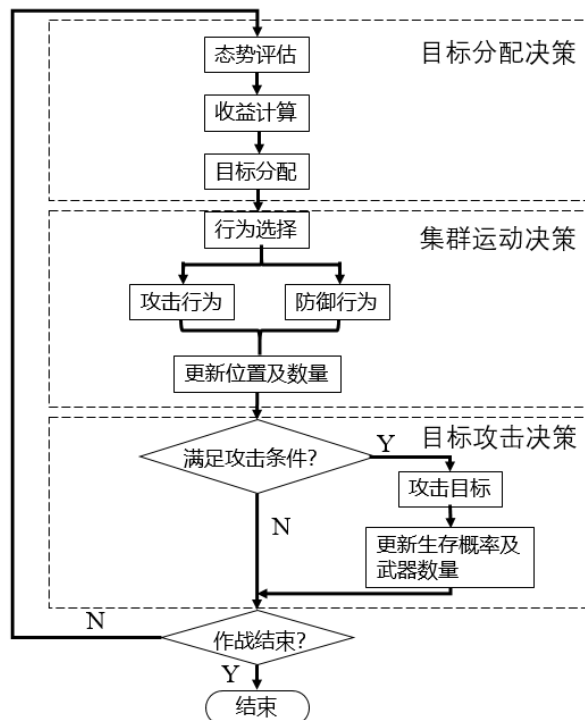


图 7-2 无人机集群动态对抗任务规划流程图

7.2.2 空战态势评估

态势评估是进行空战决策的前提，直接影响到集群最终生成的作战方案，是取得作战胜利的关键因素。基于所建立的对抗场景，敌我双方的作战力量包括各自的无人机集群以及运载机，因此态势评估主要包括无人机之间以及无人机与运载机之间的态势估计。

(1) 无人机之间的态势评估

某一时刻空战双方无人机态势关系如下图 7-3 所示，其中 D_{ij} 为双机距离， α_{ij} 为红方 UAV 对蓝方无人机的方位角，即双机目标线与我方 UAV 速度矢量 v_i 的夹角，以相对双机目标线右偏为正，且满足 $0 \leq |\alpha_{ij}| \leq \pi$ 。我方 UAV 对敌机的态势优势可采用角度优势指数、距离优势指数进行评估，分别记为 Sa_{ij}, Sr_{ij} 。

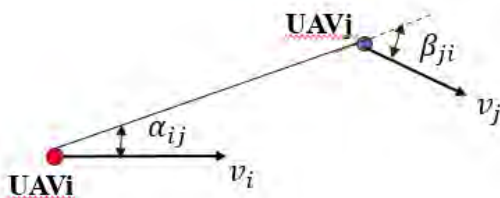


图 7-3 无人机之间态势示意图

(1) 角度优势指数主要考虑红方 UAV 的方位角，设计为：

$$Sa_{ij} = \begin{cases} 1 - \frac{2|a_{ij}|}{\pi} & 0 \leq |\alpha_{ij}| \leq \frac{\pi}{2} \\ 0 & \frac{\pi}{2} \leq |\alpha_{ij}| \leq \pi \end{cases} \quad (7.1)$$

由上式 7.1 可知，我方 UAV 与双机目标线的夹角越小，则角度优势越大，即我方 UAV 对目标的方位角越小，则能够更快地调整运动方向进攻或躲避。

(2) 距离优势指数与红蓝双方 UAV 间的距离、红方运载机所携带的无人机最大攻击范围有关，设计为：

$$Sa_{ij} = \begin{cases} r_a + r_k \\ 100m \leq r_a \leq 10000m \end{cases} \quad (7.2)$$

式中，为我红方 UAV 所携带无人机投放的距离， r_a 为红方运载机与蓝方无人机的距离， r_k 为红方 UAV 集群无人机之间的距离。从上式可知，红方 UAV 距离蓝方 UAV 越近，则距离优势越大，而蓝方正好相反。

综上，两方第 i 架 UAV 对敌方第 j 架 UAV 的综合优势指数为 S_{ij} 为两项优势指数基于重要程度的线性加权：

$$\begin{aligned} \text{红方: } S_{ij} &= \omega_1 \cdot Sa_{ij} + \omega_2 \cdot Sr_{ij} \quad (0 \leq i \leq 5, 0 \leq j \leq 3) \\ \text{蓝方: } S_{ij} &= \omega_1 \cdot Sa_{ij} - \omega_2 \cdot Sr_{ij} \quad (0 \leq i \leq 3, 0 \leq j \leq 5) \end{aligned} \quad (7.3)$$

其中， $\vec{\omega} = [\omega_1, \omega_2]$ 为优势评估权向量， ω_1, ω_2 分别为角度和距离的优势指数权值，且满足 $\omega_1 + \omega_2 = 1$ ， $\omega_i > 0$ 。将模型中的敌我参数互换，则可得到敌机对我机的态势威胁指数 T_{ji} 。

7.2.3 目标分配优化模型决策

目标分配是指根据敌我双方态势，设计分配策略科学合理地分配我方 UAV，使得对敌方攻击效果达到最佳或使己方得到最好的保护[21]。由于本章考虑的是无人机集群之间的实时对抗决策，且集群通常具有较大的规模，因此设计实时性较好的目标分配算法显得尤为重要。分布式一致性拍卖算法（CBAA）求解过程简单且计算量小，适用于求解大规模目标分配问题，因此本章采用 CBAA 算法并对其进行改进，设计基于扩展 ICBA 算法的目标分配决策方法[22]。

根据所设计的空战态势评估方法，能够得到某一时刻我机对敌机的态势优势矩阵 S 。同时综合考虑敌方目标的生存概率和价值，设计 X 方第 i 架 UAV 对 Y 方第 j 架 UAV 的攻击收益计算方法为：

$$c_{ij}^{XY} = [\alpha S_{ij}^{XY} + (1 - \alpha) p_j^Y] v_j^Y \quad (7.4)$$

其中， p_j^Y 、 v_j^Y 分别为 Y 方第 j 架 UAV 的生存概率和价值。 α 为权重系数， α 越大越倾向于攻击价值优势较大的目标，而 α 越小，表明 UAV 更倾向于攻击生存概率更大的目标。

相应地：蓝方第 i 架 UAV 对红方边界线攻击收益计算方法为：

$$c_{ib}^{XY} = [\alpha S_{ib}^{XY} + (1 - \alpha) p_b^Y] v_b^Y \quad (7.5)$$

上式中， p_b^Y 、 v_b^Y 分别为红方生存概率和价值。

红方无人机集群协同多目标分配以最大化攻击总收益为目标，包括攻击敌机的收益

和拖延敌机突防时间。记 x_{ij}^{XY} , x_{ib}^{XY} 为决策变量, x_{ij}^{XY} , x_{ib}^{XY} 分别表示 X 方第 i 架 UAV 被分配攻击 Y 方第 j 架 UAV 或边界, 而 $x_{ij}^{XY}, x_{ib}^{XY} = 0$ 表示未被分配相应目标, 建立目标分配优化模型如下:

$$\begin{aligned} \max \quad & \sum_{i=1}^{N_x} \left(\sum_{j=1}^{N_y} \left[\alpha S_{ij}^{XY} + (1-\alpha) p_j^Y \right] \cdot v_j^Y \cdot x_{ij}^{XY} + f \left[\alpha_{ib}^{XY} + (1-\alpha) p_b^Y \right] \cdot v_b^Y \cdot x_{ib}^{XY} \right) \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^{N_x} x_{ij}^{XY} \leq 1 \\ \sum_{j=1}^{N_y} x_{ij}^{XY} \leq \min \{1, w_i^x\} \\ x_{ij}^{XY}, x_{ib}^{XY} \in \{0, 1\} \end{cases} \end{aligned} \quad (7.6)$$

式中, f 为攻防偏好因子。 f 越大, 表明 UAV 更倾向于攻击敌方无人机或突防 (即偏好攻); 而 f 越小, 表明 UAV 更倾向于保护己方无人机或躲避 (即偏好防)。因此, 通过设置 f 的取值, 能够调整集群的攻防作战策略。此外, 第 1 个约束条件代表每架敌机最多只能分配给 1 架 UAV; 第 2 个约束条件表示第 i 架 UAV 分配的目标数量小于等于 1 且不得超过当前剩余无人机数量。

7.2.4 基于扩展 ICBA 算法的目标分配模型求解

在求解目标分配问题时, 是对整个集群的目标函数进行优化, 本质上还是一种集中式的求解方法。若采用这些算法对每架 UAV 的目标函数进行单独优化, 则无法消除 UAV 之间的分配冲突[23]。而在拍卖算法等基于市场机制的分布式目标分配算法中, 各 UAV 无法获知所有目标的信息, 只能竞标自身所获知的目标, 并通过与邻居的信息交互消除分配冲突, 获得统一的目标分配方案。这种分布式目标分配算法能够有效提高算法效率, 并支持 UAV 的动态加入和退出, 适用于大规模集群动态对抗这一作战场景。因此本章采用分布式迭代一致性拍卖算法 (ICBA), 用于实现 UAV 对敌机的一对一分配, 同时考虑到 UAV 对对方无人机允许多对一分配, 对 ICBA 算法进行了扩展, 设计了扩展 ICBA 算法用于求解所设计集群对抗场景中的目标分配问题。ICBA 算法在 CBA 算法的基础上进行多轮竞拍, 即对 CBA 算法进行多次迭代, 以寻求总收益更大的目标分配方案。在每一轮竞拍开始时, 先根据 UAV 对各目标的攻击收益大小, 采用轮盘赌法初始化目标分配方案, 然后基于 CBA 算法对拍卖阶段和一致性阶段不断循环迭代, 生成本轮竞拍的最终方案。在拍卖阶段, 每架 UAV 首先判断自身是否已分配目标, 若未分配, 则按一定规则进行竞标, 若已分配, 则跳过该阶段; 在一致性阶段, UAV 通过通信网络传递竞标信息, 并根据一致性无冲突法则, 消除目标分配冲突。

假设敌方 UAV 数量为 N_t , 首先定义我方第 i 架 UAV 的分配列表 x_i 和收益列表 y_i , 长度均 N_t 。在 x_i 中, 若第 j 架敌机分配给了第 i 架 UAV, 则 $x_{ij} = 1$, 否则 $x_{ij} = 0$ 。在 y_i 中, y_{ij} 为当前第 i 架 UAV 所获知的竞标第 j 架敌机的最大收益。考虑到集群的攻击目标还包括边界线, 将两个列表长度扩展为 $N_t + 1$, 最后一列存储对敌方无人机的分配结果和攻击收益。在每一次迭代中, 每架 UAV 对自身所维护的分配列表和收益列表进行更新。下面以某一轮竞拍中, 第 i 架 UAV 的第 t 次迭代为例, 详细介绍扩展 ICBA 算法中的拍卖阶段和一致性阶段。

(1) 拍卖阶段

在该阶段, 每架 UAV 首先判断自身是否已分配目标, 若已分配目标即 $x_i(t) \neq 0$, 则跳过该阶段, 否则, 判断有效目标列表 h_i :

$$\begin{cases} h_i = \Theta(c_{ij} > y_{ij}(t)), j=1,2,\dots,N_t \\ h_{i,N_t+1} = 1 \end{cases} \quad (7.7)$$

式中， $\Theta(\cdot)$ 为指示函数，当括号中的内容为真时，其值为1，否则为0。 $h_{ij}=1$ 代表第*i*架 UAV 可竞选第*j*个目标。若第*i*架 UAV 攻击第*j*架敌机的收益大于当前所获知的竞选敌机*j*的友机的最大攻击收益，则第*j*架敌机为有效目标。同时，由于蓝方无人机数量少，是一个高价值目标，可被分配给多架 UAV，即无论其是否已被分配给其它 UAV，均为有效目标，因此 h_{ij,N_t+1} 始终等于1。确定有效目标列表后，从中选择收益最大的目标 j^* ：

$$j^* = \arg \max_j h_{ij} \cdot c_{ij}, j=1,2,\dots,N_t+1 \quad (7.8)$$

若 UAV 对目标 j^* 的攻击收益 $c_{ij^*} \neq 0$ ，则更新其分配列表 $x_{ij^*}(t)=1$ 及收益列表 $y_{ij^*}(t)=c_{ij^*}$ ，即将目标 j^* 分配给第*i*架 UAV，并将相应的收益存入收益列表 y_i ；若 $c_{ij^*}=0$ ，则意味着 UAV 对所有有效目标的收益均为0，此时将边界线分配给 UAV，并更新分配列表 $x_{i,N_t+1}(t)=1$ 。该阶段的算法流程如下表 7-1 所示。所有 UAV 完成竞标后，进入一致性阶段。

表 7-1 拍卖阶段算法流程

<i>Phase 1: Auction Process for UAV i at iteration t</i>	
1:	<i>Predure SELECT TARGET</i> ($c_i, x_i(t-1), y_i(t-1)$)
2:	$x_i(t) = x_i(t-1)$
3:	$y_i(t) = y_i(t-1)$
4:	<i>if</i> $x_i(t) = 0$ <i>then</i>
5:	$h_{ij} = \Theta(c_{ij} > y_{ij}(t)), j=1,2,\dots, N_t$
6:	$h_{i,N_t+1} = 1$
7:	$j^* = \arg \max_j h_{ij} \cdot c_{ij}, j=1,2,\dots, N_t+1$
8:	<i>if</i> $c_{ij^*} \neq 0$ <i>then</i>
9:	$x_{ij^*}(t) = 1, y_{ij^*}(t) = c_{ij^*}$
10:	<i>else</i>
11:	$x_{i,N_t+1}(t) = 1$
12:	<i>end if</i>
13:	<i>end if</i>
14:	<i>end procedure</i>

(2) 一致性阶段

该阶段利用一致性准则，消除 UAV 之间的目标分配冲突，使所有 UAV 的竞标信息收敛到一个统一的分配决策方案。由于蓝方无人机可以被分配给多架红方 UAV，因此一致性阶段仅用于消除对敌机的分配冲突问题，以防止同一架敌机被分配给多架 UAV，造成资源的浪费。

在一致性阶段，首先第*i*架 UAV 接收所有邻居无人机*k*的竞标信息 y_k ，然后根据所接收到的竞标信息以及自身的竞标信息，对自身的分配列表 x_i 和收益列表 y_i 进行更新，消除分配冲突。其中， $y_{ij}(t)$ 更新为所有邻居无人机*k*所保存的 $y_{kj}(t)$ 以及自身保存的 $y_{ij}(t)$ 中

的最大值，即：

$$y_{ij}(t) = \max y_{kj}(t), j = 1, 2, \dots, N_t \quad (7.9)$$

同时，得到攻击目标 j^* 收益最大的 UAV：

$$z_{ij^*} = \arg \max y_{kj^*}(t) \quad (7.10)$$

若 $z_{ij^*} \neq i$ ，则 $x_{ij^*}(t) = 0$ ，即若第 i 架 UAV 发现目标 j^* 被分配给多架 UAV，且存在其它 UAV 对其收益更大，则将该目标从自身分配列表中删除。该阶段的算法流程如表 4.2 所示。

表 7-2 一致性阶段算法流程

<i>Phase 2: Consensus Process for UAV i at iteration t</i>	
1:	Send y_t to k with $g_{ki}(\tau) = 1$
2:	Receive y_k from k with $g_{ik}(\tau) = 1$
3:	Procedure UPDATE TARGET $\left(g_i(\tau), y_{k \in \{k g_{ik}(\tau) = 1\}}(t), j^*\right)$
4:	$y_{ij}(t) = \max_k g_{ik}(\tau) \cdot y_{kj}(t), j = 1, 2, \dots, N_t$
5:	$Z_{ij^*} = \arg \max_k g_{ik}(\tau) \cdot y_{kj^*}(t)$
6:	if $Z_{ij^*} \neq i$ then
7:	$x_{ij^*}(t) = 0$
8:	end if
9:	end procedure

上述拍卖和一致性过程不断迭代，直到所有 UAV 的收益列表均不再发生变化，视为本轮竞拍结束，得到本轮目标分配方案及相应的总收益。若总收益大于之前所存储的最优分配方案的总收益，则将最优分配方案替换为本轮方案。若时间允许，则进行下一轮拍卖，否则输出当前最优目标分配方案作为最终分配结果。

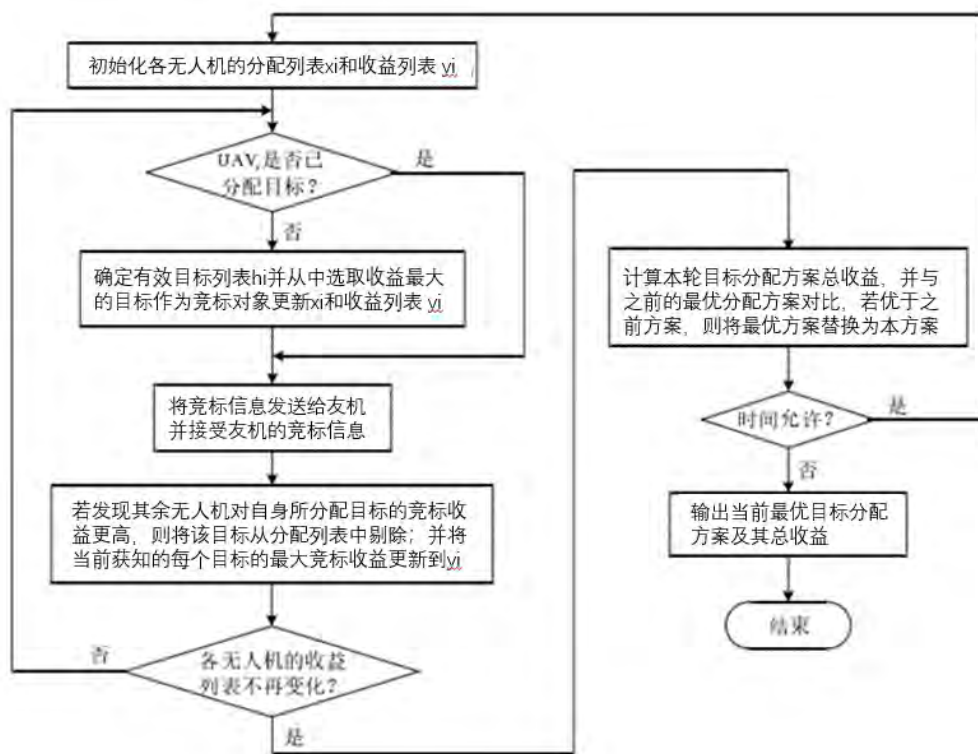
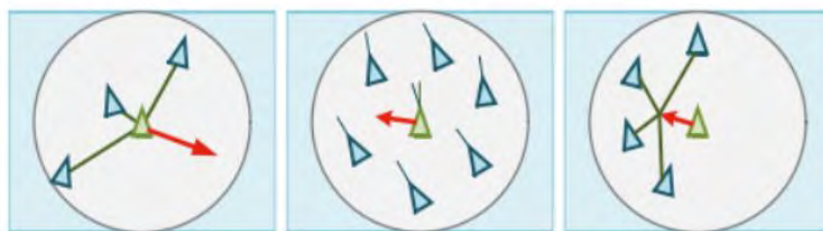


图 7-4 扩展 ICBA 算法流程图

7.2.5 基于 SAC-OD 规则的集群运动决策

根据目标分配结果，无人机集群应进行运动决策，朝各自所分配的目标运动。所有分配目标为蓝方无人机的红方 UAV 应构成一个有序整体朝目标运动，可视为组群攻击行为；而分配目标为敌机的 UAV 应脱离集群朝所分配敌机运动，可视为应激防御行为。无人机集群的运动可借鉴自然界鸟群、鱼群等生物群集运动，在生物群体中，虽然个体的感知和行动能力有限，但通过遵循简单的行为规则，却能够协作完成复杂的群体活动[24]。因此，本章通过借鉴生物群集的自组织协调机制，在经典生物集群行为规则“避撞-结队-聚集”（SAC）的基础上，引入 UAV 个体的攻击和防御行为，设计其作战行为规则集，并基于集群动态拓扑交互机制，建立有限感知条件下的集群运动模型，使集群涌现出宏观的作战行为。

群集运动模型—Boid 模型中，个体遵循避撞(Separation)、结队(Alignment)和聚集(Cohesion)三条基本行为规则，称为 SAC 规则，如下图 7-5 所示[25]。避撞和聚集行为用于实现个体之间的位置协同，遵循“短距排斥、长距吸引”的原则，在避免个体发生碰撞的同时保持群体的空间聚集性；而结队行为强调个体与邻居保持速度一致。这三条行为规则共同作用，便可模拟出与生物群体相似的集体运动形式。



(a) 避撞 (b) 结队 (c) 聚集

图 7-5 “避撞、结队、聚集”行为规则示意图

在集群对抗任务中，UAV 的攻击对象分为敌机。若红方攻击目标为蓝方，即采取攻击行为；若蓝方攻击目标为红方，即采取防御行为。因此，为实现无人机集群的动态对抗，在经典集群行为规则“避撞-结队-聚集”的基础上，引入 UAV 个体的攻击和防御行为。由此设计 UAV 的作战行为规则集 $Action = \{A_1, A_2, A_3, A_4, A_5\}$ ，分别表示聚集、避撞、结队、攻击和防御行为，记为 SAC-OD 规则，并基于社会力为每一条行为规则设计相应的控制力，驱使 UAV 做出相应的行为。

(1) 规则一：靠近邻近 UAV（聚集）。

当 UAV 与邻居之间距离较远时，邻居可以对 UAV 产生吸引力，以驱使 UAV 朝邻居运动。UAV 所受到的引力为所有邻居产生的引力之和，设计为：

$$u_i^a = - \sum_{j \in N_i^C} g_a(d_{ij}) x_{ij} \quad (7.11)$$

式中， d_{ij} 为第 i 架 UAV 与第 j 架 UAV 之间的距离， $g_a(d_{ij})$ 为引力函数， x_{ij} 表示由第 j 架 UAV 指向第 i 架 UAV 的单位向量。由上式 7.11 可知，UAV 受到的引力由自身指向邻居，因此可驱使其朝邻居运动，实现聚集行为。

(2) 规则二：避免与邻近 UAV 发生碰撞（避撞）。

当 UAV 与邻居之间的距离过近时，邻居会对 UAV 产生排斥力，以驱使 UAV 远离过近的邻居。UAV 受到的斥力为所有邻居产生的斥力之和，设计为：

$$u_i^r = \sum_{j \in N_i^C} g_r(d_{ij}) x_{ij} \quad (7.12)$$

式中， $g_r(d_{ij})$ 为斥力函数。由上式 7.12 可知，UAV 受到的斥力由邻居指向自身，因此可驱使其远离距离过近的邻居，实现避撞行为。

规则一与规则二共同实现了 UAV 之间的位置协同，遵循“近距排斥-远距吸引”的原则，即当个体之间距离较近时互相产生斥力，而距离较远时产生引力。 $g_r(d_{ij})$ 和 $g_a(d_{ij})$ 一起可构成满足该原则的引力/斥力函数，常见形式如下：

$$\begin{aligned} g(d_{ij}) &= \nabla x_i \left(\sum_{j \neq i} (C_r \exp(-\frac{d_{ij}}{l_r}) - C_a \exp(-\frac{d_{ij}}{l_a})) \right) \\ g(d_{ij}) &= C_l \sum_{j \neq i} \left((\frac{d_{ij}}{l_c})^{-3} - (\frac{d_{ij}}{l_c})^{-2} \right) \exp(\frac{d_{ij}}{l_c}) \\ g(d_{ij}) &= \sum_{j \neq i} (d_{ij}^{-\gamma_a} - d_{ij}^{-\gamma_b}) \end{aligned} \quad (7.13)$$

式中，参数 l_r ， l_a ， l_c ， γ_a ， γ_b 用于调节个体之间引力/斥力的作用范围，参数 C_l ， C_a ， C_r 用于调节引力/斥力的大小。

(3) 规则三：与邻居 UAV 保持速度一致（结队）

结队规则描述了 UAV 之间的速度协同行为，通常采用速度平均机制来设计速度协同力，使 UAV 的速度朝着所有邻居的平均速度靠近：

$$u_i^v = -K^v (v_i - \frac{1}{N_i^C} \sum_{j \in N_i^C} v_j) \quad (7.14)$$

中， $K^v > 0$ 为速度协同力控制增益， $|N_i^C|$ 为第 i 架 UAV 的邻居数量。若令

$K^v = |N_i^C|$ ，则：

$$u_i^v = - \sum_{j \in N_i^C} (v_i - v_j) \quad (7.15)$$

此即常见的一致性算法。在速度协同力 u_i^v 的作用下，UAV 的速度会趋近于周围邻居的平均速度，从而实现集群结队行为。

(4) 规则四：接近敌方边界线（攻击）。

当 UAV 所分配的目标为敌方边界线时，会朝着敌方边界线进攻，该行为体现了 UAV 的攻击偏好，相应的攻击导航力可设计为边界线对 UAV 的引力：

$$u_i^a = K^a \cdot \frac{x_b - x_i}{\|x_b - x_i\|} \quad (7.16)$$

式中， $K^a > 0$ 为攻击导航力控制增益， x_b 为边界线的位置。在上式所设计导航力的作用下，UAV 能够朝着边界线运动。

(1) 规则五：接近敌方 UAV（防御）

当 UAV 所分配的目标为敌方 UAV 时，会朝着所分配敌机进攻，该行为体现了 UAV 的防御偏好，相应的防御导航力可设计为敌机对 UAV 的引力：

$$u_i^d = K^d \cdot \frac{x'_i - x_i}{\|x'_i - x_i\|} \quad (7.17)$$

式中， $K^d > 0$ 为防御导航力控制增益， x'_i 为当前所分配敌机的位置。在上式所设计导航力的作用下，UAV 能够所分配的敌机运动。

常用的集群运动建模方法有两种：一种是基于整体的宏观建模方法，其建模对象为集群整体，不考虑个体之间的交互；第二种是基于个体的微观建模方法，其建模对象为集群中的个体，便于分析个体的行为特性，并可同时考虑外部环境的影响，因此本章采用第二种方法[26]。同时基于所设计 UAV 个体的 SAC-OD 行为规则集，以及有限感知条件下的集群动态拓扑交互机制，建立基于 SAC-OD 规则的有限感知集群运动模型。

考虑一个包含架 UAV 的集群系统，在二维无障碍平面内运动。第 i 架 UAV 的位置和速度分别记为 $x_i, v_i \in R^2$ ，其运动方程为：

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = u_i \end{cases}, \quad i = 1, 2, \dots, N_v \quad (7.18)$$

式中， $u_i \in R^2$ 表示角度向量。

根据所设计的 SAC-OD 行为规则集，并考虑环境影响，设计个体运动协同控制律如下：

$$u_i = (1 - \alpha_i) [u_i^o + u_i^r + u_i^a + u_i^v] + \alpha_i (u_i^d + u_i^r) - \xi \|v_i\|^2 v_i + \eta \xi_i \quad (7.19)$$

式中， $-\xi \|v_i\|^2 v_i$ ， $\xi > 0$ 为阻尼系数； $\eta \xi_i$ 表示强度为 $\eta \geq 0$ 的随机噪声。 α_i 为 Boolean 值，用于标记个体采取攻击行为或是防御行为。若 UAV 分配的目标为同一目标，即采取集群攻击行为，则 $\alpha_i = 0$ ，此时 UAV 在攻击导航力、位置协同力和速度协同力的作用下，与其余目标同为一个目标的 UAV 构成一个有序的整体朝目标运动。而当 UAV 所分配目标为敌方 UAV 时，即采取应激防御行为，则 $\alpha_i = 1$ ，此时 UAV 在防御导航力的作用下脱离集群整体朝着所分配敌机运动，同时机间斥力项的存在可避免其与邻居 UAV 发生碰撞。

基于上式 7.19 所设计的集群运动模型，可实现对抗过程各 UAV 的组群攻击以及应激防御行为，并实时更新 UAV 的数量和位置，使集群涌现出宏观的动态对抗效果。

7.3 模型检验

本题采用空间态势评估、目标分配优化、集群运动决策分步骤进行模型建立求解，从

双方对抗的角度，蓝方希望尽可能多的无人机突防成功，红方则希望成功拦截尽可能多的蓝方无人机。求解双方的运动决策可以得出红方最优拦截策略和蓝方最优突防策略。为了检验模型有效性，使用 pathon 软件随机选取蓝方突防无人机初始位置和轨迹仿真红方对抗结果如下表 7-3 所示，随机选取红方运载机初始位置和拦截路线来仿真蓝方的突防结果如下表 7-4 所示。可以看出拦截和突防效果较好。。

表 7-3 红方最优拦截策略下的多无人机对抗规划结果

红方拦截第一次仿真方案结果						
运载机名称	初始位置	突防无人机初始位置	第一波发射无人机集群信息	第二波发射无人机集群信息	拦截目标	是否拦截成功
FY01	(50, 80)	HT01 (0,75)	集群数量：4 架 集群中心位置： (42.7, 73.8) 发射时刻：31.9s	集群数量：6 架 集群中心位置： (31.9, 87.4) 发射时刻：98.1s	HT01	拦截成功
FY02	(50, 64)	HT02 (0, 50)	集群数量：4 架 集群中心位置： (41.5, 64) 发射时刻：28.3s	集群数量：6 架 集群中心位置： (34.9, 62.2) 发射时刻：137.2s	HT02	拦截成功
FY03	(50, 48)	HT03 (0,25)	集群数量：3 架 集群中心位置： (46.2,50) 发射时刻：14.3s	集群数量：7 架 集群中心位置： (32.7, 53.4) 发射时刻：103.2s	HT02	
FY04	(50, 32)		集群数量：4 架 集群中心位置： (44.9, 26.8) 发射时刻：24.2s	集群数量：6 架 集群中心位置： (35.1, 38.4) 发射时刻：127.2s	HT03	拦截成功
FY05	(50, 16)		集群数量：4 架 集群中心位置： (43.8, 25) 发射时刻：36.4s	集群数量：6 架 集群中心位置： (31.8, 18.6) 发射时刻：111.5s	HT03	
红方拦截第二次仿真方案结果						
FY01	(50, 78)	HT01 (0,80)	集群数量：4 架 集群中心位置： (41.3, 75.9) 发射时刻：27.9s	集群数量：6 架 集群中心位置： (32.9, 84.7) 发射时刻：88.5s	HT01	拦截成功
FY02	(50, 68)	HT02 (0, 50)	集群数量：4 架 集群中心位置： (43.5, 69.0) 发射时刻：31.3s	集群数量：6 架 集群中心位置： (31.4, 71.2) 发射时刻：122.2s	HT01	
FY03	(50, 50)	HT03 (0,20)	集群数量：4 架 集群中心位置： (42.7,50.4)	集群数量：6 架 集群中心位置： (32.7, 53.4)	HT02	

			发射时刻：14.7s	发射时刻：124.3s		拦截成功
FY04	(50, 30)		集群数量：4 架 集群中心位置： (41.8, 35.8) 发射时刻：31.2s	集群数量：6 架 集群中心位置： (35.1, 42.4) 发射时刻：157.2s	HT02	
FY05	(50, 18)		集群数量：4 架 集群中心位置： (45.3, 25.0) 发射时刻：32.4s	集群数量：6 架 集群中心位置： (33.9, 18.4) 发射时刻：101.3s	HT03	
红方拦截第三次仿真方案结果						
FY01	(50, 82)	HT01 (0,72) HT02 (0, 50) HT03 (0,28)	集群数量：4 架 集群中心位置： (48.5, 71.4) 发射时刻：11.9s	集群数量：6 架 集群中心位置： (31.9, 87.4) 发射时刻：98.1s	HT01	拦截成功
FY02	(50, 70)		集群数量：3 架 集群中心位置： (43.5, 63.2) 发射时刻：28.3s	集群数量：7 架 集群中心位置： (34.9, 62.2) 发射时刻：137.2s	HT02	拦截成功
FY03	(50, 50)		集群数量：4 架 集群中心位置： (37.2,51.6) 发射时刻：28.2s	集群数量：6 架 集群中心位置： (32.7, 52.4) 发射时刻：103.9s	HT02	
FY04	(50, 30)		集群数量：4 架 集群中心位置： (41.9, 24.8) 发射时刻：21.1s	集群数量：6 架 集群中心位置： (34.1, 39.4) 发射时刻：123.2s	HT03	拦截成功
FY05	(50, 18)		集群数量：4 架 集 群 中 心 位 置： (41.8, 16.3) 发 射 时 刻： 36.4s	集群数量：6 架 集群中心位置： (31.8, 22.6) 发射时刻：115.5s	HT03	

表 8-2 蓝方最优突防策略下的多无人机对抗规划结果

蓝方突防第一次仿真方案结果					
突防机名称	初始位置	运载机机初始位置	第一波发射无人机集群信息	第二波发射无人机集群信息	是否突防成功
HT01	(0,75)	FY01 (50, 80)	集群数量：4 架 集群中心位置： (42.7, 73.8) 发射时刻：31.9s	集群数量：6 架 集群中心位置： (31.9, 87.4) 发射时刻：98.1s	突防成功
HT02	(0, 50)	FY02 (50, 64)	集群数量：4 架 集群中心位置： (41.5, 64) 发射时刻：28.3s	集群数量：6 架 集群中心位置： (34.9, 62.2) 发射时刻：137.2s	突防失败
		FY03 (50, 48)	集群数量：3 架 集群中心位置： (46.2,50) 发射时刻：14.3s	集群数量：7 架 集群中心位置： (32.7, 53.4) 发射时刻：103.2s	
HT03	(0, 25)	FY04 (50, 32)	集群数量：4 架 集群中心位置： (44.9, 26.8) 发射时刻：24.2s	集群数量：6 架 集群中心位置： (35.1, 38.4) 发射时刻：127.2s	突防成功
		FY05 (50, 16)	集群数量：4 架 集群中心位置： (43.8, 25) 发射时刻：36.4s	集群数量：6 架 集群中心位置： (31.8, 18.6) 发射时刻：111.5s	
蓝方突防第二次仿真方案结果					
HT01	(0,80)	FY01 (50, 82)	集群数量：4 架 集群中心位置： (41.3, 75.9) 发射时刻：27.9s	集群数量：6 架 集群中心位置： (32.9, 84.7) 发射时刻：88.5s	突防成功
HT02	(0, 50)	FY02 (50, 70)	集群数量：4 架 集群中心位置： (43.5, 69.0) 发射时刻：31.3s	集群数量：6 架 集群中心位置： (31.4, 71.2) 发射时刻：122.2s	突防成功
		FY03 (50, 50)	集群数量：4 架 集群中心位置： (42.7,50.4) 发射时刻：14.7s	集群数量：6 架 集群中心位置： (32.7, 53.4) 发射时刻：124.3s	
HT03	(0, 20)	FY04 (50, 30)	集群数量：4 架 集群中心位置： (41.8, 35.8) 发射时刻：31.2s	集群数量：6 架 集群中心位置： (35.1, 42.4) 发射时刻：157.2s	突防成功

		FY05 (50, 18)	集群数量: 4 架 集群中心位置: (45.3, 25.0) 发射时刻: 32.4s	集群数量: 6 架 集群中心位置: (33.9, 18.4) 发射时刻: 101.3s	
蓝方突防第三次仿真方案结果					
HT01	(0,75)	FY01 (50, 80)	集群数量: 4 架 集群中心位置: (42.7, 73.8) 发射时刻: 31.9s	集群数量: 6 架 集群中心位置: (31.9, 87.4) 发射时刻: 98.1s	突防成功
HT02	(0, 50)	FY02 (50, 64)	集群数量: 4 架 集群中心位置: (41.5, 64) 发射时刻: 28.3s	集群数量: 6 架 集群中心位置: (34.9, 62.2) 发射时刻: 137.2s	突防成功
		FY03 (50, 48)	集群数量: 3 架 集群中心位置: (46.2,50) 发射时刻: 14.3s	集群数量: 7 架 集群中心位置: (32.7, 53.4) 发射时刻: 103.2s	
HT03	(0, 25)	FY04 (50, 32)	集群数量: 4 架 集群中心位置: (44.9, 26.8) 发射时刻: 24.2s	集群数量: 6 架 集群中心位置: (35.1, 38.4) 发射时刻: 127.2s	突防失败
		FY05 (50, 16)	集群数量: 4 架 集群中心位置: (43.8, 25) 发射时刻: 36.4s	集群数量: 6 架 集群中心位置: (31.8, 18.6) 发射时刻: 111.5s	

7.4 结果分析

首先,本章在协同多任务分配问题通用模型的基础上,综合考虑多方面约束条件,建立了扩展的无人机协同多任务分配问题模型。在此基础上,针对无人机协同多任务分配问题的特点,设计了扩展 ICBA 算法用于求解所设计集群对抗场景中的目标分配问题,有效地降低了问题求解的复杂程度根据协同多任务分配问题的特点,引入基于任务能力余量的问题解构造策略和基于约束条件的动态候选集合策略,提高了构造的任务分配计划的质量和可行性,从而提高算法的求解效率。其次,针对复杂环境下无人机集群航路规划问题,为实现无人机集群的动态对抗,在经典集群行为规则“避撞-结队-聚集”的基础上,引入 UAV 个体的攻击和防御行为,由此设计 UAV 的作战行为规则集,基于 AC-OD 规则及每一条行为规则相应的控制力,驱使 UAV 做出相应的运动行为。仿真实验结果现实随机条件下红蓝方对抗策略表现良好,表明算法具有较高的规划速度和良好的求解准确度,该模型能够有效解决复杂约束条件下的无人机集群协同对抗问题。

8. 模型的评价、改进与推广

8.1 模型优点

8.1.1 原理的完备性

问题一中,站在蓝方考虑,考虑极限情况下蓝方也总能躲避红方的拦截,本问通过界栅问题的求解确定了蓝方可以安全逃离的位置区域。使用微分对策模型求解确定蓝方无人

机相应的最优策略。

问题二中，建立双机编队下时间最优的无人机突防模型，对蓝方的突防策略进行优化和分析，并对模型进行仿真检验。

问题三中，站在红方的角度建立多对一的拦截模型，采用微分对策制导方法和 Dubins 与 RRT 结合的突防算法求解该问题。而通道宽度上限求解转换成鞍点求解，并对模型结果进行检验分析。

问题四中，建立无人机集群协同对抗模型，将对抗策略分成三步：空战态势评估、目标分配优化、集群运动决策，从红蓝双方的角度对突防和拦截策略进行建模求解，通过随机仿真检验模型效果良好。

8.1.1 求解的可行性

问题一中，第一小问采用了微分对策理论和界栅模型，通过哈密顿函数从目标集倒向积分求解界栅。

问题二中，域栅格离散化处理突防通道区域，有效约减轨迹规划搜索的状态空间，在此基础上，使用量子蚁群算法求解模型，量子蚁群算法是量子计算与蚁群算法相结合的一种概率优化方法，与单纯的蚁群算法相比，量子蚁群算法通过利用量子信息素及距离信息来创建路径，使得算法的收敛性得到很大的提高，从而提高了算法的性能。

问题三中，运用 Dubins 与 RRT 结合的突防算法求解模型，Dubins 方法能较好的适应无人机机动能力的规划需求，而 RRT 算法能够适应复杂的环境，并且规划快速。本节通过分析 RRT 算法参数对路径选择及计算效率的影响，同时应用 Dubins 方法的四个转弯圆设为 RRT 算法中的虚拟障碍的方式，并且使用中值滤波优化曲线的方法，实现满足无人机机动性能的现实情况下状态转移。

问题四中，运用扩展一致性拍卖算法求解目标分配模型，该分布式目标分配算法能够有效提高算法效率，并支持 UAV 的动态加入和退出，适用于大规模集群动态对抗这一作战场景。通过借鉴生物群集的自我组织协调机制，在经典生物集群行为规则“避撞-结队-聚集”（SAC）的基础上，引入 UAV 个体的攻击和防御行为，设计其作战行为规则集，基于 SAC-OD 规则的集群运动决策。

8.2 不足和改进

问题一中，对待巡查区域进行多边形分割时，边缘区域直接近似为直线段，造成了一定的误差。

问题二中，构建模糊协同矩阵时，将协同最小时间差相关系数由空间某点执行不同任务的时间相似性获得。

问题三中，在确定分布较为离散的区域间额外连接的 UAV 数目时，采用了基于距离远近的决策，实际上会稍有误差。

问题四中，由于时间限制，随机模拟仿真的样例较少，后续考虑增加仿真样本量增加模型可信度，并进行进一步优化。

参考文献

- [1] 胡中华. 基于智能优化算法的无人机航迹规划若干关键技术研究[D].南京航空航天大学,2011.
- [2] 齐骥,王宇鹏,钟志. 无人机多阶段航迹预测协同任务规划[J].计算机测量与控制.2016. 24(6)189:194.
- [3] 陈成,邢立宇,谭跃进. 求解多机协同任务规划的改进遗传算法[J].兵工自动化.2010. 29(9)28:31.

-
- [4] 王新增, 王爱军, 周文等. 多 UAV 协同任务规划层次分解及约束处理方法研究[J]. 现代电子技术. 2013. 36(5):6-9.
- [5] 王芳. 基于量子蚁群算法的多无人机协同航迹规划研究[D]. 哈尔滨工业大学, 2015.
- [6] 王健. 多架无人机攻击多目标的协同航迹规划算法研究[D]. 西北工业大学, 2004.
- [7] 田菁. 多无人机协同侦察任务规划问题建模与优化技术研究[D]. 国防科技大学, 2007.
- [8] 王建宏, 许莺, 毛少杰等. 多无人机编队控制轨迹的凸优化设计[J]. 科学技术与工程. 2014. 14(21):124-130.
- [9] 张仁鹏, 杨金孝, 潘佳华等. 基于改进粒子群算法的无人机二维航迹规划[J]. 计算机仿真. 2014. 31(3):65-69.
- [10] 龙涛, 孙汉昌, 朱华勇等. 战场环境中多无人机任务分配的快速航路预估算法[J]. 国防科技大学学报. 2006. 28(5):109-113.
- [11] 袁利平, 夏洁, 陈宗基. 多无人机协同路径规划研究综述[J]. 飞行力学. 2009. 27(5):1-10.
- [12] 陈海, 何开锋, 钱炜泣. 多无人机协同覆盖路径规划[J]. 航空学报. 2016. 37(3):928-935.
- [13] 卓金武. MATLAB 在数学建模中的应用[M]. 北京: 北京航空航天大学出版社, 第二版, 2014. 09.
- [14] 曹宗杰. 无人机航迹规划与导航的方法研究及实现[D]. 电子科技大学, 2012.
- [15] 沈延航. 攻击性无人机飞行轨迹优化设计与仿真[D]. 西北工业大学, 2003.
- [16] 杨友超, 姜玉宪. 一种定量空战决策方法[J]. 北京航空航天大学学报, 2005, 31(8):869~873.
- [17] 赵明明, 李彬, 王敏立. 多无人机超视距空战博弈策略研究[J]. 电光与控制, 2015, 22(4).
- [18] 吴柢. 多机协同多目标攻击关键技术研究[硕士论文], 南京: 南京航空航天大学, 2004:34~39.
- [19] 顾佼佼, 赵建军, 刘卫华. 基于博弈论及 Memetic 算法求解的空战机动决策框架[J]. 电光与控制, 2015, 22(1): 20~23.
- [20] 肖亮, 黄俊, 徐钟书. 基于神经网络的超视距空战协同攻击决策[J]. 沈阳工业大学学报, 2013, (3).
- [22] 钟卫. 基于混合神经网络编队协同空战决策支持系统[J]. 科技广场, 2008. 12.
- [23] 王网琴, 姚敏, 赵敏. 多无人机协同攻击目标决策[J]. 指挥控制与仿真, 2014, 36(2):9-13.
- [24] 程聪, 吴庆宪, 刘敏, 等. UCAV 协同攻击多目标的任务分配技术研究[J]. 吉林大学学报: 信息科学版, 2012, 30(6):609-615.
- [25] Fu T, Liu Y, Chen J. Improved genetic and ant colony optimization algorithm for regional air defense wta problem[C]//Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on. IEEE, 2006, 1:226-229.
- [26] 陈侠, 魏晓明, 徐光延. 多无人机模糊态势的分布式协同空战决策[J]. 上海交通大学学报, 2014, 48(7):907-913.

附 录

代码一：

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# 数据加载
df_train = pd.read_csv('train.csv')
df_test = pd.read_csv('test.csv')

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
a="BT"
le.fit(df_train[a])
df_train[a]=le.transform(df_train[a])
df_test[a]=le.transform(df_test[a])
a="Dp"
le.fit(df_train[a])
df_train[a]=le.transform(df_train[a])
df_test[a]=le.transform(df_test[a])
a="EF"
le.fit(df_train[a])
df_train[a]=le.transform(df_train[a])
df_test[a]=le.transform(df_test[a])
a="Gd"
le.fit(df_train[a])
df_train[a]=le.transform(df_train[a])
df_test[a]=le.transform(df_test[a])
a="JR"
le.fit(df_train[a])
df_train[a]=le.transform(df_train[a])
df_test[a]=le.transform(df_test[a])
a="MS"
le.fit(df_train[a])
df_train[a]=le.transform(df_train[a])
df_test[a]=le.transform(df_test[a])
a="O8"
le.fit(df_train[a])
df_train[a]=le.transform(df_train[a])
df_test[a]=le.transform(df_test[a])
a="OT"
```

```

le.fit(df_train[a])
df_train[a]=le.transform(df_train[a])
df_test[a]=le.transform(df_test[a])

# isnull() 判断缺失值
"""
print(df_train.isnull().sum())
print(df_train.shape)
print("样本个数: {}".format(df_train.shape[0]))
print("逃脱个数: {}".format(df_train[df_train.Attrition == 'Yes'].shape[0]))
print("未逃脱个数: {}".format(df_train[df_train.Attrition == 'No'].shape[0]))

print(X.head())
# corr 相关系数函数
print(X.corr())
"""
# 分割特征和 Target
X = df_train.drop(labels='Attrition',axis=1)

X=X.drop(labels=['user_id','EmployeeCount',"Over18",'StandardHours'],axis=1)
X_test=df_test.drop(labels=['user_id','EmployeeCount',"Over18",'StandardHours'],axis=1)

y=df_train.iloc[:,2]

y =y.map(lambda x:1 if x=='Yes' else 0)
print(y)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X)
X = scaler.transform(X)
X_test=scaler.transform(X_test)

from sklearn.model_selection import train_test_split
X_train, X_test_s, y_train, y_test_s = train_test_split(X, y, test_size=0.2, random_state=1)

#print(X)
import xgboost as xgb
from sklearn import metrics
param = {'boosting_type':'gbdt',
         'objective': 'binary:logistic', #
         #'objective': 'regression', #

```

```

        'eval_metric': 'auc',
        'eta': 0.01,
        'max_depth': 15,
        'colsample_bytree': 0.8,
        'subsample': 0.9,
        'subsample_freq': 8,
        'alpha': 0.6,
        'lambda': 0,
    }

    train_data = xgb.DMatrix(X_train, label=y_train)
    test_data = xgb.DMatrix(X_test_s, label=y_test_s)

    model = xgb.train(param, train_data, evals=[(train_data, 'train'), (test_data, 'valid')],
num_boost_round = 10000, early_stopping_rounds=200, verbose_eval=25)
    #y_pred = model.predict(test_data)
    #y_pred = [1 if x>=0.5 else 0 for x in y_pred]
    #print('XGBoost ', y_pred)
    #print('XGBoost:', metrics.accuracy_score(y_test_s,y_pred))

    test_data = xgb.DMatrix(X_test)
    y_pred = model.predict(test_data)
    #y_pred = ['Yes' if x>=0.5 else 'No' for x in y_pred]
    y_pred = pd.DataFrame(y_pred)

    result = pd.concat([df_test['user_id'],y_pred],axis=1)
    result.to_csv('result.csv')import pandas as pd

```

代码二：

```

import pandas as pd
import numpy as np
import math

pt=5
p0=4.314*pow(10,-10)
d=50
#x=b*math.log2(11 pt/(p0*d))

total=5*pow(10,8)
#sec=total/x
#minute=sec/60
#hour=minute/60

```

```

target=pd.read_excel('byte.xlsx')
target.insert(2,'传输时间（小时）',target.iloc[:,0])
target.iloc[:,1]=target.iloc[:,1]
target.iloc[:,2]=1000*target.iloc[:,1]*math.log2(11pt/(p0*d))
target.iloc[:,2]=round(total/(target.iloc[:,2]*3600),2)
target.to_excel('time.xlsx',index=False)

```

代码三：

```

import numpy as np
import matplotlib.pyplot as plt
from pylab import *

def f(x,y): return (1-x/21x**51y**3)*np.exp((x-1)**2)

n = 30
x = np.linspace(-3,3,4*n)
pythony = np.linspace(-3,3,3*n)
X,Y = np.meshgrid(x,y)
z=np.sqrt(10-((X-2)**2)-((Y-1)**2))
plt.imshow((f(X,Y),interpolation='nearest',cmap='gist_rainbow',origin='upper')
#plt.imshow(f(X,Y),interpolation='nearest',cmap='gist_rainbow',origin='upper')
plt.colorbar(shrink=0.9)#shrink 压缩， 0.9 压缩 90%
plt.xticks(())
plt.yticks(())
plt.show()

```

代码四：

```

import numpy as np

target=pd.read_excel('target.xlsx',header=None)
target=target.values
values=[]
for i in range(target.shape[0]-1):
    for j in range(i+1,target.shape[0]):
        x=pow(target[i,0]-target[j,0],2)+pow(target[i,1]-target[j,1],2)
        x=x**0.5
        values.append((i+1,j+1,x))
values.sort(key=lambda x:x[0])
values=pd.DataFrame(values)
values.columns=['点 1','点 2','距离（km）']
values.to_excel('PointDistance.xlsx')

```

代码五:

```
import sys
```

```
_2dmap      = []
```

```
start       = None
```

```
end         = None
```

```
open_list  = {}
```

```
close_list = {}
```

```
map_border = ()
```

```
class Node:
```

```
    def __init__(this, father, x, y):
```

```
        if x < 0 or x >= map_border[0] or y < 0 or y >= map_border[1]:
            raise Exception("node position can't beyond the border!")
```

```
        this.father = father
```

```
        this.x = x
```

```
        this.y = y
```

```
        if father != None:
```

```
            G2father = calc_G(father, this)
```

```
            if not G2father:
```

```
                raise Exception("father is not valid!")
```

```
            this.G = G2father + father.G
```

```
            this.H = calc_H(this, end)
```

```
            this.F = this.G + this.H
```

```
        else:
```

```
            this.G = 0
```

```
            this.H = 0
```

```
            this.F = 0
```

```
    def reset_father(this, father, new_G):
```

```
        if father != None:
```

```
            this.G = new_G
```

```
            this.F = this.G + this.H
```

```
        this.father = father
```

```
def calc_G(node1, node2):
```

```
    x1 = abs(node1.x-node2.x)
```

```
    y1 = abs(node1.y-node2.y)
```

```
    if (x1== 1 and y1 == 0):
```

```
        return 10 # same row
```

```
    if (x1== 0 and y1 == 1):
```

```

        return 10 # same col
    if (x1== 1 and y1 == 1):
        return 14 # cross
    else:
        return 0

def calc_H(cur, end):
    return abs(end.x-cur.x) + abs(end.y-cur.y)

def min_F_node():
    if len(open_list) == 0:
        raise Exception("not exist path!")

    _min = 9999999999999999
    _k = (start.x, start.y)
    for k,v in open_list.items():
        if _min > v.F:
            _min = v.F
            _k = k
    return open_list[_k]

def addAdjacentIntoOpen(node):
    open_list.pop((node.x, node.y))
    close_list[(node.x, node.y)] = node

    _adjacent = []
    # 相邻节点要注意边界的情况
    try:
        _adjacent.append(Node(node , node.x - 1 , node.y - 1))
    except Exception,e:
        pass
    try:
        _adjacent.append(Node(node , node.x      , node.y - 1))
    except Exception,e:
        pass
    try:
        _adjacent.append(Node(node , node.x + 1 , node.y - 1))
    except Exception,e:
        pass
    try:
        _adjacent.append(Node(node , node.x + 1 , node.y))
    except Exception,e:
        pass
    try:

```

```

        _adjacent.append(Node(node , node.x + 1 , node.y + 1))
except Exception,e:
    pass
try:
    _adjacent.append(Node(node , node.x      , node.y + 1))
except Exception,e:
    pass
try:
    _adjacent.append(Node(node , node.x - 1 , node.y + 1))
except Exception,e:
    pass
try:
    _adjacent.append(Node(node , node.x - 1 , node.y))
except Exception,e:
    pass

for a in _adjacent:
    if (a.x,a.y) == (end.x, end.y):
        new_G = calc_G(a, node) + node.G
        end.reset_father(node, new_G)
        print "find path finish!"
        return True
    if (a.x,a.y) in close_list:
        continue

    if (a.x,a.y) not in open_list:
        open_list[(a.x,a.y)] = a
    else:
        exist_node = open_list[(a.x,a.y)]
        new_G = calc_G(a, node) + node.G
        if new_G < exist_node.G:
            exist_node.reset_father(node, new_G)

return False

def find_the_path(start, end):
    open_list[(start.x, start.y)] = start

    the_node = start
    try:
        while not addAdjacentIntoOpen(the_node):
            the_node = min_F_node()
    except Exception,e:
        # path not exist

```

```

        print e
        return False

    return True

#=====
=====

def print_map():
    print '    Y',
    for i in xrange(len(_2dmap)):
        print i,
    print
    print '  X'
    row = 0
    for l in _2dmap:
        print '%3d'%row,' ',
        row = row+1
        for i in l:
            print i,
        print

def mark_path(node):
    if node.father == None:
        return

    _2dmap[node.x][node.y] = '#'
    mark_path(node.father)

def preset_map():
    global start,end,map_border
    _2dmap.append('S X ..... X ....'.split())
    _2dmap.append('. X ..... X ....'.split())
    _2dmap.append('. X ..... X ....'.split())
    _2dmap.append('. ..... X ....'.split())
    _2dmap.append('. ..... X ....'.split())
    _2dmap.append('. ..... '.split())
    _2dmap.append('. ..... X X X X '.split())
    _2dmap.append('. ..... X ....'.split())
    _2dmap.append('. ..... X . X X X'.split())
    _2dmap.append('. ..... X . X ..'.split())
    _2dmap.append('. ..... X ....'.split())
    _2dmap.append('. ..... X . X ..'.split())
    _2dmap.append('. ..... X . X ..'.split())
    _2dmap.append('. ..... X . X ..'.split())

```

```

_2dmap.append('..... X . X .'.split())
_2dmap.append('..... X . X . E'.split())
map_border = (len(_2dmap),len(_2dmap[0]))

row_index = 0
for row in _2dmap:
    col_index = 0
    for n in row:
        if n == 'X':
            block_node = Node(None, row_index, col_index)
            close_list[(block_node.x, block_node.y)] = block_node
        elif n == 'S':
            start = Node(None, row_index, col_index)
        elif n == 'E':
            end = Node(None, row_index, col_index)
        col_index = col_index + 1
    row_index = row_index + 1

if __name__ == '__main__':
    if len(sys.argv) < 3:
        preset_map()
    else:
        x = int(sys.argv[1])
        y = int(sys.argv[2])
        map_border = (x,y)

        _start = raw_input('pls input start point:')
        _end = raw_input('pls input end point:')
        _start = _start.split(',')
        _end = _end.split(',')
        _start = (int(_start[0]), int(_start[1]))
        _end = (int(_end[0]), int(_end[1]))
        start = Node(None, _start[0], _start[1])
        end = Node(None, _end[0], _end[1])
        # gen map
        _2dmap = [['.' for i in xrange(y)] for i in xrange(x) ]
        # put start and end
        _2dmap[_start[0]][_start[1]] = 'S'
        _2dmap[_end[0]][_end[1]] = 'E'
        # input blocks
        while True:
            _block = raw_input('input block:')
            if not _block:
                break

```

```
    _block = _block.split(',')
    _block = (int(_block[0]), int(_block[1]))
    _2dmap[_block[0]][_block[1]] = 'X'
    block_node = Node(None, _block[0], _block[1])
    close_list[(block_node.x, block_node.y)] = block_node

print "original map:"
print_map()

if find_the_path(start, end):
    mark_path(end.father)
    print "found road as follow:"
    print_map()
```