

| | |
|------|---------|
| 学校代码 | 10699 |
| 分类号 | TP391.9 |
| 密 级 | 公开 |
| 学 号 | |

题目 编队协同智能空战战术
 决策模型研究

作者 ***

专业领域 电子信息
指导教师 ***副教授
培养单位 电子信息学院
申请日期 2024 年 01 月

西北工业大学
硕士学位论文

题目：编队协同智能空战战术
决策模型研究

专业领域： 电子信息
作者： ***
指导教师： ***

2024年1月

**Title: Research on Tactical Decision-Making
Models for Coordinated Squadron-Based
Intelligent Aerial Combat**

By

Under the Supervision of Associate Professor

A Dissertation Submitted to
Northwestern Polytechnical University

In Partial Fulfillment of the Requirement
For The Degree of
Master of Electronic Information

Xi'an P. R. China
January 2024

摘要

随着无人机相关领域研究的不断深入，多无人机协同研究成为当下无人机研究的主流方向。本文面向未来无人机空战战术决策展开研究，主要包括进入作战区域前的航路规划引导和进入作战区域后的协同战术决策。在进入作战区域前，对多无人机进行航路规划引导，在进入作战区域后，对无人机空战机动决策和编队协同策略模型构建展开研究。基于以上内容，本文主要从以下三个方面展开研究：

(1) 针对无人机协同航路规划问题，本文提出一种基于改进灰狼优化算法的无人机协同航路规划模型，实现了抵达作战区域前的多机引导与航路规划。首先，构建了无人机协同航路规划模型，包括协同约束分析、单机评价函数和多机协同航迹评价函数等；其次，针对灰狼算法（Grey Wolf Optimizer, GWO）收敛速度慢和易陷入局部最优等问题进行改进，利用非线性函数优化收敛系数同时根据不同灰狼的适应度设计权重比例因子，建立了改进的灰狼优化算法（NI-GWO）模型；最后，建立多无人机航路规划的仿真平台，完成算法进行验证，并对算法的鲁棒性进行分析。仿真结果表明，相较于传统算法，本章所提的 NI-GWO 模型在解决多机航路规划问题中拥有着卓越的表现，同时算法在迭代次数和作战场景上有较强鲁棒性。

(2) 针对空战机动决策问题，本文提出了一种基于 E-SAC 算法的无人机空战决策模型，实现了空战机动决策中对目标的追踪与打击。首先，建立导弹攻击区解算的数学模型，包括导弹运动与导引建模、基于改进的神经网络进行了攻击区拟合；其次，针对 SAC 算法在解决机动决策问题时收敛速度慢和易陷入局部最优等问题进行了改进，利用 Expert Actor 增强智能体的探索效率，建立了 E-SAC 算法模型；最后，基于 MaCA 平台实现无人机一对一场景下的空战机动决策仿真，并对改进算法的有效性进行分析。仿真结果表明，本文所提的 E-SAC 算法模型可以解决无人机机动决策问题，同时该算法相较于 SAC 算法在收敛速度和机动决策方面有显著的提升。

(3) 针对无人机编队协同空战决策问题，本文提出了一种基于分层强化学习的空战策略模型，实现了多无人机编队协同空战策略模型的构建。首先，建立了无人机编队协同空战元策略模型，包括雷达开关、主动干扰、主动探测、队形转换、机动决策以及元策略模型的封装；其次，针对无人机空战战术决策中终止函数设计方法的不明确性，建立了基于分层强化学习的无人机全流程空战策略模型；最后，基于 MaCA 平台实现无人机四对四场景下的编队协同空战战术决策仿真，并对模型的有效性进行分析。仿真结果表明，本文所提的分层强化学习模型可以对无人机编队做出有效的协同空战决策，同时，相较于传统的 Option-Critic 模型所提的模型收敛速度更快，决策效率更高。

关键词：无人机编队；战术决策；航路规划；机动决策；分层强化学习

Abstract

As research in unmanned aerial vehicle (UAV) related fields deepens, multi-UAV collaborative research has emerged as a mainstream direction. This paper focuses on UAV tactical decision-making for future aerial combat, encompassing both pre-combat area entry route planning and post-entry collaborative tactical decision-making. Prior to entering combat zones, multi-UAV flight path guidance is conducted, followed by an exploration of UAV aerial combat maneuvering decision-making and formation collaborative strategy models upon entry. This research unfolds in three key areas:

- (1) For collaborative UAV route planning, a novel model based on an improved Grey Wolf Optimization algorithm (NI-GWO) is proposed. This model facilitates multi-UAV guidance and planning before reaching the combat zone. Initially, a collaborative route planning model for UAVs, including collaborative constraints analysis, individual UAV evaluation functions, and multi-UAV trajectory evaluation functions, is constructed. The Grey Wolf Optimizer (GWO) is then enhanced to overcome issues like slow convergence and local optimum entrapment, using a nonlinear function to optimize convergence coefficients and designing weight proportion factors based on the fitness of different wolves. The efficacy of this model is validated through simulation platforms for multi-UAV route planning, demonstrating its robustness and superior performance compared to traditional algorithms.
- (2) Addressing aerial combat maneuver decision-making, an innovative UAV combat decision model based on the Enhanced Soft Actor-Critic (E-SAC) algorithm is introduced, focusing on target tracking and engagement in aerial combat. Initially, a mathematical model for missile attack zone calculation is established, incorporating missile motion and guidance modeling, and attack zone fitting using an improved neural network. The SAC algorithm is then modified, introducing an Expert Actor to enhance exploration efficiency, resulting in the E-SAC model. Simulations for one-on-one UAV combat scenarios validate the effectiveness of this improved algorithm, which shows significant advancements over the SAC algorithm in terms of convergence speed and maneuver decision-making.
- (3) For UAV formation collaborative aerial combat decision-making, a layered reinforcement

learning-based aerial combat strategy model is proposed. This involves establishing a meta-strategy model for UAV formation collaborative aerial combat, including aspects like radar usage, active jamming, active detection, formation transitions, and maneuver decisions. The uncertainty in termination function design methods in UAV aerial combat tactical decision-making is addressed by developing a full-process aerial combat strategy model based on layered reinforcement learning. Four-on-four UAV formation collaborative aerial combat simulations on the MaCA platform demonstrate the model's effectiveness, offering faster convergence and higher decision efficiency compared to traditional Option-Critic models.

Key words: UAV formation; Tactical decision-making; Route planning; Maneuver decision-making; Hierarchical reinforcement learning.

目 录

| | |
|----------------------------------|------|
| 摘要 | I |
| Abstract | III |
| 目录 | V |
| 图索引 | IX |
| 表索引 | XIII |
| 第 1 章 绪论 | 1 |
| 1.1 研究背景及意义 | 1 |
| 1.2 国内外研究现状 | 2 |
| 1.2.1 无人机协同航路规划研究现状 | 2 |
| 1.2.2 无人机空战机动决策研究现状 | 3 |
| 1.2.3 无人机编队协同空战战术决策的研究现状 | 5 |
| 1.3 论文研究内容与创新点 | 7 |
| 1.3.1 论文研究内容 | 7 |
| 1.3.2 论文创新点 | 9 |
| 1.4 论文组织结构 | 10 |
| 1.5 本章小结 | 12 |
| 第 2 章 无人机空战战术决策相关技术 | 13 |
| 2.1 无人机数学模型 | 13 |
| 2.1.1 无人机动力学模型 | 13 |
| 2.1.2 无人机运动学模型 | 14 |
| 2.2 强化学习理论基础 | 14 |
| 2.2.1 MDP 与 SMDP | 15 |
| 2.2.2 基于价值的学习 | 18 |
| 2.2.3 基于策略的学习 | 19 |
| 2.2.4 Actor-Critic 算法 | 19 |
| 2.2.5 分层强化学习 | 20 |
| 2.3 本章小结 | 22 |
| 第 3 章 基于改进灰狼算法的无人机协同航路规划模型 | 23 |
| 3.1 问题分析 | 23 |
| 3.2 多机协同航路规划模型 | 23 |

| | |
|-----------------------------------|----|
| 3.2.1 协同约束分析 | 24 |
| 3.2.2 单机航迹评价函数 | 24 |
| 3.2.3 多机协同航迹评价函数 | 25 |
| 3.3 基于改进灰狼优化算法的多无人机航路规划模型 | 25 |
| 3.3.1 灰狼优化算法 | 25 |
| 3.3.2 改进灰狼优化算法 | 29 |
| 3.4 仿真结果与分析 | 32 |
| 3.4.1 实验环境设定 | 32 |
| 3.4.2 仿真结果与分析 | 32 |
| 3.5 本章小结 | 47 |
| 第 4 章 基于深度强化学习的无人机空战机动决策模型 | 49 |
| 4.1 问题分析 | 49 |
| 4.2 导弹攻击区解算方法 | 49 |
| 4.2.1 导弹数学模型 | 49 |
| 4.2.2 导弹运动与导引模型 | 50 |
| 4.2.3 传统 BP 神经网络的导弹攻击区拟合方法 | 53 |
| 4.2.4 基于改进神经网络的攻击区拟合方法 | 56 |
| 4.3 基于改进 SAC 算法的无人机空战机动决策方法 | 61 |
| 4.3.1 状态空间 | 61 |
| 4.3.2 动作空间 | 62 |
| 4.3.3 奖励函数 | 62 |
| 4.3.4 E-SAC 算法 | 63 |
| 4.4 仿真结果与分析 | 67 |
| 4.4.1 实验环境设定 | 67 |
| 4.4.2 训练结果与分析 | 69 |
| 4.5 本章小结 | 72 |
| 第 5 章 基于分层强化学习的编队协同策略模型 | 73 |
| 5.1 问题分析 | 73 |
| 5.2 编队协同决策元策略模型 | 73 |
| 5.2.1 雷达开关模型 | 74 |
| 5.2.2 主动干扰模型 | 75 |
| 5.2.3 主动探测模型 | 75 |
| 5.2.4 队形转换模型 | 77 |

| | |
|----------------------------|-----------|
| 5.2.5 机动决策模型 | 77 |
| 5.2.6 元策略模型封装 | 77 |
| 5.3 改进分层强化学习算法战术决策模型 | 79 |
| 5.3.1 元策略决策网络模型 | 79 |
| 5.3.2 顶层策略网络模型 | 83 |
| 5.4 仿真结果与分析 | 84 |
| 5.4.1 实验环境设定 | 84 |
| 5.4.2 全流程仿真结果与分析 | 85 |
| 5.5 本章小结 | 91 |
| 第 6 章 总结与展望 | 错误!未定义书签。 |
| 6.1 论文总结 | 93 |
| 6.2 后续展望 | 94 |
| 参考文献 | 95 |
| 致 谢 | 111 |
| 在学期间发表的学术成果和参加科研情况 | 113 |

图索引

| | |
|-------------------------------------|----|
| 图 1-1 论文研究内容..... | 8 |
| 图 1-2 论文组织结构图..... | 11 |
| 图 2-1 无人机三自由度质点模型示意图..... | 13 |
| 图 2-2 强化学习中智能体与环境之间交互的示意图..... | 15 |
| 图 2-3 马尔可夫决策过程表示图..... | 16 |
| 图 2-4 MDP 与 SMDP 状态比较..... | 17 |
| 图 2-5 Actor-Critic 算法基本流程 | 19 |
| 图 3-1 基于改进灰狼算法的无人机协同航路规划模型内容框图..... | 23 |
| 图 3-2 灰狼种群等级制度..... | 25 |
| 图 3-3 狼群捕猎模型..... | 26 |
| 图 3-4 二维位置向量和潜在的未来位置..... | 27 |
| 图 3-5 灰狼算法中的位置更新..... | 28 |
| 图 3-6 捕猎猎物和搜索猎物..... | 28 |
| 图 3-7 收敛因子变化曲线..... | 30 |
| 图 3-8 两架无人机在不同算法下的仿真结果图..... | 34 |
| 图 3-9 两架无人机的航路规划飞行数据..... | 34 |
| 图 3-10 三架无人机在不同算法下的仿真结果图..... | 35 |
| 图 3-11 三架无人机的航路规划飞行数据..... | 36 |
| 图 3-12 三架无人机的因子强度图..... | 36 |
| 图 3-13 四架无人机在不同算法下的仿真结果图..... | 37 |
| 图 3-14 四架无人机的航路规划飞行数据..... | 38 |
| 图 3-15 四架无人机背景下的因子强度图..... | 38 |
| 图 3-16 迭代次数为 100 次的仿真结果图..... | 39 |
| 图 3-17 迭代次数为 100 次的航路规划飞行数据..... | 40 |
| 图 3-18 迭代次数为 100 次的因子强度图..... | 40 |
| 图 3-19 迭代次数为 100 次的仿真结果图..... | 41 |
| 图 3-20 迭代次数为 60 次的航路规划飞行数据..... | 42 |
| 图 3-21 迭代次数为 60 次的因子强度图..... | 42 |
| 图 3-22 减少威胁区数量后的仿真结果图..... | 43 |
| 图 3-23 减少威胁区后航路规划飞行数据..... | 44 |

| | |
|-------------------------------------|----|
| 图 3-24 减少威胁区后因子强度图..... | 44 |
| 图 3-25 增加威胁区数量的仿真结果图..... | 46 |
| 图 3-26 增加威胁区后航路规划飞行数据..... | 46 |
| 图 3-27 增加威胁区后因子强度图..... | 47 |
| 图 4-1 基于深度强化学习的无人机空战机动决策模型内容框图..... | 49 |
| 图 4-2 导弹与目标相对运动关系示意图..... | 50 |
| 图 4-3 BP 神经网络结构图..... | 53 |
| 图 4-4 BP 神经网络算法流程图..... | 54 |
| 图 4-5 BP 神经网络仿真结果图(1)..... | 55 |
| 图 4-6 BP 神经网络仿真结果图(2)..... | 56 |
| 图 4-7 TBPNN 算法流程图..... | 58 |
| 图 4-8 TBPNN 仿真结果图(1)..... | 60 |
| 图 4-9 TBPNN 仿真结果图(2)..... | 61 |
| 图 4-10 空战态势图..... | 61 |
| 图 4-11 无人机空战示意图..... | 62 |
| 图 4-12 E-SAC 算法示意图 | 65 |
| 图 4-13 MaCA 环境软件结构及运行流程..... | 68 |
| 图 4-14 训练奖励曲线..... | 70 |
| 图 4-15 环境 1 中无人机一对一机动决策训练仿真过程..... | 71 |
| 图 4-16 环境 4 中无人机一对一机动决策训练仿真过程..... | 72 |
| 图 5-1 基于分层强化学习的编队协同策略模型研究内容..... | 73 |
| 图 5-2 无人机编队协同空战策略模型的分层结构..... | 74 |
| 图 5-3 雷达探测重叠区域分析..... | 74 |
| 图 5-4 机载雷达作战过程示意图..... | 76 |
| 图 5-5 元策略封装结构图..... | 78 |
| 图 5-6 元策略与空战全流程的联系..... | 79 |
| 图 5-7 Option-Critic 结构 | 81 |
| 图 5-8 Options 模型构建及训练流程 | 82 |
| 图 5-9 策略选择器模型构建..... | 83 |
| 图 5-10 我机分布式搜索策略..... | 86 |
| 图 5-11 初始编队示意图..... | 87 |
| 图 5-12 训练奖励曲线图..... | 88 |
| 图 5-13 无人机编队四对四机动决策训练仿真过程..... | 88 |

| | |
|--------------------------------|----|
| 图 5-14 训练曲线图..... | 89 |
| 图 5-15 无人机编队四对四机动决策训练仿真过程..... | 90 |

表索引

| | |
|---|----|
| 表 3-1 航路规划模型仿真参数表..... | 32 |
| 表 3-2 初始位置和目标位置设置表..... | 33 |
| 表 3-3 威胁区设置表..... | 33 |
| 表 3-4 初始位置和目标位置设置表..... | 35 |
| 表 3-5 初始位置和目标位置设置表..... | 37 |
| 表 3-6 威胁区设置表..... | 43 |
| 表 3-7 增加后的威胁区设置表..... | 45 |
| 表 4-1 BP 神经网络参数..... | 54 |
| 表 4-2 评估分数模块伪代码..... | 58 |
| 表 4-3 优化主循环模块伪代码..... | 59 |
| 表 4-4 基于 E-SAC 的无人机自主机动决策算法伪代码 | 66 |
| 表 4-5 无人机空战机动决策仿真参数表..... | 68 |
| 表 4-6 空战初始态势..... | 69 |
| 表 4-7 环境 1 训练结果..... | 69 |
| 表 4-8 环境 2 训练结果..... | 69 |
| 表 4-9 环境 3 训练结果..... | 70 |
| 表 4-10 环境 4 训练结果..... | 70 |
| 表 5-1 基于 Option-Critic 的算法伪代码 | 80 |
| 表 5-2 奖励设置..... | 83 |
| 表 5-3 基于分层强化学习的编队协同智能空战战术决策模型仿真参数表..... | 85 |
| 表 5-4 雷达开关策略有效性验证..... | 85 |
| 表 5-5 雷达开关策略有效性验证..... | 86 |
| 表 5-6 阵形转换策略有效性验证..... | 87 |
| 表 5-7 分层强化学习训练结果..... | 89 |
| 表 5-8 分层强化学习仿真结果..... | 90 |

第1章 绪论

本章首先讨论编队协同智能空战战术决策模型的研究背景与意义。在此基础上，分别对无人机协同航路规划、无人机空战机动决策和无人机协同空战战术决策的国内外研究现状进行总结和分析，最后，凝练出本文的研究内容和创新点。

1.1 研究背景及意义

在军事领域，无人机（Unmanned Aerial Vehicle, UAV）技术的飞速发展^[1]正彰显其日益重要的战略地位。无人机因其卓越的机动性和灵活性而作为未来先进武器系统新的发展方向^[2]。伴随着计算机、电子信息技术的突飞猛进，无人机技术已取得显著的进步^[3]，从而极大地推动了军事任务向多样化和自主化的演进。

在现代战争中，无人机的单机作战已经扩展到多机编队协同作战^[4]。这种转变不仅体现了技术的进步，更标志着战术和战略层面的创新。无人机编队协同作战能力的提高，意味着能更有效地应对快速变化的战场环境，实现更高的作战效率和适应性。因此，深入探索和优化无人机编队的协同决策机制变得尤为重要，这不仅推动了无人机技术的发展，也为未来的空中作战策略提供了新的视角。目前，人工智能算法在多无人机协同作战决策算法中扮演着重要的角色。

自 21 世纪以来，人工智能进入了一个快速发展的阶段，它的核心是通过模拟人类的认知过程，使得机器能够执行原本需要人类智能才能完成的复杂任务^[5]。随着技术的不断进步，人工智能已经超越了简单任务的自动化，扩展到需要复杂决策和认知能力的领域。强化学习作为人工智能的一个关键分支，通过奖励机制引导机器学习策略，已在围棋^[6]和电子游戏^[7]等领域展现出超越人类的能力。如今这种基于强化学习的人工智能技术也正逐渐应用于军事领域^[8]，特别是在编队协同空战战术决策中^[9]，通过深入研究智能空战战术决策模型，人工智能可以实现更加高效、自主和智能化的协同作战策略，显著提升战斗机任务系统的性能和适应能力，也为现代战争的未来发展提供了新的视角和可能性。

在此背景下，采用智能化策略以实现无人机编队协同空战战术决策，不仅能够丰富未来军事战略和战术的理论基础，且具有实践意义。论文以进入作战区域前对无人机的航路规划引导和进入作战区域后对无人机的空战机动决策以及编队协同策略模型构建展开研究。此研究使得无人机编队在执行攻击任务时能够独立于人工操控，根据战场上的态势变化，自主做出精准的攻击决策，以提高无人机编队在作战中的自治能力和智能化水平，推动未来无人作战系统的发展。

1.2 国内外研究现状

多无人机战术决策研究包含了无人机空战的多个流程，可以将无人机战术决策的研究划分为抵达作战区域前的航路规划和抵达作战区域后的战术决策。多无人机在抵达作战区域前，需要考虑空域中的威胁区等因素进行航路规划；多无人机抵达作战空域后，根据战场的态势选择合适的战术策略与敌机进行对抗，完成与整个空战流程。其中空战战术决策包含顶层编队战术策略和底层单无人机机动策略。基于此，本文的三个关键技术分别为：无人机协同航路规划、无人机空战机动决策和无人机编队协同策略模型构建。

1.2.1 无人机协同航路规划研究现状

多无人机航路规划是根据各无人机的特定任务，为每架无人机规划出从起始点到目标点的航路轨迹，实现指定性能指标最优或较优^[10]。

与单无人机航路规划相比，多无人机航路规划的复杂性主要体现为：1) 规划空间范围大而且复杂，存在着各种空间障碍和动态威胁等；2) 约束条件多，不但要考虑规划出的可飞行航路符合无人机实际的动力学和运动学特性，而且要考虑满足时间和空间的协同性、航路的隐蔽性等；3) 无人机航路规划还要适应战场动态变化，能够对航路进行在线实时调整。

针对多无人机航路规划的复杂性，传统算法已不再适合处理当前的大规模规划问题。目前，国内外研究学者提出了不同类型的元启发式算法来解决无人机自主轨迹规划问题。例如人工势场方法^[11]、模拟退火算法^[12]。文献[13]提出了一种基于遗传算法和进化策略的混合算法，用于电磁优化问题。为了使算法更适合灾难场景，文献[14]提出了一种自适应选择变异约束差分进化算法，根据个体适应度和约束选择个体。文献[15]提出了离散杜鹃搜索和模拟退火算法来解决连续优化问题。启发式优化算法具有快速规划速度、良好的并行性和易于操作的优势，但该类算法易陷入局部最优解，并且非常依赖启发式信息，这使得算法复杂性、普适性以及收敛速度难以确定。

另一种是群体智能优化算法，越来越多的研究致力于实现算法的收敛性和搜索能力之间的平衡^[16-18]。群体智能优化算法基于对生物群体行为的研究，生物群体通过合作、信息交换和群体之间的互动来实现优化目标。常见的群智能算法如粒子群优化算法^[19]、蚁群优化算法^[20]。文献[21]提出了一种改进的粒子群优化算法。蚁群优化算法^{[22][23]}已被引入以自适应优化位置数据的聚类数量并解决单一 k-means 算法的过拟合问题。基于逻辑映射的改进鲸鱼优化算法^[24]被设计用于进行参数识别，与许多传统优化算法不同，该算法依赖于每个生物体通过无意识进化和优化行为来优化生存状态，以适应不同的任务环境。尽管上述提及的群智能优化算法在许多应用场景中表现出色，但它们常面临陷入局部最优解的挑战。

为了解决这一问题，在2014年，Mirjalili受到了灰狼狩猎行为的启发提出一种新型元启发式算法——灰狼算法(Grey Wolf Optimizer, GWO)^[25]。该算法基于灰狼群体的多层社会结构，并利用狼群的社会行为来寻找和围捕猎物，以确定围捕猎物的最佳位置，与其他元启发式算法相比，GWO算法由于存在全局搜索和局部搜索的机制，因而提高了全局最优解时的效率和准确性，通常能够在较少的迭代次数内找到解决方案，这有助于减少计算时间，并且实现相对简单，不需要复杂的参数调整，易于使用和理解。该算法模拟了灰狼的社会行为和领导层次结构。GWO算法的仿生结构使其具有在实际应用中相对容易实施和在处理各种类型的优化问题时表现出较好的灵活性的优势^[26]。GWO算法已被应用于解决许多工程应用和控制问题，如负荷频率控制^[27]、特征选择^[28]和无人机航路规划^[29]。文献[30]提出了一种启发式灰狼优化器来解决数值优化问题。文献[31]提出了一种受天体物理学启发的灰狼算法来解决工程设计问题。文献[32]提出了一种探索增强型灰狼优化器来解决高维数值优化。

根据对现有文献的分析，可以观察到元启发式算法在处理多无人机航路规划问题时显示出一定的优势。然而，也有观点指出，传统的元启发式算法在某些情况下可能会面临一些挑战，如有时可能会陷入局部最优解，以及收敛速度可能相对较慢^[33]。因此需要对算法进一步改进和优化。借助文献^[34-39]中的先进优化思想，本文对GWO算法进行改进，通过非线性函数优化收敛系数并为不同适应度的灰狼分配不同的权重比例因子，以增强GWO算法的探索和开发能力并提升传统GWO算法的收敛速度，当存在较大的收敛系数和比例权重时，将有助于算法提高收敛速度，并更快地收敛到更优解的附近。

1.2.2 无人机空战机动决策研究现状

无人机自主机动决策是指无人机执行作战任务时，能够基于空战环境、态势等因素自主生成飞行决策的过程^[40]。随着空战环境的日益复杂化和任务需求的多样化，基于地面站控制的无人机空战机动决策已经无法满足复杂的空战需求。基于此，国内外相关学者对无人机机动决策领域的研究主要分为：基于经典算法和基于人工智能的决策方法。

近年来，经典算法在解决无人机空战机动决策问题时主要以博弈论等算法为代表，如文献[41]中，研究者开发了一种基于博弈矩阵的方法，可以通过比较不同机动组合的方向、距离、速度和地形间隙得分，生成丘陵地形中一对一空战的低空飞行飞机的智能机动决策。文献[42]则设计了一种基于微分博弈理论的生成算法，用于近距离空对空战斗的无人机，它采用层次决策结构和评分函数矩阵来评估UCAV的空中优势。文献[43]介绍了一种用于模拟一对一空战中飞行员机动决策的多阶段影响图博弈方法，这一方法不仅以图形化的方式展示了决策过程和飞机动力学模型，还考虑了飞行员在不确定性条件下的偏好，为最优空战机动策略分析和空战模拟器中战斗机动选择的自动化决策系统开发提供了新的思路。文献[44]深入探讨了一对一空中战斗的微分博弈模型，并分析了

双目标博弈公式及其解决方案的复杂性。文献[45]则通过分析三维轨迹、相对角度的历史和占有率表格，研究了基于微分博弈的一对一近视距离空对空战斗机动生成算法，该算法结合评分函数矩阵和敌方机动分析来确定最优战术动作。文献[46]介绍了一种处理一对一空战机动中水平飞行定速问题的近似动态规划方法，该方法能够快速适应不断变化的战术环境，并提供了长期规划视角和卓越性能，无需对空战策略进行具体编码。文献[47]介绍了一种适用于未知环境中的改进人工势场法，结合 Tangent Bug 算法进行无人机的避障操作，并在有威胁区的环境下利用 A*算法生成预规划路径。

在机动决策的研究领域中，智能算法已成为推动技术创新的主要驱动力。这些算法，特别是基于人工智能的方法，在处理复杂决策环境和动态适应新情况方面表现出卓越的能力。文献[48]在构建超视距空中对抗训练环境的基础上，提出了启发式 Q-Network 方法，结合专家经验和强化学习，有效地提升了空中对抗机动策略的学习效率和实战应用性能。文献[49]则开发了基于端到端强化学习的深度 Q 网络人工智能策略，它能够直接从高维感官输入中学习，并在一系列游戏测试中达到与专业人类玩家相当的性能。文献[50]利用飞机简化运动学方程、改进的三自由度控制指令，以及基于轨迹采样、特征提取和策略学习的方法，开发了一种近似动态规划方法，显著提升了无人机在模拟空战训练中的智能决策能力。文献[51]构建了基于深度强化学习的无人机短程空战控制模型，通过锁定参数网络重启机制优化深度确定性策略梯度算法，显著提高了无人机在复杂战斗机动控制方面的性能和效率。文献[52]设计了包含能量状态的奖励函数，并将其应用与强化学习算法之中，实现了空战智能决策，经验证，该文献所提出的模型可以有效提升无人机空战决策的能力。文献[53]设计了基于粒子群优化径向基函数的预测方法，结合模拟操作命令和最终奖励值，提出了一种深度确定性策略梯度算法，能够在一定时间内将最终奖励值反馈到之前的奖励值进行离线训练，有效提高了算法的收敛速度。文献[54]采用基于邻近策略优化的深度强化学习方法，专注于自主近距空战策略的学习。通过端到端的方式从观察中提炼策略，并在设计的连续动作空间中实施，显著提高了学习效率，并能以大约 97% 的胜率成功击败极小极大策略的对手。文献[55]提出了改进的 (Deep Deterministic Policy Gradient, DDPG) 算法，修改了无人机模型，并基于向量距离处理了经验数据，以提高计算效率。由于 DDPG 算法包含众多超参数并对其敏感，在算法训练阶段调整参数需要很长时间，于是研究人员在 DDPG 算法的基础上提出了双延迟深度确定性策略梯度(Twin Delayed Deep Deterministic Policy Gradient, TD3)算法^[56]，优化了 DDPG 算法中值函数过估计的问题。在文献[57]中，基于 TD3 的无人机导航算法被提出，允许无人机在有多个威胁区的动态环境中执行导航任务。虽然该方法可以通过在先前任务中获得的知识或经验减少在各种应用场景中策略的训练成本，但所需的先验知识仍需要长时间训练。由于空中战斗过程中的状态空间和动作空间较为庞大，深度强

化学习算法在解决无人机在空战中的机动决策问题时，对策略的探索将会变得较为困难。与确定性策略算法不同，软演员批评家(Soft Actor-Critic, SAC)算法^[58]旨在最大化通过熵正则化的累积奖励，而不仅仅是累积奖励。SAC 算法可以增加行动的随机性，鼓励智能体在训练过程中进行探索，从而加快后续学习速度^[59]。许多基准研究已经证明了其最先进的性能。然而，该算法仍有一些局限性，在训练过程中需要大量样本来学习良好的无人机机动决策策略，这导致训练时间长和训练期间样本利用率低。

对上述文献分析可得，基于经典算法的无人机机动决策方法在某些情况下可能表现出一定的局限性，例如在灵活性、实时性以及在处理复杂问题方面。同时，基于学习的方法中的确定性策略算法在训练时间和收敛速度方面也存在一些挑战。这些为无人机机动决策的进一步研究提供了改进的空间和潜在的研究方向。因此，论文采用了人工智能算法对无人机机动决策问题进行研究，并针对其样本利用率低的缺陷进行改进，提出一种 Expert-Actor Critic 算法，通过 Expert Actor 采样环境来获取 Expert 经验样本，防止算法陷入局部最优，并加速其训练过程。

1.2.3 无人机编队协同空战战术决策的研究现状

随着空战环境的日益复杂化和任务需求的多样化，单无人机在面对复杂战场情况时，作战能力会产生明显的劣势。因此，多无人机系统协同执行作战任务是未来无人机发展的主流趋势^[60]，相关学者已开始广泛研究多无人机系统的协同决策问题。目前主流的多机空战战术决策方法有：基于规则的方法^{[61][62]}、基于搜索的方法^{[63][64]}和基于学习的方法^{[65][66]}。

基于规则的方法依赖于专业知识，通过抽象出具有 IF-THEN 结构的问题特定逻辑树来进行决策。文献[61]研究了无人机群体的空中作战战术，构建了一个简单的规则集，并在仿真分析中对智能体进行设计并建立了相关空战模型，以模拟多无人机协同战术。文献[62]开发了一个基于遗传模糊树的二对四的空中作战决策模型。它在模拟空中作战中击败了专业人类飞行员。然而，由于硬编码规则的局限，这种方法不够灵活，无法应对超出专业知识的复杂情景。

基于搜索的方法为使用一些并行搜索和迭代机制来找到明确目标函数的次优解。文献[63]使用多目标遗传算法找到多无人机任务规划问题的帕累托前沿解。文献[64]提出了一种智能自组织蚂蚁群优化算法，用于多无人机协同搜索和攻击任务规划。它可以找到最佳路径点。然而，这种方法需要明确的目标函数，并执行在线搜索。因此，基于搜索的方法的实时性较差。随着问题变得更加复杂，找到最佳解也变得更加困难。

基于学习的方法主要为强化学习(Reinforcement Learning , RL)算法，该方法主要通过智能体反复试验学习以获得最优策略，其中神经网络用于映射状态和行动。文献[65]提出了一种基于多智能体强化学习的三对三空中作战方法，该算法在群体空中作战环境

中训练了多个智能体，并使用场景转移训练和自我对弈来提高多智能体强化学习的学习效率。文献[66]提出了一种基于强化学习的无人机集群任务调度算法，旨在实现无人机集群内任务的高效协作。该算法使无人机能够根据实时数据自动调整任务策略，有效处理动态环境中的任务执行，并通过解决信道分配等关键问题，显著提高了无人机集群的任务性能效率和协作能力。文献[67]研究了深度强化学习算法在多无人机协同控制中的应用，并提出了一种改进的多智能体联合邻近策略优化算法，该算法通过集中学习和分散执行以及移动窗口平均法，有效提高了多无人机系统的协作性和奖励值总和。文献[68]介绍了一个基于 F-16 Fighting Falcon 飞机的仿真模型，用于验证飞机在进行地面碰撞避免和其他自主机动时的性能。该模型包含 16 个连续变量和分段非线性微分方程，使用 MATLAB 和 Python 进行模拟。通过内外环控制器和有限状态机实现自主机动，该模型旨在作为分析航空航天系统行为的基准测试和起点。文献[69]提出了一种新型的自主空战算法，专注于视距内枪炮交战，并基于基本战斗机动进行高性能、实时计算。

在利用强化学习解决全流程空战决策问题时，需要克服在空战环境中奖励稀疏的问题，因此奖励函数的设计的复杂性使得将强化学习应用于空战决策之中成为了挑战^{[70][71]}。文献[72]构建了一个高保真度的空战模拟环境，并提出了一个关键的空战事件奖励塑形机制来减少情节性的胜负信号，使训练过程快速收敛。实施结果表明，强化学习可以在超视距条件下产生多种有价值的空战战术行为。文献[73]研究了在超视距空战中，利用深度强化学习算法提升多架无人机的合作决策和智能优化能力。通过构建了一个新型的超视距空战环境作为深度强化学习的训练环境，以提供高精度的仿真环境，在此基础上提出了将近端策略优化与长短期记忆网络相结合，以处理不完整信息并同时实现智能决策优化。实验验证了深度强化学习可以提高超视距空战中多飞机协同决策的智能化水平。文献[74]基于原始深度强化学习方法设计了一种奖励函数，奖励的设计维度包括机动带来的实时收益以及最终结果的收益。对于奖励稀疏的空战机动决策问题，文献[75]应用了一种基于课程的学习方法来设计一个角度、距离和混合的决策课程，与原始方法相比，提高了训练的速度和稳定性，能够处理来自不同方向的目标。文献[76]提出将课程设计视为马尔可夫决策过程，对智能体与任务互动时知识的积累进行建模，并提出了一种近似执行最优策略的方法，以生成适用于每个智能体的个性化课程。文献[77]通过采用多代理邻近策略优化算法，成功开发了针对超视距空战的无人作战飞行器协同决策方法。文献[78]提出了一种多代理分层策略梯度算法，用于解决空对空对抗中的作战策略选择问题。该算法通过对抗性自我博弈学习掌握多种策略，并通过分层决策网络有效处理复杂和混合动作。

由以上研究可以看出，目前对多无人机战术决策问题的研究主要聚焦在特定场景中引入强化学习训练后决策逻辑的合理性^{[79][80]}，但是各种独立状态的组合形成了一个非常

大的决策空间，这会导致状态维度的爆炸^[81]的问题。于是研究人员提出了一个基于分层强化学习(Hierarchical Reinforcement Learning, HRL)的空战框架^[82]以降低强化学习决策方法的训练难度，这一框架将决策过程分为顶层策略和底层策略两部分，其中顶层策略根据敌机的相对位置和我机的状态，向底层策略发送宏观指令。底层策略负责直接操控无人机以响应这些宏观指令。目前采用分层强化学习来解决编队协同智能空战战术决策问题的相关研究较少。主要研究包括，文献[83]采用了结合分层架构和最大熵强化学习的方法，通过软演员-评论家算法^[84]训练顶层策略和元策略，并且该方法整合了专家知识和策略模块化。文献[85]对多无人机近距离自主空中战斗决策的问题进行研究，文章设计了一种改进的分层强化学习架构，使其能够处理不同规模编队的空战场景，其结合了循环软演员-评论家算法和竞争性自我对弈来学习子策略。文章所提的多智能体分层强化学习算法，为多飞机自主空中战斗提供了解决方案。文献[86]提出了一个分层多智能体强化学习框架，用于解决多个智能体的空对空战斗决策问题。该框架将决策过程分为两个层次，低层次策略控制个体单位的行动，高层指挥策略根据整体任务目标发布宏观指令。文献[87]引入了一种改进的分层机动决策架构并将其应用于双飞机近距离空中战斗场景中。文章结合了 Soft Critic-Actor 算法和竞争性自我对弈，以整合子策略的知识，并使用强化学习技术实现了近似的纳什均衡主策略。这项研究为未来智能编队空中战斗提供了解决方案，同时也为有人或无人飞机的协同作战提供了指导。

文献分析表明，在处理多无人机战术决策问题时，传统算法可能面临计算量大和实时性不足的挑战。强化学习方法在某些方面表现出色，但它们也面临着奖励稀疏和维度爆炸等问题。因此，在利用强化学习对无人机编队协同智能空战战术决策进行研究时，仍有许多关键问题需要进一步解决。为了解决分层强化学习中模型的不敏感性，终止函数设置不合理导致算法最终无法收敛到稳定状态的问题。本论文根据不同的任务设计了相应的元策略模型，并通过将该模型封装并输入环境状态返还动作，利用 Actor-Critic 网络训练终止函数，以完成无人机编队在空战场景中的作战任务需求，增强多无人机协同战术决策的自主化和智能化。

1.3 论文研究内容与创新点

1.3.1 论文研究内容

随着无人机技术在军事领域的广泛应用及人工智能技术的快速发展，研究多无人机协同智能战术决策变得尤为重要。智能化和自主化的无人机编队不仅能提升单机作战的效率，还能有效优化整体作战策略。因此，如何将人工智能技术有效融入多无人机协同空战战术决策，以增强其自主决策和协同作战能力，并在执行复杂空域任务时充分发挥潜能，已成为一个研究热点。

本研究根据无人机编队协同战术决策的需求，将研究内容分为三个部分。首先是抵

达作战区域的航路规划问题，采用改进的灰狼算法来规划无人机编队的自主航路。其次是在作战区域内的无人机机动决策问题，运用基于深度强化学习的方法来制定无人机的空战机动决策。最后是在作战区域内的无人机编队协同战术决策问题，本文基于分层强化学习算法完成无人机编队协同空战战术决策。本文研究内容如图 1-1 所示。

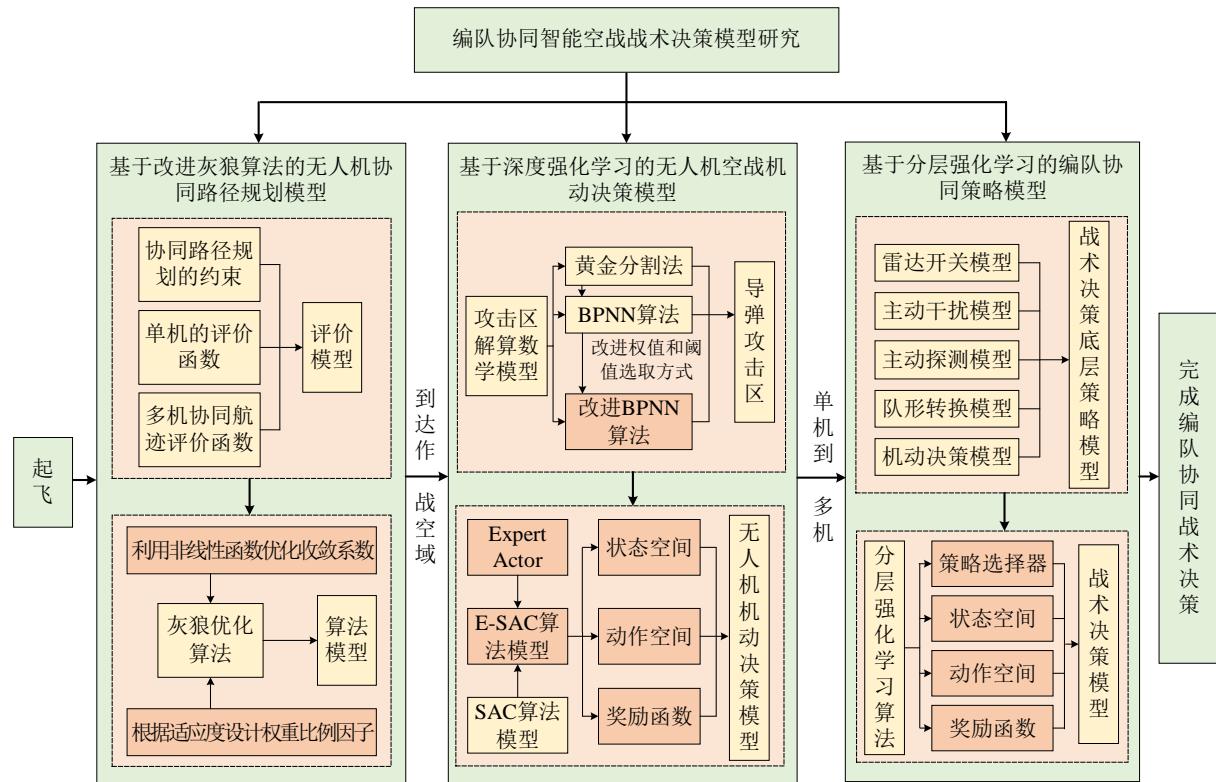


图 1-1 论文研究内容

论文具体研究内容如下：

(1) 基于改进灰狼算法的无人机协同航路规划模型

在抵达作战区域前，首先需要进行自主航路规划。针对灰狼算法在多无人机航路规划的不足，提出了一种面向多无人机的改进灰狼优化算法用于优化多机的航路规划问题。首先，对灰狼优化算法在多机航路规划的不足进行改进，利用非线性函数优化灰狼算法的收敛系数并提出面向多机的改进的灰狼优化算法；其次，基于 MATLAB 设计多无人机航路规划的仿真平台，模拟多无人机躲避威胁区并抵达任务目标点的作战任务；最后，将本文所提的 NI-GWO 算法代入仿真平台进行验证，并将其与传统的 GWO 算法和 MP-GWO 算法的多机航路规划性能进行比较并分析算法的鲁棒性。

(2) 基于深度强化学习的无人机空战机动决策模型

在抵达作战区域后，每架无人机都需要基于当前空战态势进行自主机动决策。针对传统 SAC 算法机动决策中状态空间大、训练不易收敛等问题，提出了一种 E-SAC 算法

用于无人机空战场景下的自主机动决策。首先，对导弹攻击区进行解算，基于 BP 神经网络实现导弹攻击区的快速拟合；其次建立无人机空战机动决策模型，提出一种专家知识经验嵌入的 E-SAC 算法。最后，基于 MaCA 仿真平台完成无人机机动决策策略的仿真，并将本文所提的 E-SAC 算法与传统的 SAC 算法进行对比验证算法性能。

(3) 基于分层强化学习的编队协同空战战术决策模型

在到达作战区域后，无人机编队需要根据当前的空战态势做出编队协同决策。针对多机空战中编队协同决策响应速度慢的问题，基于分层强化学习算法，提出了一种无人机编队协同空战策略模型。首先，建立无人机编队协同决策的元策略模型，包括基于专家经验的雷达开关策略、主动干扰策略、主动探测策略和队形转换策略，完成多维元策略的封装。其次基于 Option-Critic 算法的策略选择器进行改进，提出改进 Option-Critic 算法的策略选择器来训练终止函数，完成顶层策略选择器的构建；最后，基于 MaCA 仿真平台对本文构建的基于分层强化学习无人机编队协同策略模型进行验证，并将本文所提改进 Option-Critic 算法与传统的 Option-Critic 算法进行对比验证算法收敛速度。

1.3.2 论文创新点

论文在基于灰狼优化算法的无人机协同自主航路规划、基于深度强化学习的无人机机动决策和基于分层强化学习的编队协同空战战术决策三个方面进行研究，创新点如下：

(1) 提出了一种基于灰狼优化算法的多无人机航路规划方法，解决了多无人机协同航路规划问题。

针对传统灰狼算法在解决多无人机协同航路规划容易陷入局部最优的问题，提出了基于改进灰狼算法的多无人机协同航路规划方法。论文基于灰狼算法，设计了非线性函数来优化收敛系数，并通过计算灰狼的适应度来优化灰狼算法的更新位置，提升算法的探索能力。结果表明，提出的改进灰狼优化算法在求解无人机自主航路规划问题中具有有效性，且学习收敛速度更快。

(2) 提出了一种基于深度强化学习的无人机机动决策方法，解决了基于攻击区终端约束的无人机自主机动决策问题。

针对导弹攻击区终端约束的无人机自主机动决策问题，提出了基于深度强化学习的空战机动决策方法。论文基于 SAC 算法在连续状态空间-动作空间对无人机的自主机动决策进行了研究，设计了基于态势评估的奖励函数，并在此基础之上提出了 E-SAC 算法对自主机动决策进行研究，并利用解算得到的攻击区作为模型的终端约束。仿真结果表明，提出的方法对于求解攻击区终端约束的无人机自主机动决策问题中具有有效性，且改进后的方法提升了无人机机动决策训练的速度，同时避免陷入局部最优。

(3) 提出了基于分层强化学习的无人机编队协同空战战术决策方法，解决了多无人机空战协同战术决策问题。

针对无人机编队协同空战战术决策问题，提出了基于分层强化学习的空战战术决策方法。论文将空战任务进行分解，并利用专家经验模型设计了雷达开关、主动干扰、主动探测和队形转换的底层模型，结合(2)中设计的机动决策模型，通过分层强化学习让顶层策略选择器能够根据底层策略与环境的交互来学习如何选择最优的元策略，将不同底层策略联系起来，实现空战的全流程作战仿真。结果表明，所提方法在不同战场态势下的空战对抗中表现出了高度的灵活性和适应性。

1.4 论文组织结构

基于以上研究内容和创新点，论文分为六章进行论述，各章节的组织结构如图 1-2 所示：

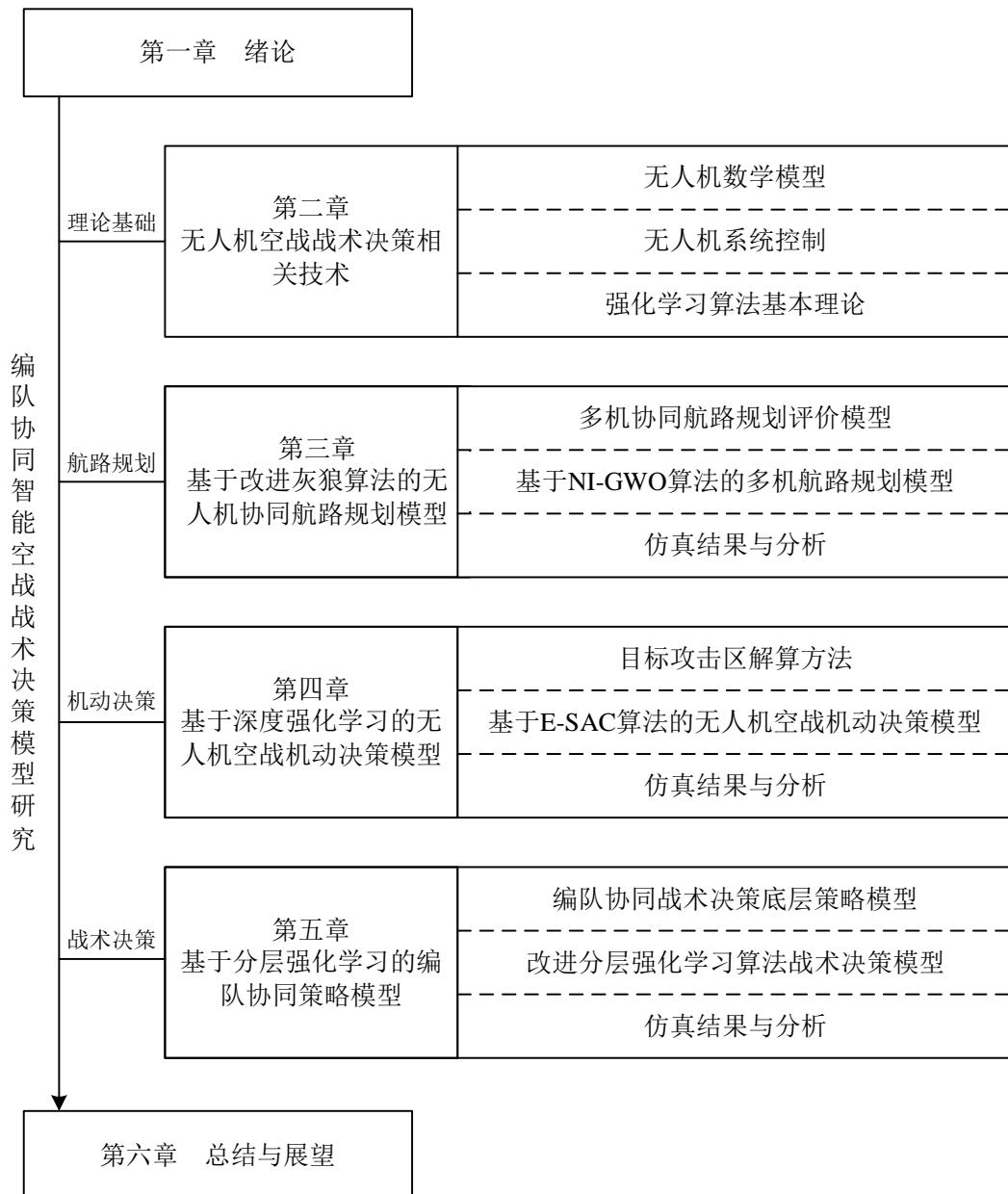


图 1-2 论文组织结构图

第一章 绪论

介绍了论文的研究背景及意义，对国内外多无人机协同航路规划、无人机空战机动决策和多无人机协同空战战术决策方法的研究现状进行了分析，同时阐述了论文的研究内容、组织架构和创新点。

第二章 无人机空战战术决策相关技术

引入了无人机数学模型，接着介绍了强化学习算法基本原理，其中包括了马尔可夫决策和半马尔可夫决策过程、基于价值学习和基于策略学习的强化学习方法、Actor-Critic 方法介绍，最后简述了分层强化学习算法的原理。

第三章 基于改进灰狼算法的无人机协同航路规划模型

首先建立了多机协同航路规划评价模型；其次基于灰狼优化算法，对其收敛系数和比例权重进行改进；最后进行仿真验证。

第四章 基于深度强化学习的无人机空战机动决策模型

首先根据导弹攻击区解算的相关模型，并采用神经网络进行快速拟合；其次设计了基于 E-SAC 算法的带有终端约束的机动决策模型；最后进行了仿真验证。

第五章 基于分层强化学习的编队协同策略模型

首先根据无人机编队协同全流程空战任务，建立了基于分层强化学习的元策略模型；其次基于分层强化学习算法提出改进的元策略选择器；最后在 MaCA 平台与敌机算法博弈，验证分层架构的有效性。

第六章 总结与展望

对无人机编队协同智能空战战术决策研究工作进行了总结，并提出了论文研究工作的后续研究内容。

1.5 本章小结

本章首先介绍了无人机编队协同智能空战战术决策的研究背景并分析其研究意义；其次分别对多无人机协同航路规划、无人机空战机动决策和多无人机协同空战战术决策的国内外研究现状进行了总结和分析；最后介绍了论文的主要研究内容、创新点以及本论文的组织结构。

第2章 无人机空战战术决策相关技术

本章首先介绍了无人机数学模型，包括无人机动力学模型和运动学模型，其次介绍了强化学习算法基本原理，包括了马尔可夫决策和半马尔可夫决策过程、基于价值学习和基于策略学习的强化学习方法和 Actor-Critic 方法，最后讨论了分层强化学习算法的基本思想，为后面章节奠定理论基础。

2.1 无人机数学模型

构建无人机数学模型时，考虑到实际飞行中系统的复杂性和多变因素，采取了一系列简化的假设以便于分析^[88]：

- 1) 忽略地球自转和曲率效应，包括离心加速度和科里奥利加速度的影响，从而简化地球物理环境对模型的复杂性；
- 2) 假定重力加速度在无人机的不同飞行高度上保持恒定，避免了高度变化带来的计算复杂度；
- 3) 将无人机视作一个质量和质心位置在燃料消耗过程中保持不变的刚体，简化了动力系统和能源消耗的影响；
- 4) 忽略大气气流等气象条件对无人机飞行状态的影响，使模型更加专注于无人机本身的运动特性。

基于上述假设，本文构建了无人机在东北天惯性坐标系中的三自由度质点模型，如图 2-1 所示。

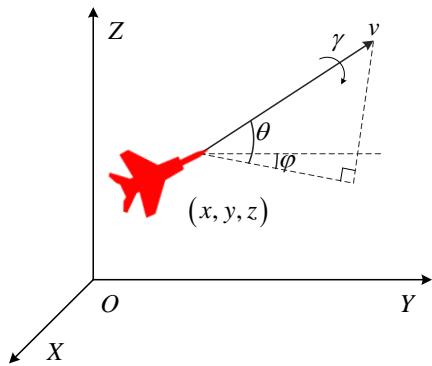


图 2-1 无人机三自由度质点模型示意图

如图 2-1 所示，该方法在简化现实飞行条件的同时，仍然能够有效地捕捉无人机的关键运动特性，为理论分析和应用开发提供了坚实的基础。

2.1.1 无人机动力学模型

无人机动力学模型是通过数学描述无人机动力学特性的工具，其建立基于 2.1 的一

系列假设，如式(2-1)所示。

$$\begin{cases} \frac{dv}{dt} = g(n_x - \sin \theta) \\ \frac{d\theta}{dt} = \frac{g}{v}(n_z \cos \gamma - \cos \theta) \\ \frac{d\varphi}{dt} = -\frac{n_z g \sin \gamma}{v \cos \theta} \end{cases} \quad (2-1)$$

其中， v 为无人机在参考系中的速度； θ 为无人机俯仰角，表示无人机速度与水平面之间夹角； φ 为航向角，表示无人机速度在水平面的投影与 Y 轴（正北方向）之间的夹角； γ 为滚转角，可以用来控制无人机航向角 φ ； g 为重力加速度； n_x 为无人机切向过载，可以用来控制无人机飞行速度 v ； n_z 为无人机法向过载，可以用来控制无人机航迹倾斜角 θ 。

2.1.2 无人机运动学模型

无人机运动学模型是以数学方式描述无人机运动规律的工具。基于上述一系列假设的基础，建立了无人机三自由度的运动学方程组：

$$\begin{cases} v_x = \frac{dx}{dt} = v \cos \theta \cos \varphi \\ v_y = \frac{dy}{dt} = v \cos \theta \sin \varphi \\ v_z = \frac{dz}{dt} = v \sin \theta \end{cases} \quad (2-2)$$

其中， x 、 y 、 z 分别为无人机在东北天惯性坐标系中 X 轴、 Y 轴、 Z 轴的坐标。

2.2 强化学习理论基础

强化学习（Reinforcement Learning, RL）是一门重要的机器学习分支，经历了长期的研究和发展，如今已在多领域取得广泛的应用，包括但不限于智能推荐^[89-98]、游戏攻略^[99-106]、生物医药^[107-110]、网络安全^[111]、自动驾驶^[112]、机器人技术^[113-117]、自然语言处理^[118-124]等广泛领域。在智能推荐领域中，强化学习的应用使得网络平台可以根据每个用户的特殊需求，按顺序推荐个性化学习路径；在游戏攻略领域中，强化学习算法可以让智能体通过对游戏规则的学习，从而提升其推理能力和决策能力，从而达成游戏所需目标；在生物医药领域中，强化学习算法可以帮助人类发现新的药物或材料，提升疾病预测的准确度；在网络上安全领域中，强化学习算法可以作为检测病毒或者恶意攻击软件的入侵的关键技术，降低潜在的网络威胁对人们做出的影响；在自动驾驶领域中，强化学习算法在机器人技术中，强化学习的应用赋予机器人互动的能力，使其能够不断优化决策策略，从而在与环境互动中提高行为选择的效力；在自然语言处理领域中，强化

学习算法辅助计算机实现有效的决策，从而生成两罐的文字以及准确地回复内容，提升了人机交互的能力，实现人机的无障碍沟通等。总之，强化学习的进步，推动着不同领域的发展，也促使人类具有更好的解决问题的方法^[125-126]。

强化学习的核心理念在于通过智能体与环境的互动来寻求最佳的决策策略。它依赖反馈信息来增强有效和有利行为，同时减弱无效和不利的行动，最终达到习得最佳策略以实现设定目标的目的，其原理图如图 2-2 所示。

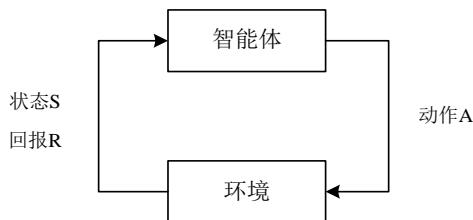


图 2-2 强化学习中智能体与环境之间交互的示意图

如图 2-2 所示，强化学习涵盖了四个核心要素：

策略 (Policy)：智能体在当前环境下可采取的行为的映射；

回报信号 (Reward Signal)：智能体在选择一个行为后所能获得的反馈，不同状态下执行相同的行为也可能会产生不同的奖励回报值，而强化学习的目标就是找到累积奖励值最大的方式，该值得大小可以显示出某个决策当下的优劣；

值函数 (Value Function)：某个状态的价值，即从当前状态开始，经过一系列动作，到达中止状态所累计获得的奖励，该值的大小可以显示出某个决策长期的优劣；

环境模型 (Environment Model)：智能体可以利用环境模型来预测环境所带来的响应结果。

2.2.1 MDP 与 SMDP

(1) 马尔可夫决策过程

马尔可夫决策过程(Markov Decision Process, MDP)是现实中抽象出来的一个概念，也是强化学习的数学模型，目前被广泛应用于各个领域^[127]，如决策设计^[128-133]、目标跟踪^[134-140]、智能控制^[141-144]、路况识别^[145-148]、网络通信^[149-151]、航路规划^[152-154]、目标识别^[155-161]、声呐通信^[162-167]、导航定位^[168-169]。

1957 年贝尔曼^[170]提出了马尔可夫决策过程，即在智能体与环境的互动中学习并实现目标的过程，是一个无记忆的随机过程。智能体基于当前环境状态选择动作，环境响应这些动作并引导智能体进入新状态。同时，环境提供奖励信号，表示智能体通过选择动作来追求最大化的数值奖励。这种连续的互动过程定义了马尔可夫决策过程。智能体通过不断调整其策略以最大化长期奖励来学习如何在特定环境中做出决策。因此，MDP 提供了一种强大的框架，可用于建模和解决各种决策问题。如图 2-3 所示。

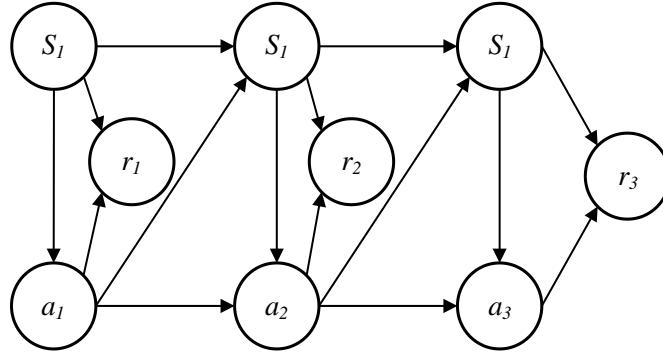


图 2-3 马尔可夫决策过程表示图

在马尔可夫决策中主要关注智能体的状态(State)、行为(Action)、最终所获得的奖励(Reward)和概率(Probability)，可用五元组 $\langle S, A, R, P, \gamma \rangle$ 进行描述，各元素的具体含义如下：

S (State Space) 表示智能体所观察到的所有环境特征数量，即状态空间，通常用 $S = \{s_1, s_2, \dots, s_m\}$ 来表示，智能体在环境中遇到的任何情形都会有一个状态与之对应。

A (Action Space) 表示智能体进行决策可选择执行的所有动作数量，即动作空间，通常用 $A = \{a_1, a_2, \dots, a_n\}$ 来表示，它的作用往往是改变状态。

R (Reward Function) 表示在某个状态下智能体与环境互动所获得的反馈，即回报函数，如果奖励值为正，代表当前行为是被鼓励的；奖励如果为负，代表当前行为应当被惩罚。**R** 在状态 s 下，当智能体选择执行动作 a ，然后过渡到另一个状态 s' 时，环境为此行为提供的即时奖励值可以用符号表示为

$$R_{ss'}^a = E[r_{t+1} | S_t = s, A_t = a, S_{t+1} = s'] \quad (2-1)$$

其中， $R_{ss'}^a$ 为智能体在状态 s 执行动作 a 后，进入下一个状态 s' 的奖励函数； r_{t+1} 为智能体在时刻 $t+1$ 获得的即时奖励； S_{t+1} 为智能体在时刻 $t+1$ 的状态； A_t 为智能体在 t 时刻执行的动作； S_t 为智能体在时刻 t 的状态。

P (Transition Probability) 表示状态转移概率，即智能体在某一状态 s 下采取特定动作 a 后，状态转移为另一状态 s' 的概率，如式所示：

$$P_{ss'}^a = P[S_{t+1} = s' | S_t = s, A_t = a] \quad (2-2)$$

其中， $P_{ss'}^a$ 为智能体在状态 s 执行动作 a 后，进入下一个状态 s' 的奖励函数； S_{t+1} 为智能体在时刻 $t+1$ 获得的即时奖励； A_t 为智能体在 t 时刻执行的动作； S_t 为智能体在时刻 t 的状态。

γ 为折扣因子，其范围为 $0 < \gamma < 1$ 。在马尔可夫决策过程中，折扣因子的存在可以让优化函数收敛于有限值，从而避免了无限循环。通常由于远期收益其不确定性较大，因此，对于越早获得的奖励，将被认为越重要，越晚获得的奖励，其衰减越严重，这导致

智能体在做出决策时更加侧重于近期的回报， $\gamma=0$ 表示只关注最近奖励， $\gamma=1$ 表示会关注所有的价值。

每个时刻智能体执行一个动作并获得相应奖励，奖励函数和状态转移矩阵共同构成了环境，而 P 和 R 则定义了模型。马尔可夫决策过程大致可以描述为：首先在 $t=0$ 时刻，初始状态 s_0 以概率 $P(s_0)$ 进行随机初始化；随后，智能体根据当前状态 s_t 选择行动 a_t ；环境响应智能体的动作，产生奖励 r_t ，条件概率为 $R(\cdot|s_t, a_t)$ ；同时，环境根据条件概率 $P(\cdot|s_t, a_t)$ 确定下一时刻的状态 s_{t+1} ；智能体接收奖励 r_t 和状态 s_{t+1} ；这一过程不断循环进行。

(2) 半马尔可夫决策过程

在 MDP 中，每个状态的转换和奖励的获得发生在动作执行后的下一个时间步。这一假设简化了决策过程的建模，但在许多现实世界的应用中，它不足以精确地描述动作的持续影响，由此产生了半马尔可夫决策过程（Semi-Markov Decision Process, SMDP）扩展了 MDP。在 SMDP 中，决策点不再是按固定时间间隔出现，而是可以基于更复杂的时间模型，这意味着在两次连续的决策之间，可以有不同的时间跨度，这更贴近那些动作结果需要时间发展的现实情况。两者的状态区别如图 2-4 所示。

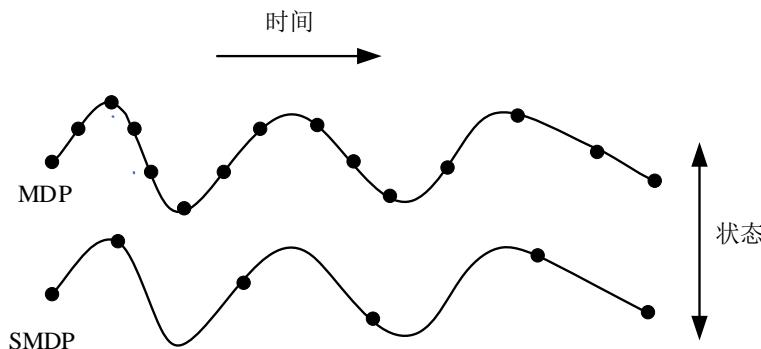


图 2-4 MDP 与 SMDP 状态比较

可以看出 SMDP 是 MDP 的一般化过程，SMDP 是将标准 MDP 中的状态 s 执行动作 a 到达状态 s' 的过程中固定单位时间步长变为可变时间步长。设置时间步长为 N ，MDP 的概率转移函数 $P(s, a, s')$ 和期望奖励 $R(s, a, s')$ 可以相应的扩展到 SMDP 中为 $P(s', N | s, a)$ 和 $R(s', N | s, a)$ ，根据贝尔曼方程，确定性的策略 π 下的状态值函数 $V^\pi(s)$ 可以定义为执行策略 π 在状态 s 下所得到的奖励和未来预期奖励的加权总和，即：

$$V^\pi(s) = \bar{R}(s, \pi(s)) + \sum_{s', N} P(s', N | s, \pi(s)) \gamma^N V^\pi(s') \quad (2-3)$$

其中， $\bar{R}(s, \pi(s))$ 表示状态 s 下执行策略 $\pi(s)$ 后的平均即时奖励，该奖励与转移概率 P 和奖励函数 R 密切相关。

SMDP 的这种特性使其成为模拟和解决那些涉及延迟效应或长期规划决策的理想

工具。例如，在智能空战决策的影响可能不会立即显现，而是随着时间的推移逐渐展现。通过引入动作的持续时间和复杂的时间依赖性，SMDP 提供了一个更为细腻和适应性强的框架，用于捕捉和优化这些类型的顺序决策问题。本论文的顶层策略采用了半马尔可夫决策的思想进行设计与训练。

2.2.2 基于价值的学习

在价值学习中涉及到值函数，包括动作价值函数 $Q_\pi(s, a)$ ，最优动作价值函数 $Q_*(s, a)$ ^[171]。

定义 U_t 为 t 时刻之后所有奖励之和，即：

$$U_t = \sum_{x=t}^H \gamma^x r_x \quad (2-4)$$

动作价值函数定义为：

$$Q_\pi(s_t, a_t) = E[U_t | S_t = s_t, A_t = a_t] \quad (2-5)$$

即动作价值函数为在已知当前状态为 s_t 并采取动作 a_t 的情况下对未来累计奖励的期望。动作价值函数的值与策略 π 有关，定义最优动作价值函数为：

$$Q_*(s_t, a_t) = \max_\pi Q_\pi(s_t, a_t) \quad (2-6)$$

最优动作价值函数对应的是最佳策略的动作价值函数，即对应：

$$\pi^* = \arg \max_\pi Q_\pi(s_t, a_t) \quad (2-7)$$

最优动作价值函数对应最优策略，如果可以知道最优动作价值函数，那么可以使用最优动作价值函数对当前 s_t 状态下可选择的动作进行打分，即计算动作对应的累计奖励，并选择得分最高的动作^[172]。

大多数的强化学习都是无模型的，本文所研究的强化学习算法也都是基于无模型的强化学习。对于无模型强化学习，其值函数更新方法最常用的是时间差分算法 (Temporal Difference, TD)，该算法基于式(2-8)的最优贝尔曼方程并使用 TD 误差来更新网络参数。

$$Q_*(s_t, a_t) = E_{S_{t+1} \sim T(s_t, a_t)} [R_t + \gamma \cdot \max_{A \in \mathcal{A}} Q_*(S_{t+1}, A) | S_t = s_t, A_t = a_t] \quad (2-8)$$

对式(2-8)做蒙特卡洛近似后，得：

$$Q_*(s_t, a_t) \approx r_t + \gamma \cdot \max_{a \in \mathcal{A}} Q_*(s_{t+1}, a) \quad (2-9)$$

用神经网络 $q(s_t, a_t; \theta)$ 拟合最优动作价值函数，则 TD 误差可以表示为：

$$\delta_t = q(s_t, a_t; \theta) - (r_t + \gamma \cdot \max_{a \in \mathcal{A}} q(s_{t+1}, a; \theta)) \quad (2-10)$$

其中， θ 为神经网络参数。定义目标函数为均方 TD 误差：

$$J(\omega) = \frac{1}{2} \delta_t^2 \quad (2-11)$$

最后，使用梯度下降更新神经网络 $q(s, a; \theta)$ 的参数。

2.2.3 基于策略的学习

不同于价值学习，策略学习通过求解一个优化问题，直接学习最优策略函数或是对最优策略函数的近似如策略网络，策略学习通过策略梯度来更新模型^[173]。对于式(2-5)，将 t 时刻状态 s_t 和 t 时刻动作 a_t 作为已知观测值，将 $t+1$ 时刻后的状态和动作看成未知变量，并求期望得状态价值函数为：

$$V_\pi(s_t) = E_{A_t \sim \pi(\cdot | s_t; \omega)} [Q_\pi(s_t, A_t)] \quad (2-12)$$

其中， ω 为策略网络参数， $V_\pi(s_t)$ 用来衡量当前状态 s_t 的好坏， $V_\pi(s_t)$ 越大。说明累计奖励 U_t 越大。策略 π 越好，即参数 θ 越好， $V_\pi(s_t)$ 也会越大。

因此，定义状态价值函数 $V_\pi(S)$ 的期望为目标函数：

$$J(\omega) = E_S [V_\pi(S)] \quad (2-13)$$

故策略学习可以描述为一个 $J(\omega)$ 最化的优化问题，即：

$$\max_\theta J(\omega) \quad (2-14)$$

根据策略梯度定理，策略梯度可以表示为：

$$\frac{\partial J(\omega)}{\partial \omega} = E_S \left[E_{A \sim \pi(\cdot | S; \omega)} \left[\frac{\partial \ln \pi(A | S; \omega)}{\partial \omega} \cdot Q_\pi(S, A) \right] \right] \quad (2-15)$$

最后，使用梯度上升对策略网络参数进行更新。

2.2.4 Actor-Critic 算法

“演员-评论家”方法(Actor-Critic, AC) 结合价值学习和策略学习算法，使用一个神经网络 $\pi(a | s; \omega)$ 作为策略网络输出动作，其中 ω 为策略网络参数。同时，使用另一个神经网络 $q(s, a; \theta)$ 作为价值网络用来近似动作价值函数 $Q_\pi(s, a)$ ，其中 θ 为价值网络参数^[174]。与 2.2.2 节中价值学习中的近似不同，此处价值网络用来近似动作价值函数而不是最优动作价值函数。以离散动作空间为例，设动作维数为 n ，价值网络输出 n 维向量，向量的每一维代表一个动作的评分，以此来评价动作的好坏。Actor-Critic 方法学习的基本流程如图 2-5 所示。

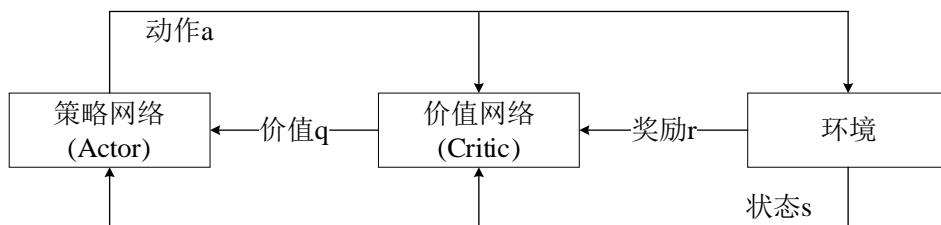


图 2-5 Actor-Critic 算法基本流程

(1) 价值网络更新

Actor-Critic 方法在训练过程中，需要对价值网络 $q(s, a; \theta)$ 进行更新，使价值网络输出的动作价值能够更加准确地反映出真实动作价值，价值网络基于 SARSA 算法进行更新，SARSA 算法基于式(2-16)的贝尔曼方程更新网络：

$$Q_{\pi}(S_t, A_t) = E_{S_{t+1}, A_{t+1}} [R_t + \gamma \cdot Q_{\pi}(S_{t+1}, A_{t+1}) | S_t = s_t, A_t = a_t] \quad (2-16)$$

在 t 时刻，价值网络输出动作价值 $q(s_t, a_t; \theta)$ 在 $t+1$ 时刻，实际观测到 r ， s_{t+1} ， a_{t+1} ，价值网络输出动作价值 $q(s_{t+1}, a_{t+1}; \theta)$ ，由此 TD 误差可以表示为：

$$\delta_t = q(s_t, a_t; \theta) - (r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \theta)) \quad (2-17)$$

价值网络更新目标函数定义为：

$$J(\theta) = \frac{1}{2} \delta_t^2 \quad (2-18)$$

最后，根据梯度下降算法更新网络参数：

$$\theta \leftarrow \theta - \alpha \cdot \delta_t \cdot \nabla_{\theta} q(s_t, a_t; \theta) \quad (2-19)$$

其中， α 为价值网络学习率。

(2) 策略网络更新

策略网络 $\pi(a | s; \omega)$ 基于式(2-15)的策略梯度进行更新。在 t 时刻，策略网络根据观测到的状态 s_t 输出动作 a_t ，同时，使用价值网络输出对动作的评价 $q(s, a; \theta)$ ，将其作为式(2-15)中 $Q_{\pi}(S, A)$ 的近似，得到近似策略梯度表示如式所示：

$$\mathbf{g}(s_t, a_t; \omega) = \nabla_{\omega} \ln \pi(a_t | s_t; \omega) \cdot q(s_t, a_t; \theta) \quad (2-20)$$

最后，基于梯度上升更新策略网络参数 ω 可以表示如式所示：

$$\omega \leftarrow \omega + \beta \cdot g(s_t, a_t; \omega) \quad (2-21)$$

其中， β 为策略网络学习率。

2.2.5 分层强化学习

分层强化学习是强化学习领域中的一个分支。传统强化学习通过与环境的交互，进行试错，从而不断优化策略。但是强化学习的一个重要不足就是维数灾难，当系统状态的维度增加时，需要训练的参数数量会随之进行指数增长，这会消耗大量的计算和存储资源。

分层强化学习将复杂问题分解成若干子问题，子问题分解有两种方法：

- (1) 将复杂任务拆分成多个子任务，对每个子任务分别求解；
- (2) 将前一个子问题的策略加入到下一个子问题的策略学习中。

分层强化学习核心思想是通过算法结构设计对策略和价值函数施加各种限制，或者使用本身就可以开发这种限制的算法。

常见的分层强化学习方法可以分成以下 3 类：基于选项(Options)的学习、基于分层

局部策略(Hierarchical partial policy)的学习和基于子任务(Sub-task)的学习。接下来对这三类算法作简要介绍。

(1) 基于选项的分层强化学习

选项^[175]是一个由三个元素构成的元组 $\langle 1, \pi, \beta \rangle$ 。其中 $\pi: S \times A \rightarrow [0,1]$ 代表策略，是一个基于状态空间和动作空间的概率分布函数； $\beta: S \rightarrow [0,1]$ 是终止条件， $\beta(s)$ 表示状态 s 有 $\beta(s)$ 的概率终止并退出当前选项； $I \subseteq S$ 表示选项的初始状态。

一个基于选项的学习过程如下：智能体首先初始化在某一状态，然后其选择某一选项，然后按照这一选项的策略 π 选择一个动作或者其他选项，然后执行动作或者进入新的选项，继续循环选择或者终止。

对于每一个状态 s ，可用的选项用 $O(s)$ 来表示， $\mu(s, o)$ 表示在状态 s 下，进入选项 o 的概率。基于选项的 Q-learning 更新方程如式所示：

$$Q(s, o) \leftarrow Q(s, o) + \alpha \left[r + \gamma^k \max_{o' \in O_s} Q(s', o') - Q(s, o) \right] \quad (2-22)$$

(2) 基于分层局部策略的学习

这一类方法以分层抽象机 (Hierarchical abstract machines, HAMs)^[176]为代表，其核心思想是用抽象机对可以采取的策略进行限制。

HAMs 拥有有限个抽象机的集合 H ，每个 $h \subseteq H$ 都是一个 HAM 抽象机，它包括以下 3 部分：抽象机状态，状态转移方程，初始化方程。

一个抽象机的状态空间 S 有 4 种不同的状态，分别是动作、调用、选择和终止。初始化方程决定了抽象机的初始状态。当抽象机在动作和调用状态下时，状态转移方程决定抽象机的下一个状态，并获得相应奖励。

在一个 HAM 的状态空间中，动作意味着执行某一动作，调用的作用是选择下一个抽象机，选择则是非确定性地选择抽象机的下一个状态，终止则是终止当前抽象机，并恢复到调用前的抽象机。此时 HAM Q-learning 的更新方程如式所示：

$$Q([s_c, m_c], a) \leftarrow Q([s_c, m_c], a) + \alpha \left[r_c + \beta_c V([t, n]) - Q([s_c, m_c], a) \right] \quad (2-23)$$

这里的 Q 值是它的扩展形式， $Q([s, m], a)$ 表示了在以下条件下的 Q 值：环境状态 s ，抽样机状态 m ，以及当 m 是选择时所执行的动作 a 。下标 c 表示选择点。

(3) 基于子任务的学习

基于子任务的学习以 MAXQ 价值函数分解(MAXQ value function decomposition)方法为代表。它由 Dietterich^[177]在 2000 年提出。这种方法的思想是将一个马可夫决策过程 M 分解成多个子任务，解决了每个子任务也就解决了原本的复杂问题。每个子任务都有终止状态 T ，子动作空间 A 。子任务的目标就是从某一初始状态不断执行动作，进入新状态，直至终止。

2.3 本章小结

本章主要介绍了无人机编队协同智能空战战术决策问题的相关技术。首先，介绍了无人机的数学模型；然后，阐述了强化学习算法的基本原理，包括马尔可夫决策和半马尔可夫决策过程、基于价值学习和基于策略学习的方法和 Actor-Critic 方法；最后，本章探讨了分层强化学习算法的核心思想，为后续章节中的算法应用奠定了坚实的理论基础。

第3章 基于改进灰狼算法的无人机协同航路规划模型

无人机编队的航路规划是完成与敌机对抗任务的第一步，无人机编队需要基于规划好的路径飞抵作战空域后开始战术决策。本章以灰狼算法作为研究理论工具，研究了在具有大量威胁区的环境下无人机编队航路规划的方法。首先，介绍了多无人机航路规划的模型；其次，对灰狼算法进行改进，提出了NI-GWO算法；最后通过设计多组仿真实验，以验证该算法的有效性。

3.1 问题分析

无人机编队在进行自主机动决策前需要从起飞点出发飞往作战空域，在这一阶段，无人机需要规避敌方威胁区并进行航路规划前往目标点。无人机编队的航路规划是指无人机编队在执行任务时，基于环境、威胁等因素，不依赖于操作人员，引导无人机编队到达目标点的过程。无人机编队航路规划是无人机进入作战区域前的关键一环。

针对无人机编队航路规划问题，本文将无人机在进入作战区域前需要规避的敌方威胁区简化为火控威胁区和雷达威胁区，将无人机编队从起始点到目标点的航路规划作为求解目标。本章利用改进的灰狼优化算法完成多无人机航路规划，本章具体研究内容如图3-1所示。

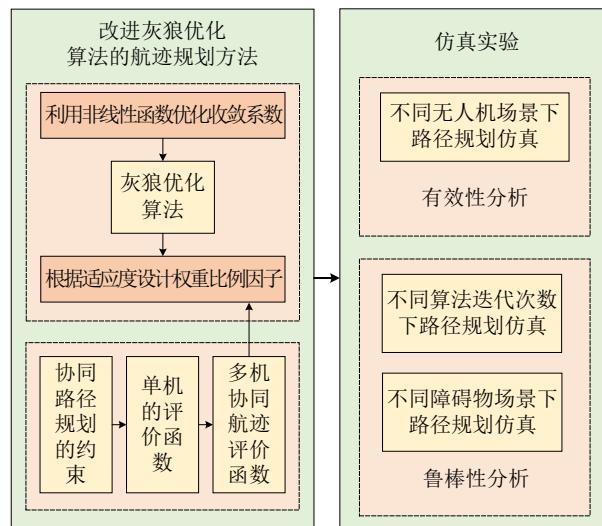


图 3-1 基于改进灰狼算法的无人机协同航路规划模型内容框图

3.2 多机协同航路规划模型

设 $V = \{V_i, i = 1, 2, \dots, N_v\}$ 为执行任务的多无人作战飞机集合， $T = \{T_i, i = 1, 2, \dots, N_v\}$ 为与各无人机相对应的目标所构成的集合 $M = \{M_j, j = 1, 2, \dots, N_M\}$ 为敌方威胁集合，任务区 E 为 N_r 行 N_c 列的栅格环境。对单个无人作战飞机来说，从初始状态(起始点)到最终状态(目标点)是由一系列航迹点组成，按规则连接起始点至目标点之间的所有航迹点即得

航迹。设 V_i 最大航程为 $L_{\max}^{(i)}$ ，最大转弯角为 $\theta_{\max}^{(i)}$ ，速度范围为 $[v_{\min}^{(i)}, v_{\max}^{(i)}]$ ，任务起点为 S_i ，目标点为 G_i ，航迹节点为 P_{i-k} ($k=1, 2, \dots, n-1$)，所以 V_i 的航迹可表示为 $S_i \xrightarrow{\Gamma(q)} P_{i-1} \cdots P_{i-(n-1)} \xrightarrow{\Gamma(q)} G_i$ ，其中 $\Gamma(q)$ 表示约束条件，表示 q 航迹约束参数。

3.2.1 协同约束分析

协同约束是指为了保证航迹在满足单机航迹约束条件的同时，使整个多无人作战飞机编队中各个无人机能够顺利完成任务。

(1) 空间协同约束

空间协同约束又叫无碰撞约束^[178]，要求多无人作战飞机之间的最小距离不小于最小安全飞行距离，即：

$$d_{i-j} \geq d_s \quad (3-1)$$

式中 d_{i-j} 为飞行过程中第 i 个多无人作战飞机与第 j 个多无人作战飞机的航迹点之间最小的距离， d_s 为多无人作战飞机之间最小安全飞行距离。

(2) 时间协同约束

文献^[179]和^[180]中对时间协同约束条件的定义方法的求解运算量比较大，不利于协同航迹规划的实时性。因此，文中采用步骤较少、计算量较少的一种方法，即设置了多无人作战飞机到达终点的指令时间和求解其到达目标终点的时间范围，通过判断二者的关系，来确定时间代价函数。

设多无人作战飞机到达目标终点的指令时间为 t_c ，通过指令时间约束保证多无人作战飞机在时间上协同；根据多无人作战飞机的速度范围与航迹长度可以求得多无人作战飞机到达目标终点的实际时间范围 $[t_{\min}^i, t_{\max}^i]$ ，则：

$$t_{\min}^i = \frac{L_i}{v_{\max}^i}, t_{\max}^i = \frac{L_i}{v_{\min}^i} \quad (3-2)$$

其中， v_{\max}^i 与 v_{\min}^i 分别为多无人作战飞机最大速度和最小速度； L_i 为第 i 个多无人作战飞机的航迹长度。

若 $t_{\min}^i \leq t_c \leq t_{\max}^i$ ，则第 i 个多无人作战飞机能按照规定的时间完成任务；否则不能完成任务，没有达到编队的协同性，需要对时间代价进行计算，具体计算方法后面将会给出。

3.2.2 单机航迹评价函数

无人机航迹代价函数一般也可以作为航迹评价的指标函数，主要包括燃油代价、高度代价和威胁代价^[181-183]。航迹代价的计算采用如下公式：

$$J = \sum_{i=1}^n (w_1 f_O(l_i) + w_2 f_{T-i}) \quad (3-3)$$

其中， l_i 表示第*i*段航迹的长度， f_o 表示关于航迹长度的燃油代价函数， f_{T-i} 为第*i*段航迹受到的威胁代价，其主要取决于多无人作战飞机是否处于威胁区域或在威胁区域的什么位置； w_1 和 w_2 为权系数，且 $w_1 + w_2 = 1$ 。

3.2.3 多机协同航迹评价函数

多无人机协同航迹评价函数主要在单机基础上结合多机在时域与空域上的协同，主要考虑时间和空间代价。协同航迹评价函数的计算公式如式所示：

$$J = \sum_{i=1}^m (w_1 f_{o-i} + w_2 f_{T-i} + w_3 f_{m-i} + w_4 f_{c-i}) \quad (3-4)$$

其中， m 为多无人作战飞机的数量； f_{o-i} 、 f_{T-i} 、 f_{m-i} 和 f_{c-i} 分别表示第*i*个无人机的燃油、威胁、时间和碰撞代价； w_i 为各个代价的权重。通过 3.2.1 节中时间协同约束的分析，时间代价的表示如式所示：

$$f_{m-i} = \begin{cases} 0 & , t_{\min^i} \leq t_c \leq t_{\max^i} \\ |t_i - t_c| & , t_{\min^i} > t_c \text{ or } t_{\max^i} < t_c \end{cases} \quad (3-5)$$

其中， t_i 为第*i*个多无人作战飞机达到目标终点实际时间。对于碰撞代价计算，以第*i*个多无人作战飞机为例，首先计算该多无人作战飞机在各个航迹点上当前时间与前*i-1*个多无人作战飞机的距离，如果该距离小于最小安全飞行距离，则记 1 次碰撞，依次对整条航迹的所有航迹点进行碰撞计数，并求和得到该多无人作战飞机航迹的总碰撞次数。计算公式如式所示：

$$f_{c-i} = \begin{cases} 1 & , d_{i-j} < d_s \\ 0 & , d_{i-j} \geq d_s \end{cases} \quad (3-6)$$

其中， $j \leq i-1$ ，指第*i*个多无人作战飞机与前*i-1*比较。

3.3 基于改进灰狼优化算法的多无人机航路规划模型

3.3.1 灰狼优化算法

灰狼优化算法(Grey Wolf Optimizer, GWO)作为基于种群的优化算法之一，是一种模拟自然界灰狼种群等级制度和捕食猎物行为的算法，该算法基于狼群之间协作的机制来达到优化的目的。每匹狼在群体中都有自己的角色，它们遵循自然规律，形成一个严格的等级制度，如图 3-2 所示。

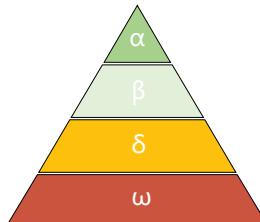


图 3-2 灰狼种群等级制度

灰狼等级的首层是灰狼部落的最高领袖，被称为 Alpha 狼，它负责管理狼群，对捕

猎事务进行总体谋划和决定；灰狼社会等级的第 2 层是狼群的第二领袖，被称为 Beta 狼，它在 Alpha 狼进行总体谋划和决定的过程中主要起协助作用，在灰狼种群的支配权中仅次于 Alpha 狼；灰狼社会等级的第 3 层是 Delta 狼，也被称为侦察兵，负责侦查、放哨、看护等事务；灰狼社会等级的第 4 层在灰狼部落里地位最低，被称为 Omega 狼，它对平衡灰狼部落内部关系至关重要。前 3 个等级依次是适应度最好的 3 个解，分别定义为最优解、次优解和次次优解，并且这 3 组解指导其他狼(Omega)向着目标搜索。在优化过程中，狼群更新 Alpha、Beta、Delta 和 Omega 的位置。

在狼群的捕食过程中，包括包围、狩猎、攻击和搜寻四个步骤，在追踪并确定猎物的位置后，其他个体在前三头狼的带领下，以他们的位置为基准不断更新自身位置，逐渐逼近猎物直至完成捕食行为，具体过程如图 3-3 所示。

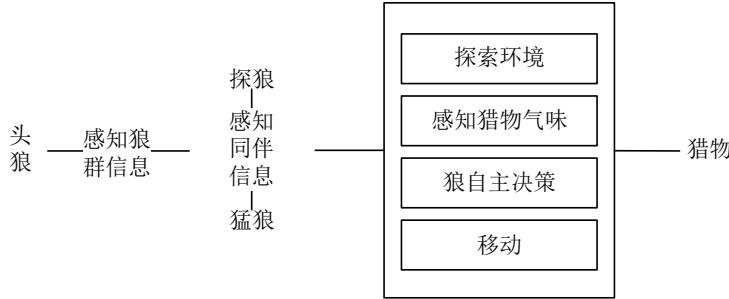


图 3-3 狼群捕猎模型

(1) 包围

在狩猎时，灰狼会包围猎物。定量地模拟灰狼的包围行为，提出了以下方程：

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(i) - \vec{X}(i) \right| \quad (3-7)$$

$$\vec{X}(i+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (3-8)$$

其中， i 代表当前迭代数， \vec{X}_p 为猎物的位置向量， \vec{X} 为灰狼的位置向量。 \vec{A} 和 \vec{C} 为系数向量，其计算方式如下：

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3-9)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (3-10)$$

其中，收敛因子 \vec{a} 的各个分量在迭代过程中，以线性的方式从 2 减少到 0，而 r_1 和 r_2 是 $[0,1]$ 范围内的随机向量。更新参数 a 的方程如下：

$$\vec{a} = 2 - i \left(\frac{2}{I} \right) \quad (3-11)$$

其中， i 代表最新的迭代次数， I 是迭代总数。

二维位置向量和多个潜在位置如图 3-4 所示。灰狼可以根据猎物的位置 $X_{(i)}$ 更新其位置。通过改变 \vec{A} 和 \vec{C} 向量的值，可以在当前位置附近识别出最佳智能体周围的其他位

置。值得注意的是，由于随机向量 \vec{r}_1 和 \vec{r}_2 ，狼可以到达如图 3-4 所示中显示的点之间的任何位置。因此，灰狼可以使用方程式(3-7)和(3-8)在猎物周围的区域内的任何位置更新其位置。

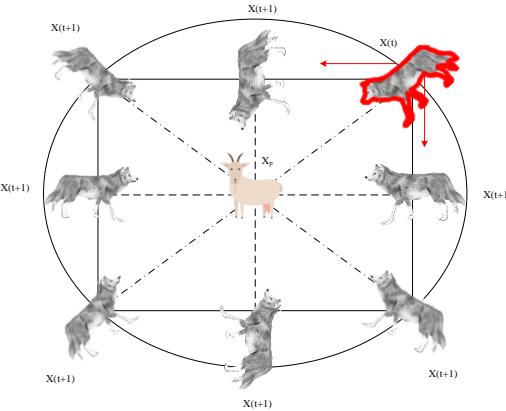


图 3-4 二维位置向量和潜在的未来位置

(2) 狩猎

灰狼有能力定位猎物并追捕它们。通常由 Alpha 狼领导狩猎，尽管有时狩猎可能由 Beta 和 Delta 狼完成。在一般的搜索环境中，最佳猎物是不清楚的。为了在数学上模拟灰狼的狩猎行为，假设 Alpha、Beta 和 Delta 狼对猎物可能位置的了解更多。因此，总是使用前三名狼的搜索结果。所有其他搜索代理，包括 Omega 狼，都被指示重新排列，以匹配顶级代理的猎物位置。在狩猎中，灰狼算法的公式如下所示：

$$\begin{cases} \vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{cases} \quad (3-12)$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \\ \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \end{cases} \quad (3-13)$$

$$\vec{X}(i+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (3-14)$$

搜索智能体根据 2D 搜索空间中的 Alpha、Beta 和 Delta 改变其位置的方式如图 3-5 所示。圆内的最终位置是由搜索空间中 Alpha、Beta 和 Delta 狼的位置随机决定的。因此，Alpha、Beta 和 Delta 决定了猎物的位置，而其他狼则在其周围随机更新它们的位置。

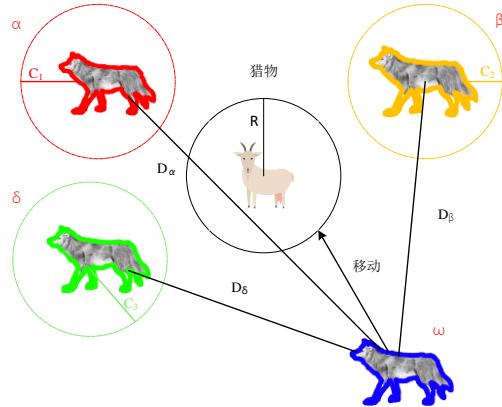


图 3-5 灰狼算法中的位置更新

(3) 攻击

灰狼在猎物停止移动时发起攻击，结束狩猎。为了模拟这种朝向猎物的分析方法，降低了 a 的值，这也减小了 \bar{A} 的波动范围。特别是， a 在迭代过程中从2降到0，随机减小在 $[-2a, 2a]$ 的范围内。当 \bar{A} 的随机值在 $[-1, 1]$ 之间时，狼未来的位置可以在其当前位置与猎物(目标解)位置之间的任何地方。当 $|A| < 1$ 时，狼群被迫向猎物发起冲锋的过程，如图 3-6 所示。

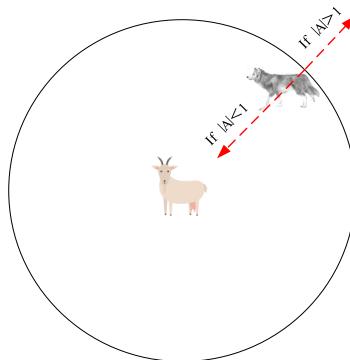


图 3-6 捕猎猎物和搜索猎物

灰狼优化算法中灰狼可以根据 Alpha、Beta 和 Delta 狼的位置来更新它们自身的位置。因此，它们可以利用建议的运算符朝向猎物发起攻击。然而，当使用这些运算符时，GWO 方法容易陷入局部最优解。

(4) 搜索

为了指导搜索过程，灰狼主要依赖于 Alpha、Beta 和 Delta 狼的位置信息。这些狼独立搜索猎物，然后重新聚集攻击猎物。为了在数学上模仿这种分散，可以使用随机值大于1或小于-1的 A 来驱使搜索代理远离猎物。这促进了探索，并使得全局应用 GWO 方法成为可能。如图 3-6 所示展示了当 $|A| > 1$ 时，灰狼如何从它们的猎物偏离以寻找更适合的猎物。

GWO 的另一个值得探究的方面是 \vec{C} 向量，它具有 $[0, 2]$ 的随机值。这个组成部分为猎物提供随机权重，以便在方程式 1 中随机强调 ($C > 1$) 或减弱 ($C < 1$) 猎物在影响距离方面的作用。因此，GWO 可能在优化中表现得更随机，偏好探索和利用，同时避免局部最优陷阱。与 A 不同， C 的下降不是线性的，这要求 C 始终在初期和后期迭代中提供随机值以优先考虑探索。在局部最优停滞的情况下，特别是在最后的迭代中，这个元素是有用的。

最后，为了开始搜索，通过 GWO 算法产生一个随机的灰狼种群。通过一系列迭代，Alpha、Beta 和 Delta 狼计算猎物的潜在位置。每个可能的响应都改变了它与猎物的距离。 a 的值从 2 降到 0，突出了探索和利用。当 $|A| > 1$ 和 $|A| < 1$ 时，代理解决方案经常从猎物分散。当满足结束标准时，GWO 算法完成整个流程。

3.3.2 改进灰狼优化算法

GWO 算法主要利用位置更新方程来找到最优解，而其存在以下缺陷：GWO 算法将狼分为四个等级，这些狼基于式(3-14)调整它们的位置以捕猎猎物，主要依靠 Alpha 狼、Beta 狼和 Delta 狼的信息来进行搜索。在解决多模态问题时，可能会导致算法停滞局部最优解处^[184]。在迭代过程中，因为 Delta 狼的解较差，在 Alpha 狼根据三只狼的信息同时调整其位置，可能引导 Alpha 狼走向错误的方向，同样，Beta 狼也可能获得错误的引导。因此，式(3-14)给予 Alpha 狼、Beta 狼和 Delta 狼同样的权重 $1/3$ ，这是不科学的^[185]。

混合方法可以有效地结合算法的优势，同时摒弃算法劣势。大量研究表明，与单一算法相比，混合算法在解决各种优化问题时表现更为出色^{[186][187]}。这种理念已广泛应用于传统的进化算法中，但在群体智能算法中，尤其是 GWO 算法领域，其应用相对较少。为了填补这一研究空白并扩展算法的应用范围，本研究首先加入非线性控制机制来优化收敛因子，然后根据狼在层次结构中的不同社会地位，对式(3-14)的位置更新方程进行了改进，以提高算法的探索能力，使其跳出局部最优，即利用非线性因子改进灰狼优化算法（Nonlinear Improved Grey Wolf Optimizer Algorithm，NI-GWO）。具体的改进方式如下：

(1) 适应度函数模型

多无人机协同航迹规划的综合评价函数为 NI-GWO 算法中适应度函数，根据 3.2 节协同航迹评价与变量之间的关系可将适应度值函数 $f(\vec{X})$ 表示为：

$$f(\vec{X}) = \min \sum_{i=1}^m (w_1 f_{o-i}(l) + w_2 f_{T-i}(p) + w_3 f_{m-i}(t) + w_4 f_{C-i}(X)) \quad (3-15)$$

(2) 非线性优化算法的收敛因子机制

随着智能算法的发展，灰狼算法被广泛应用于解决无人机的自主航路规划问题。但目前常用的灰狼算法的建模方式是收敛因子取值方式较为随机，并且缺少对无人机编队

整体适应度对灰狼位置更新的影响，无法达到算法在航路规划时的高效性和鲁棒性。于是对收敛因子的取值方式进行改进，文献[188]介绍了一种非线性控制收敛因子，由实验证明该 s 型函数变化的收敛因子可以提升算法的搜索能力，于是本章设计了非线性控制机制来优化收敛因子的取值，然后通过引入适应度值的权重来优化位置更新方式，并根据狼的适应度值对狼群算法中的位置更新方法进行改进。

由于其强大的搜索能力，群体智能优化算法被广泛用于不同类型的离散优化问题。由于全局搜索能力和局部搜索能力之间的不平衡，群体智能算法通常会导致收敛精度和优化性能的下降。在 GWO 算法中，参数 A 是算法寻找最优解的扰动因子，GWO 将根据变换值 \bar{A} 决定算法的开发和探索能力。从式(3-9)和式(3-11)中可知，参数 \bar{A} 的转换值将由收敛因子 \vec{a} 决定，随着迭代次数的增加， \vec{a} 的值将从 2 线性减少至 0。

对于 GWO 算法而言，迭代开始时较大的 \vec{a} 值可以增加狼群的搜索步长，扩大灰狼的收缩范围，避免过早的优化搜索过程成熟和过快的收敛，而迭代结束时较小的 \vec{a} 值将有助于狼群减少搜索步长，提高局部搜索能力，加快算法的收敛速度。然而，由于两种搜索类型并非完全按照线性切换，所以收敛因子的线性减少并不能实际反映出算法在多峰情况下的复杂搜索过程。因此，本文为了提升搜索能力，在 GWO 中引入了以下非线性控制机制：

$$\vec{a} = \vec{a}_{\text{initial}} \cdot \frac{1}{1 + e^{\frac{10i - 5I}{I}}} \quad (3-16)$$

其变化曲线图如图 3-7 所示。

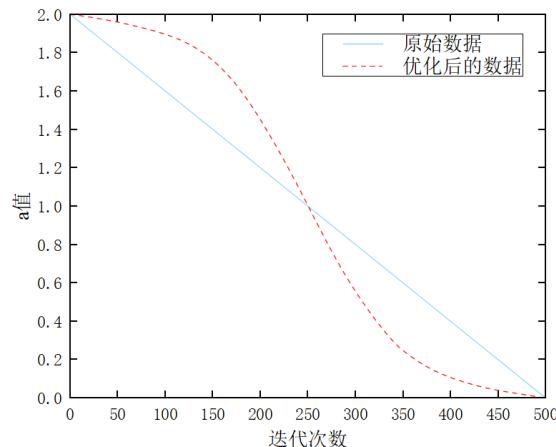


图 3-7 收敛因子变化曲线

其中， \vec{a}_{initial} 是收敛因子 \vec{a} 的初始值， i 表示当前迭代次数， I 表示最大迭代次数。如图 3-7 所示可以看出，基于 Sigmoid 函数校正的变化曲线在迭代开始时逐渐减少，使得收敛因子 \vec{a} 在更长时间内保持较高值，从而提高算法早期的全局搜索能力。由于 \vec{a} 的值

迅速下降，扰动因子 A 在迭代后期也可以保持较低的值，这增加了算法的局部搜索精度，并平衡了其全局和局部搜索的能力。

(3) 改进 GWO 的位置更新方程

在 GWO 中，前三名狼有不同的社会地位。为了更好地利用领头狼的信息，Beta 和 Delta 狼围绕 Alpha 狼进行搜索，Alpha 狼自行探索，根据狼的适应度值计算权重。为了增强算法的开发能力，给最佳狼分配更多的权重，如下所示：

$$\begin{cases} \vec{X}_\alpha = \vec{X}_\alpha + 2 \times \vec{a} \times \text{rand} \times (\vec{X}_{r_1} - \vec{X}_{r_2}) \\ \vec{X}_\beta = \vec{X}_\beta + 2 \times \vec{a} \times \text{rand} \times (\vec{X}_\alpha - \vec{X}_\beta) \\ \vec{X}_\delta = \vec{X}_\delta + 2 \times \vec{a} \times \text{rand} \times (\vec{X}_\alpha - \vec{X}_\delta) \end{cases} \quad (3-17)$$

其中， \vec{X}_α 、 \vec{X}_β 和 \vec{X}_δ 分别代表着 α 狼、 β 狼和 δ 狼的位置， \vec{a} 以线性的方式从 2 减少到 0， r_1 和 r_2 是范围在 1 到狼群数量之间的不同整数($r_1 \neq r_2$)， $\text{rand} \in [0,1]$ 是一个随机数。对于其余的狼，它们遵循位置更新方程。然而，将相同的权重赋予 \vec{X}_α 、 \vec{X}_β 和 \vec{X}_δ 是不公平的。权重取决于它们的适应度值。狼的适应度越好，它拥有的信息就越多，应该给予更多的权重。权重可以基于狼的适应度值来计算。通常，优化问题是为了找到最小化。权重的计算方式如下：

$$(w_\alpha, w_\beta, w_\delta) = (f(\vec{X}_\alpha) + f(\vec{X}_\beta) + f(\vec{X}_\delta)) \times \left(\frac{1}{f(\vec{X}_\alpha)}, \frac{1}{f(\vec{X}_\beta)}, \frac{1}{f(\vec{X}_\delta)} \right) \quad (3-18)$$

$$\vec{X}'(t+1) = \frac{w_\alpha \times \vec{X}_1 + w_\beta \times \vec{X}_2 + w_\delta \times \vec{X}_3}{w_\alpha + w_\beta + w_\delta} \quad (3-19)$$

其中， \vec{X}_α 、 \vec{X}_β 和 \vec{X}_δ 分别是 α 狼、 β 狼和 δ 狼的位置， $f(\vec{X}_\alpha)$ 、 $f(\vec{X}_\beta)$ 和 $f(\vec{X}_\delta)$ 是它们的适应度值， w_α 、 w_β 和 w_δ 是它们的权重，三个向量 \vec{X}_1 、 \vec{X}_2 和 \vec{X}_3 来自式(3-13)，式(3-18)和式(3-19)负责根据它们的适应度计算分配给前三名狼的权重。本次改进通过给最优的狼分配更多的权重(而不是传统 GWO 中的相同权重 1/3)来增强算法的开发利用。由于 α 狼对猎物位置的了解最多，因此设置最高的权重， β 狼排在第二位， δ 狼的权重最低。

利用所提出的 NI-GWO 算法求解第 3.2 节中的模型，具体过程如下：

步骤 1：设置每架无人机的起点和终点。本文实验将在二维场景中展开，起点和目标点坐标都为二维坐标。

步骤 2：建立了一个考虑了目标函数和两个约束条件的多无人机协同航路规划模型。

步骤 3：随机生成一定数量的灰狼。每只灰狼的位置代表一个可能的解，设置最大迭代次数、非线性收敛因子和其他相关参数。

步骤 4：根据多无人机协同航迹规划的综合评价函数，为每只灰狼计算适应度值，

确定群体中的 Alpha、Beta、Delta 狼。

步骤 5：根据式(3-15)中提到的 Sigmoid 函数调整收敛因子。收敛因子在迭代开始时较高，以增强全局搜索能力，随着迭代进行逐渐降低，以提高局部搜索精度。

步骤 6：根据式(3-16)和式(3-17)、(3-18)，根据 Alpha、Beta、Delta 狼的位置以及它们的适应度值，更新群体中其他灰狼的位置。权重的分配基于各狼的适应度值，以增强算法的开发利用。

步骤 7：确保所有灰狼的位置都在预定义的搜索空间内。式(3-15)用来评价该模型。

步骤 8：重复步骤 3 到步骤 7，直到达到最大迭代次数。

步骤 9：算法结束时，Alpha 狼的位置被认为是问题的最优解。无人机编队可以根据这个解进行引导。

3.4 仿真结果与分析

3.4.1 实验环境设定

为检验本文算法的性能，选取 GWO 和文献[189]所提出的 MP-GWO 算法作为对比算法。本实验中 GWO，MP-GWO 和 NI-GWO 三种算法均采用相同软件和硬件平台，操作系统为 Windows 10 系统，编程环境为 MATLAB R2020a。地图尺寸为 $20 \times 20 \text{ km}$ ，每张图中不同颜色的线段分别对应于相同环境约束和算法下不同初始位置和目标位置的无人机的航路规划结果。其中航路规划模型仿真参数如表 3-1 所示。

表 3-1 航路规划模型仿真参数表

| 参数名称 | 参数大小 |
|---------|-----------------|
| 狼群数量 | 80 |
| 最大探索步数 | 145 |
| 速度约束 | (0.3Ma ~ 0.7Ma) |
| 偏角约束 | (-60° ~ 60°) |
| 倾角约束 | (-45° ~ 45°) |
| 位置 x 约束 | (0 ~ 20km) |
| 位置 y 约束 | (0 ~ 20km) |
| 安全距离 | 500m |

3.4.2 仿真结果与分析

实验一：算法有效性分析

本次实验分别对两架、三架和四架无人机进行航路规划仿真。通过改变无人机的数量以验证所改进的算法在处理不同数量无人机时的有效性，并展现算法在这三种无人机数量场景中的适用性。

(1) 两架无人机航路规划

本次实验设定了2架无人机和2个目标点，经实验验证迭代次数设置为145次可以完成航路规划任务，每架无人机从设置好的初始位置通过GWO、MP-GWO和NI-GWO算法进行航路规划以到达相应的目标点，无人机位置信息和目标信息如表3-2所示，同时本次实验还在地图上设置了两个雷达威胁区和七个火炮威胁区如表3-3所示。

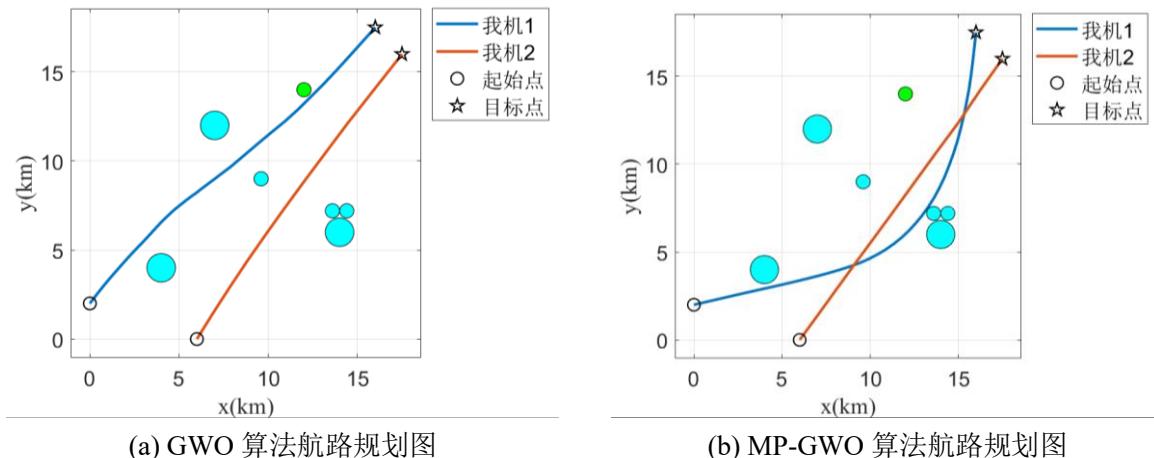
表 3-2 初始位置和目标位置设置表

| 参数名称 | 我机1 | 我机2 |
|------|-----------------|-----------------|
| 初始位置 | (0 km,2 km) | (6 km,0 km) |
| 目标位置 | (16 km,17.5 km) | (17.5 km,16 km) |

表 3-3 威胁区设置表

| 威胁区中心 | 雷达威胁区1 | 雷达威胁区2 | 火炮威胁区1 | 火炮威胁区2 | 火炮威胁区3 | 火炮威胁区4 | 火炮威胁区5 | 火炮威胁区6 |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|
| 横坐标(km) | 4 | 12 | 4 | 7 | 9.6 | 14 | 14.4 | 13.6 |
| 纵坐标(km) | 4 | 14 | 4 | 12 | 9 | 6 | 7.2 | 7.2 |
| 半径(km) | 0.4 | 0.4 | 0.8 | 0.8 | 0.4 | 0.8 | 0.4 | 0.4 |

分别利用三种算法对2架无人机在起始点到目标点进行航路规划，三种算法航路规划结果对比如图3-8所示。



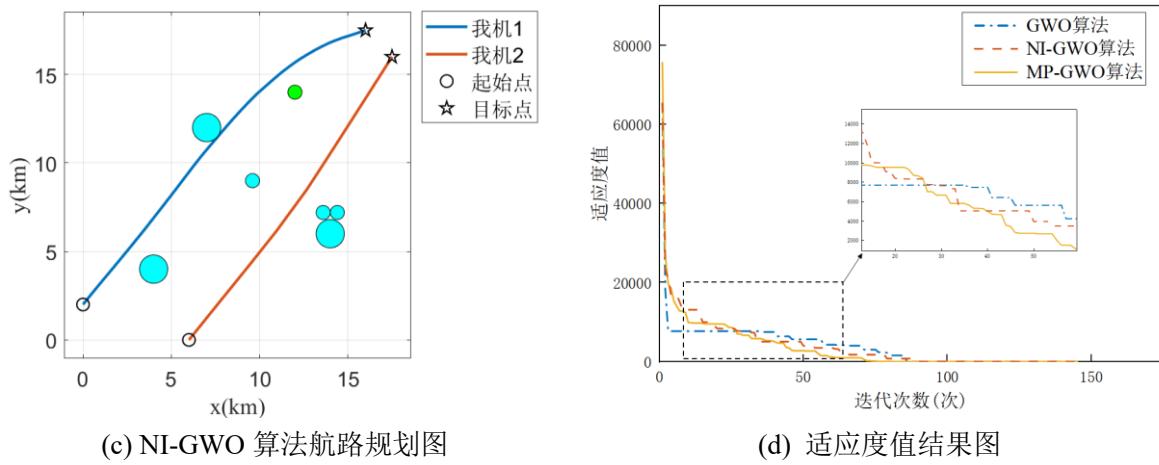


图 3-8 两架无人机在不同算法下的仿真结果图

如图 3-8 所示，三种算法都顺利完成了对 2 架无人机的路径规划。在我机 2 的航路规划中，三种算法的区别不大，而在我机 1 的航路规划中，三种算法在航行距离上产生了较大的差异。相较于 MP-GWO 算法，本文所提的 NI-GWO 算法路径规划的航行距离更短，且与 GWO 算法路径规划的结果相似。为了进一步分析无人机路径规划效果，重複 500 次仿真实验，对实验过程中的数据进行统计如图 3-9 所示。

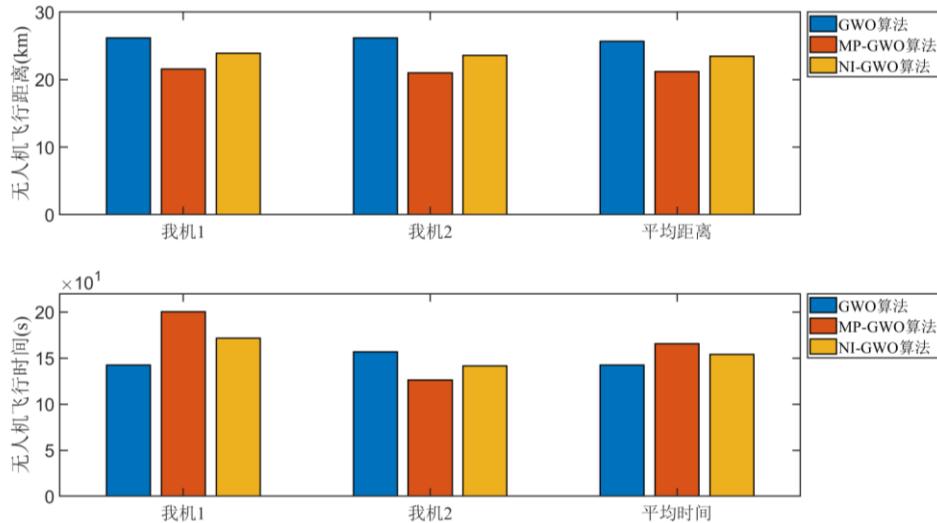


图 3-9 两架无人机的航路规划飞行数据

如图 3-9 所示，三种算法在规划两架无人机飞行过程中航行距离和航行时间相差不大，且三种算法的适应度都达到了 0.14。这表明三种算法在无人机编队路径规划过程中在无人机数量较少时都具有良好的性能。

(2) 三架无人机航路规划

在验证算法对 2 架无人机飞行路径规划的有效性后，本小节设置无人机飞行路径规划中无人机数量为 3 架。在本次仿真中，设置算法迭代次数与之前相同（145 次），飞行

地图中的威胁区信息参数不变。无人机位置信息和目标信息如表 3-4 所示。

表 3-4 初始位置和目标位置设置表

| 参数名称 | 我机 1 | 我机 2 | 我机 3 |
|------|-------------------|-----------------|-----------------|
| 初始位置 | (0 km,0 km) | (0 km,2 km) | (6 km,0 km) |
| 目标位置 | (17.5 km,17.5 km) | (16 km,17.5 km) | (17.5 km,16 km) |

如表 3-4 所示设置仿真场景，随后对三架无人机进行航路规划，航路规划如图 3-10 所示。

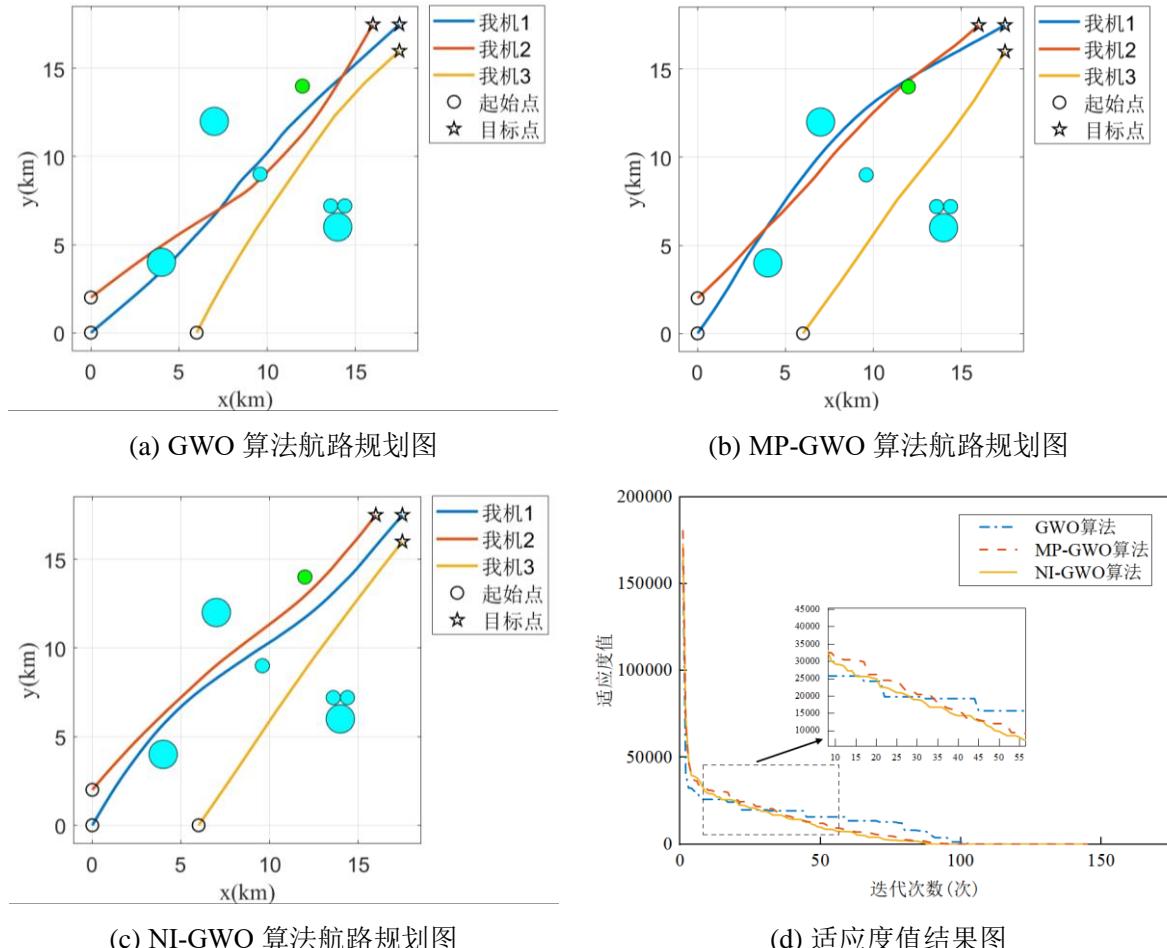


图 3-10 三架无人机在不同算法下的仿真结果图

如图 3-10 所示，三种算法对 3 架无人机航路规划都存在可行性。然而，本文所提的 NI-GWO 算法对威胁区的避障效果优于 GWO 算法和 MP-GWO 算法且规划路径距离最短。为了进一步分析无人机路径规划效果，重复 500 次仿真实验，对实验过程中的数据进行统计如图 3-11 所示。

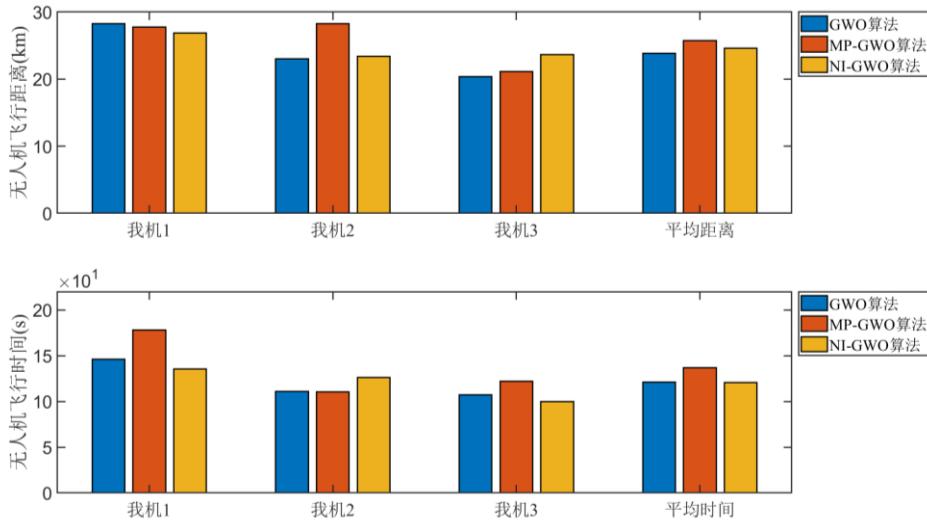


图 3-11 三架无人机的航路规划飞行数据

如图 3-11 所示，分析了三架无人机在三种航路规划算法中的飞行时间和距离，分别统计三种算法在该环境中运行的相关特征（最大航行距离、最小航行距离、最大航行时间、最小航行时间和适应度收敛值），如图 3-12 所示。

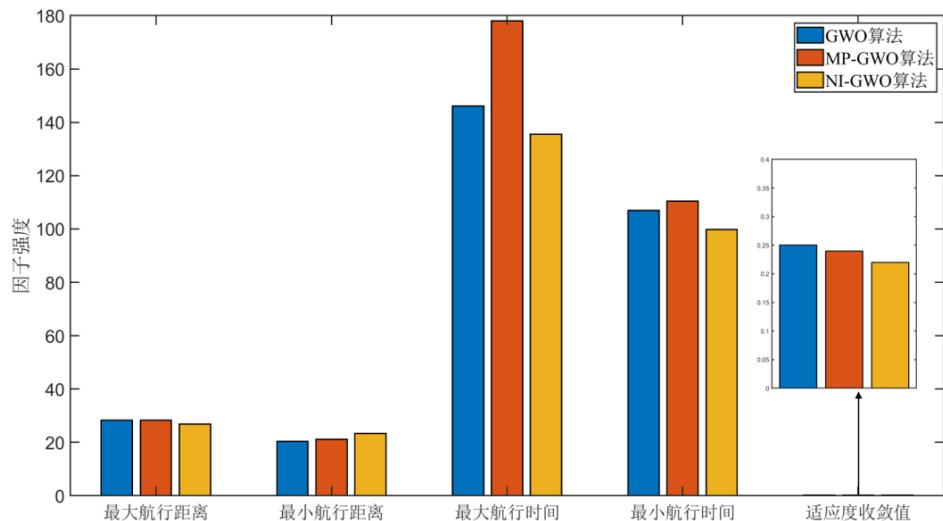


图 3-12 三架无人机的因子强度图

如图 3-12 所示，可以看出三种算法在对 3 架无人机进行航路规划过程中，NI-GWO 算法在最大航行距离、最小航行距离、最大航行时间、最小航行时间和适应度收敛值的表现上均优于两种传统算法。仿真结果表明，本文所提的 NI-GWO 算法在三架无人机路径规划效果上也优于传统的 GWO 算法和 MP-GWO 算法。同时，相比于对 2 架无人机航路规划过程中三种算法的相差无几，在 3 架无人机航路规划中本文所提的 NI-GWO 算法航路规划的效果逐渐提高，这表明本文所提的算法在面对更复杂问题中具有相对优势。

(3) 四架无人机航路规划

最后，对四架无人机进行路径规划，按照如表 3-1 所示设置算法迭代次数与威胁区信息参数后对两架无人机进行路径规划仿真。无人机位置信息和目标信息如表 3-5 所示。

表 3-5 初始位置和目标位置设置表

| 参数名称 | 我机 1 | 我机 2 | 我机 3 | 我机 4 |
|------|-----------------|---------------|---------------|-------------|
| 初始位置 | (0 km,0km) | (0km,2km) | (6km,0km) | (0km,6km) |
| 目标位置 | (17.5km,17.5km) | (16km,17.5km) | (17.5km,16km) | (16km,16km) |

不同算法下的航路规划仿真结果图如图 3-13 所示。

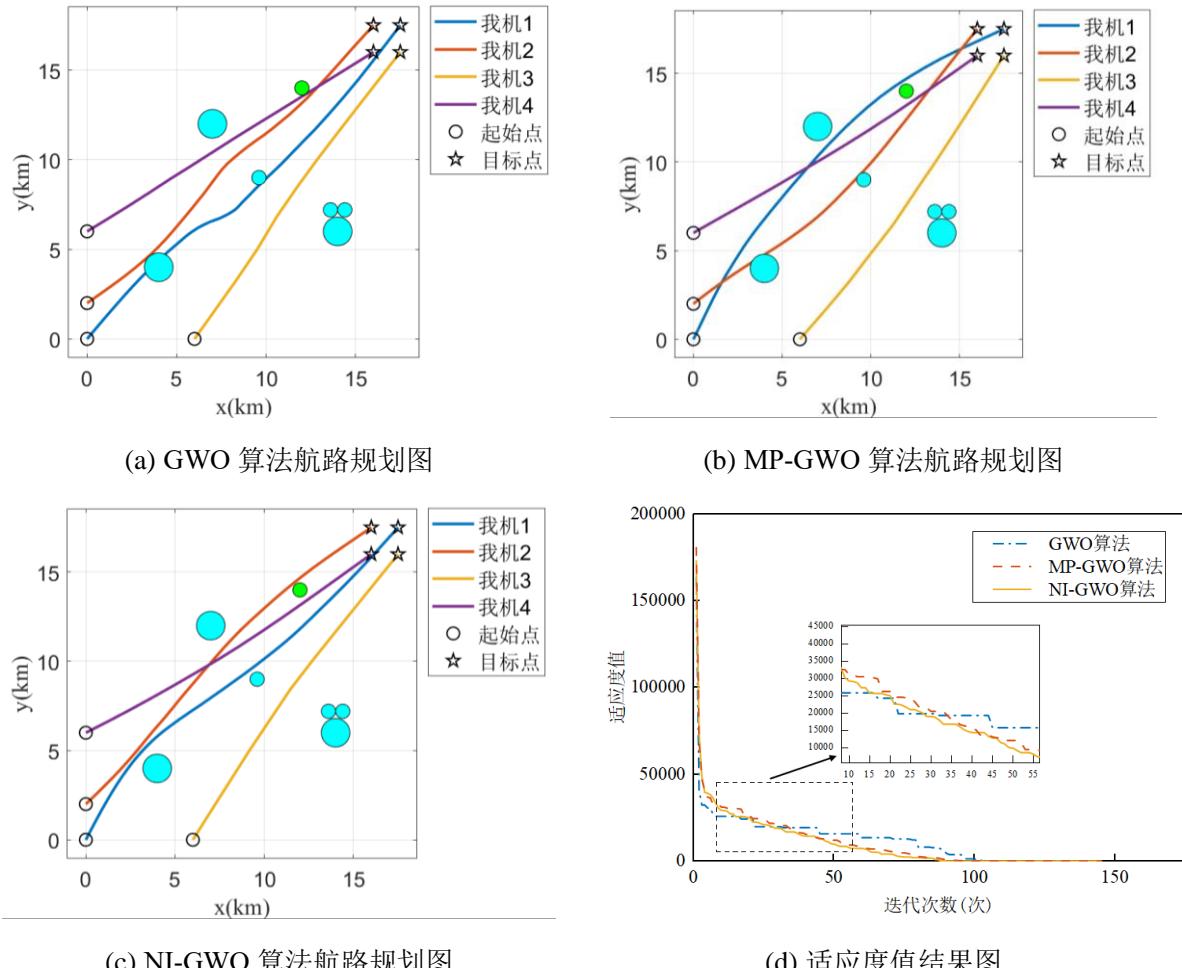


图 3-13 四架无人机在不同算法下的仿真结果图

如图 3-13 所示，GWO 算法路径规划效果最差，MP-GWO 算法路径规划效果次之，NI-GWO 算法路径规划效果最好。本文所提的 NI-GWO 算法对威胁区的避障效果最佳且规划路径距离最短。为了进一步分析无人机路径规划效果，重复 500 次仿真实验，对实验过程中的数据进行统计如图 3-14 所示。

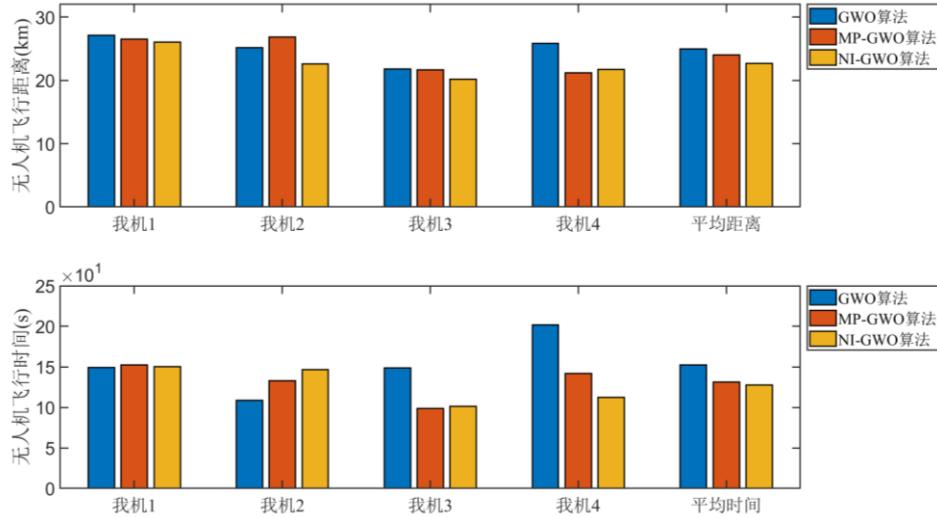


图 3-14 四架无人机的航路规划飞行数据

如图 3-14 所示，分析了四架无人机在三种航路规划算法中的飞行时间和距离，分别统计三种算法在该环境中运行的相关特征（最大航行距离、最小航行距离、最大航行时间、最小航行时间和适应度收敛值）如图 3-15 所示。

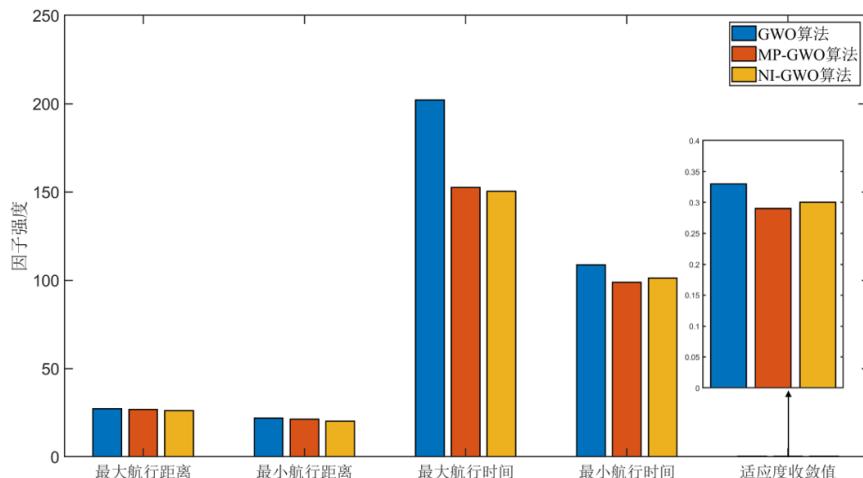


图 3-15 四架无人机背景下的因子强度图

如图 3-15 所示，可以看出本文所提的 NI-GWO 算法对 4 架无人机航路规划过程中在航行距离和航行时间上短于传统的 GWO 算法和 MP-GWO 算法，表明该算法在多机航路规划过程中可以尽可能的减少航行距离和航行时间。

综上分析，本文所提的 NI-GWO 算法在 2 架无人机航路规划中与传统算法航路规划能力相似，然而其在多机航路规划中表现出了相比传统算法更优的路径规划能力。这表明，本文所提的 NI-GWO 算法在多机航路规划上的效果上优于传统的 GWO 算法和 MP-GWO 算法，可以作为多机航路规划引导提供可靠的路径规划方法。即通过上述实验表明本文所提的 NI-GWO 算法在面对更为复杂的航路规划任务时有更好的航路规划

表现。

实验二：算法鲁棒性分析

为了检验算法的鲁棒性，本文设置了两组分析对照实验（算法迭代次数更改和路径规划场景更改）。(1) 算法迭代次数更改：为了验证模型在迭代次数降低时的表现，本实验分别设置了 100 次和 60 次迭代次数，以分析三种算法在迭代次数降低时的表现。(2) 路径规划场景更改：本实验分别增加威胁区和减少威胁区，以分析三种算法对不同威胁区场景的表现。

(1) 算法迭代次数更改

本次实验为了验证在迭代次数减少时 GWO 算法、MP-GWO 算法和 NI-GWO 算法的路径规划能力，首先将迭代次数降低至 100 次，障碍物设置与实验一相同，采用四架无人机进行仿真并记录该编队协同路径规划实验的仿真结果，具体地图场景设置如表 3-5 和表 3-3 所示。

设置迭代次数为 100 次进行仿真，对应无人机路径规划仿真图如图 3-16 所示。

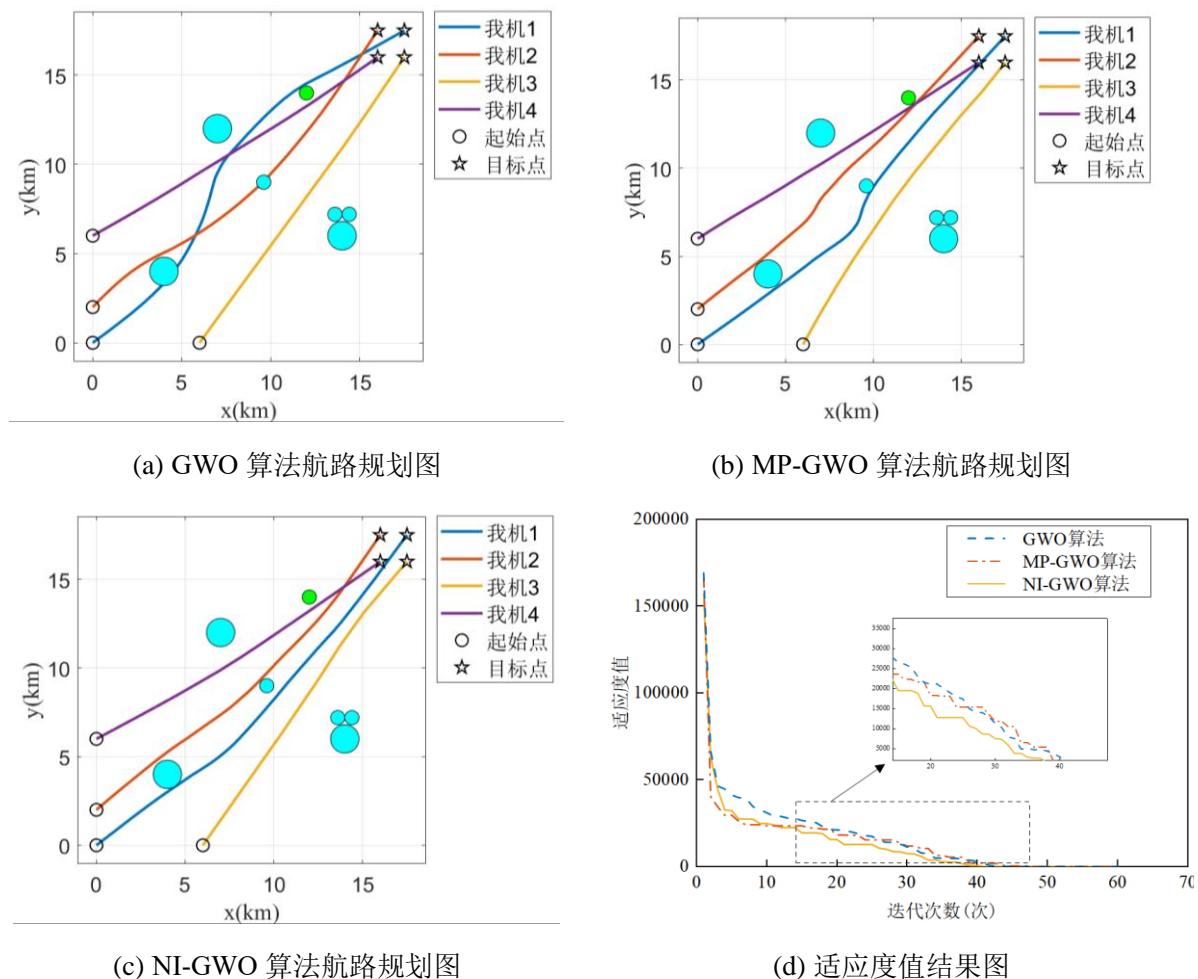


图 3-16 迭代次数为 100 次的仿真结果图

如图 3-16 所示，在迭代次数降低至 100 次时，GWO 算法路径规划效果迅速降低

(多次穿越威胁区域), MP-GWO 算法路径规划效果次之, NI-GWO 算法路径规划效果大致稳定。由上可以看出, 相较于 GWO 算法而言, 本文所提的 NI-GWO 算法对迭代次数不敏感, 可以容许一定的训练次数不足。为了进一步分析无人机路径规划效果, 重复 500 次仿真实验, 对实验过程中的数据进行统计如图 3-17 所示。

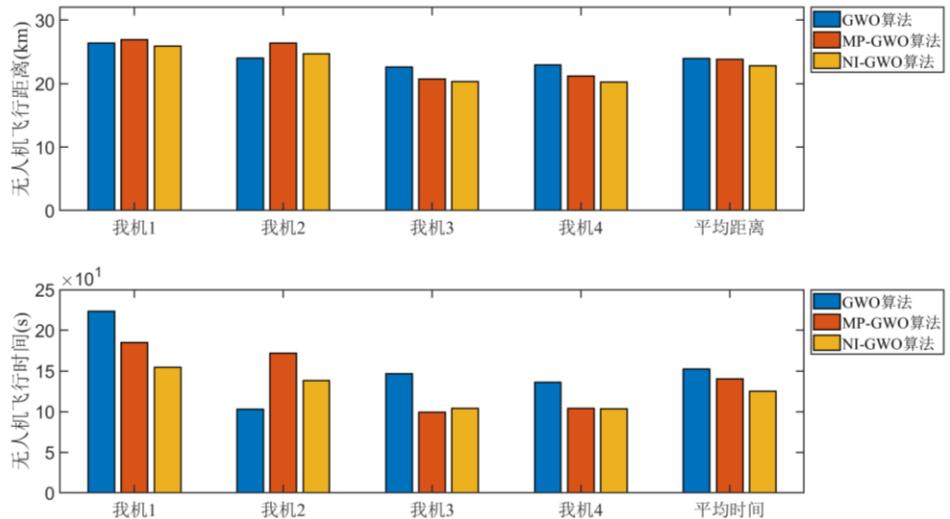


图 3-17 迭代次数为 100 次的航路规划飞行数据

如图 3-17 所示, 分析了迭代次数为 100 次时各无人机在三种航路规划算法中的飞行时间和距离, 分别统计三种算法在该环境中运行的相关特征(最大航行距离、最小航行距离、最大航行时间、最小航行时间和适应度收敛值), 如图 3-18 所示。

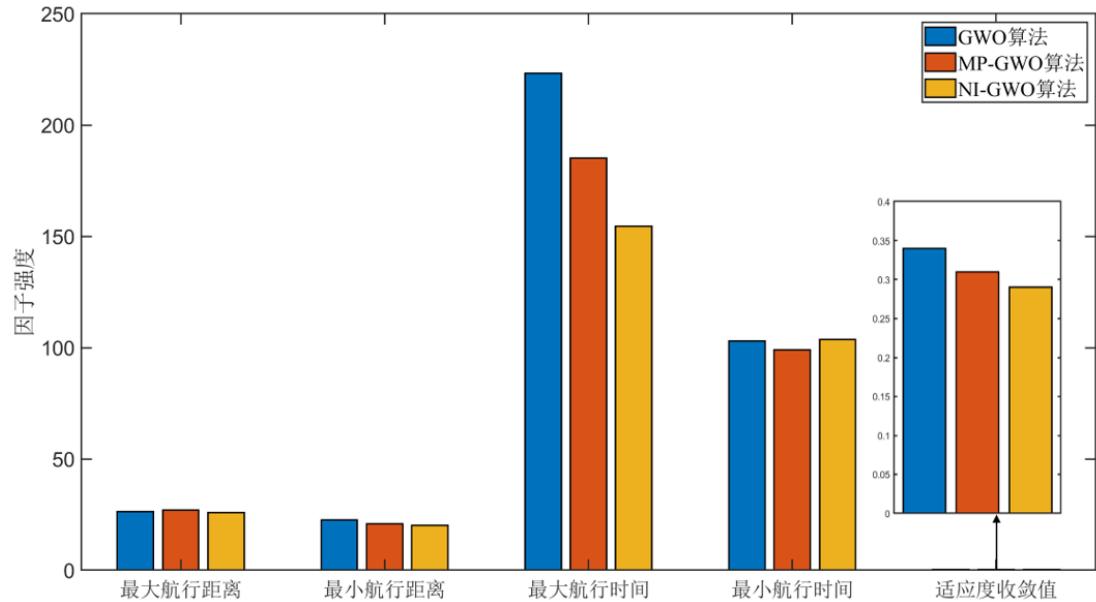


图 3-18 迭代次数为 100 次的因子强度图

从仿真统计数据可以看出, 迭代次数的降低导致了 GWO 和 MP-GWO 算法的性能迅速降低(包括迅速增长的航行距离和航行时间), 相比之下, 本文所提的 NI-GWO 算

法在遇到收敛次数降低时展现出了较优的效果。

随后将迭代次数减少至 60 次研究无人机编队的飞行性能。其仿真结果如图 3-19 所示。

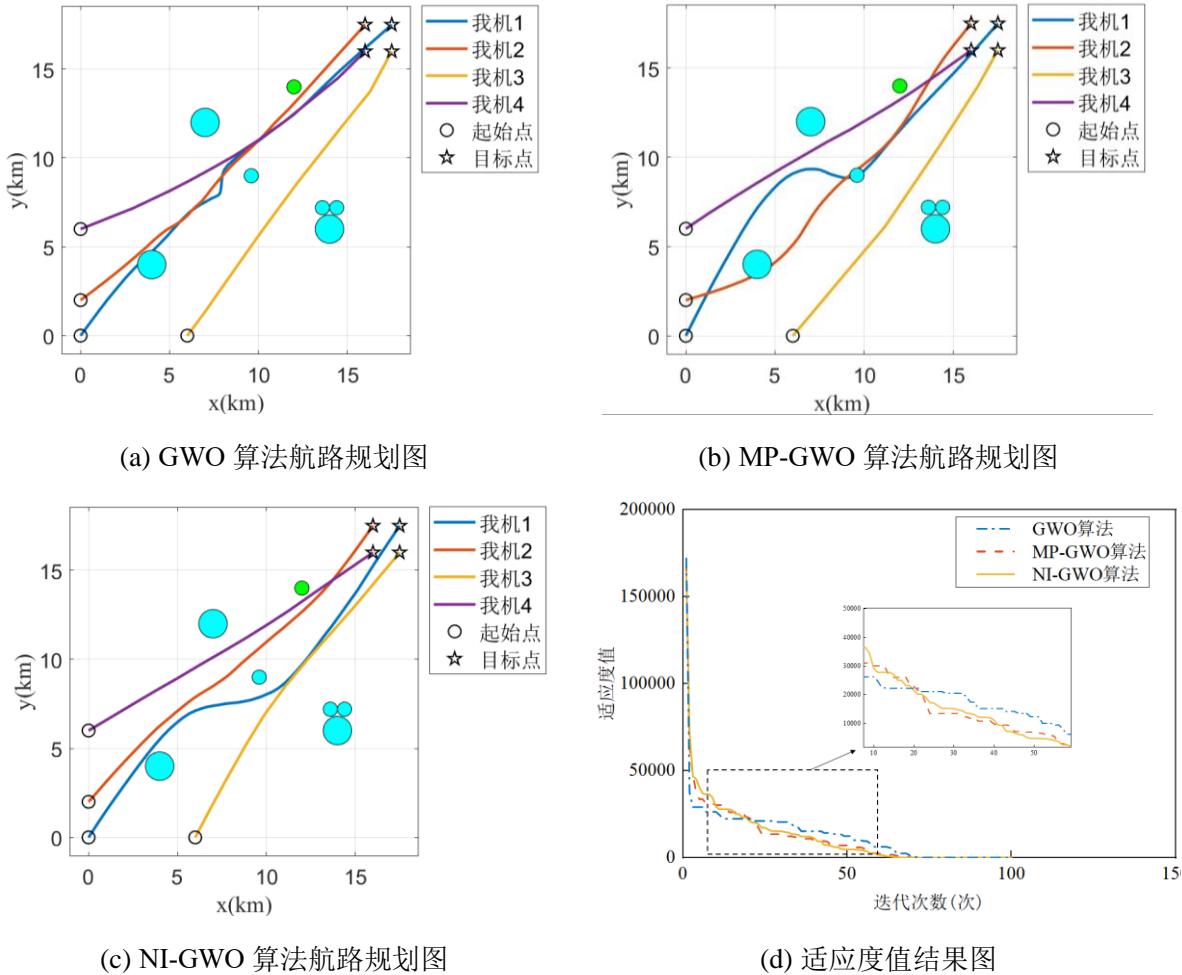


图 3-19 迭代次数为 100 次的仿真结果图

如图 3-19 所示，在迭代次数降低至 60 次时，GWO 和 MP-GWO 算法路径规划效果迅速降低（多次穿越威胁区），而 NI-GWO 算法路径规划效果大致稳定。以上分析可以看出，相较于 GWO 和 MP-GWO 算法而言，本文所提的 NI-GWO 算法对迭代次数不敏感，可以容许一定的训练次数不足。为了进一步分析无人机路径规划效果，重复 500 次仿真实验，对实验过程中的数据进行统计如图 3-20 所示。

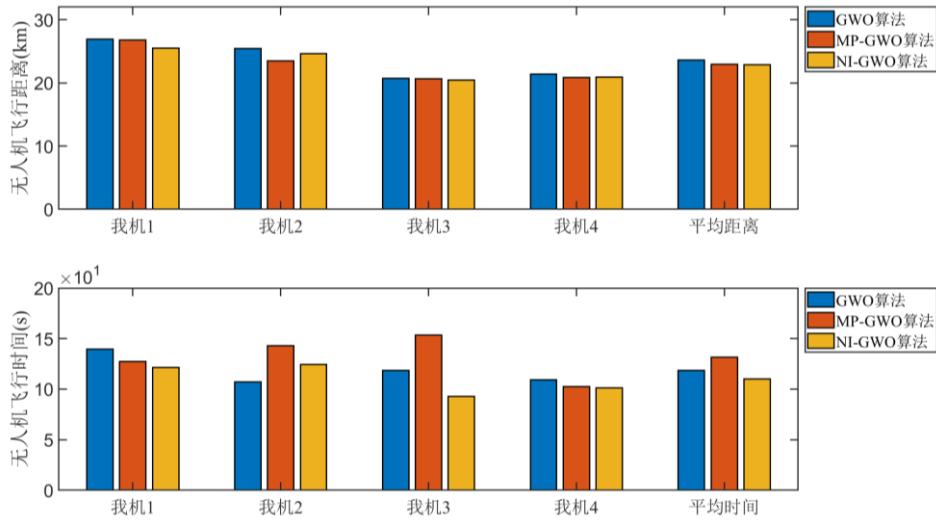


图 3-20 迭代次数为 60 次的航路规划飞行数据

如图 3-20 所示，从 60 次迭代的结果来看，三种算法的路径规划结果均有降低，可见迭代次数的降低会极大的影响无人机的路径规划能力。分别统计三种算法在该环境中运行的相关特征（最大航行距离、最小航行距离、最大航行时间、最小航行时间和适应度收敛值），如图 3-21 所示。

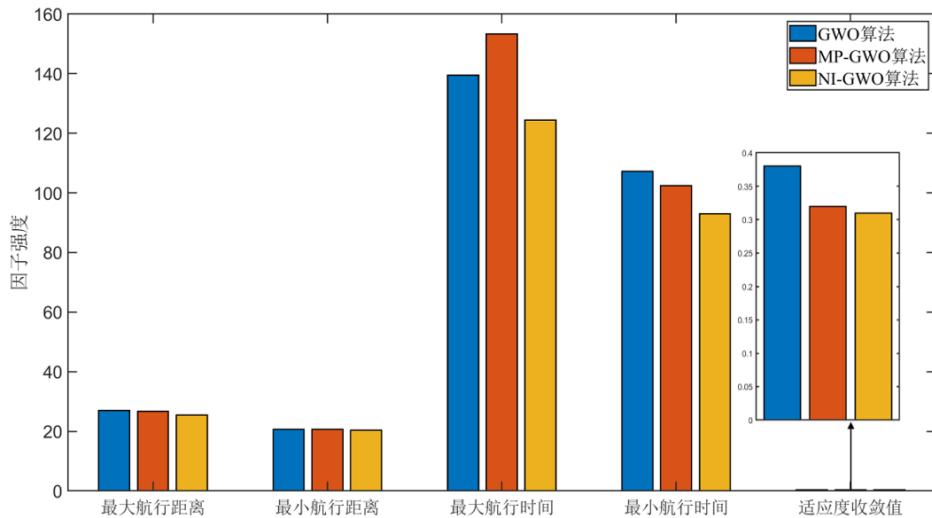


图 3-21 迭代次数为 60 次的因子强度图

从仿真统计数据可以看出，迭代次数的降低为 60 次导致了 GWO 和 MP-GWO 算法的航路规划能力迅速降低（包括迅速增长的航行距离和航行时间），相比之下，本文所提的 NI-GWO 算法在遇到收敛次数降低时展现出了较优的效果。由此可见 NI-GWO 算法在大幅度降低迭代次数的情况下并未受到较大的波动，具有较强的鲁棒性。

(2) 航路规划场景更改实验

本实验对基础的航路规划场景中的威胁区个数进行增减，以验证该算法在不同场景

下的鲁棒性。本次实验为了验证在航路规划场景改变时 GWO 算法、MP-GWO 算法和 NI-GWO 算法的航路规划能力，首先威胁区在实验一基础上进行改变，迭代次数保持不变，采用四架无人机进行仿真并记录该编队协同路径规划实验的仿真结果进行分析。

本实验对基础的航路规划场景中的威胁区个数进行增减，以验证该算法在不同场景下的鲁棒性。

首先，基于如表 3-3 所示的威胁区，本实验将在减少部分威胁区的情况下进行仿真，以验证不同算法下的鲁棒性，其中更新后的威胁区信息如表 3-6 所示。

表 3-6 威胁区设置表

| 威胁区中心 | 雷达威胁区 | 火炮威胁区 | 火炮威胁区 | 火炮威胁区 | 火炮威胁区 | 火炮威胁区 |
|---------|-------|-------|-------|-------|-------|-------|
| | 1 | 1 | 2 | 3 | 4 | 5 |
| 横坐标(km) | 4 | 7 | 9.6 | 14 | 14.4 | 13.6 |
| 纵坐标(km) | 4 | 12 | 9 | 6 | 7.2 | 7.2 |
| 半径(km) | 0.4 | 0.8 | 0.4 | 0.8 | 0.4 | 0.4 |

不同算法下的航路规划仿真结果如图 3-22 所示。

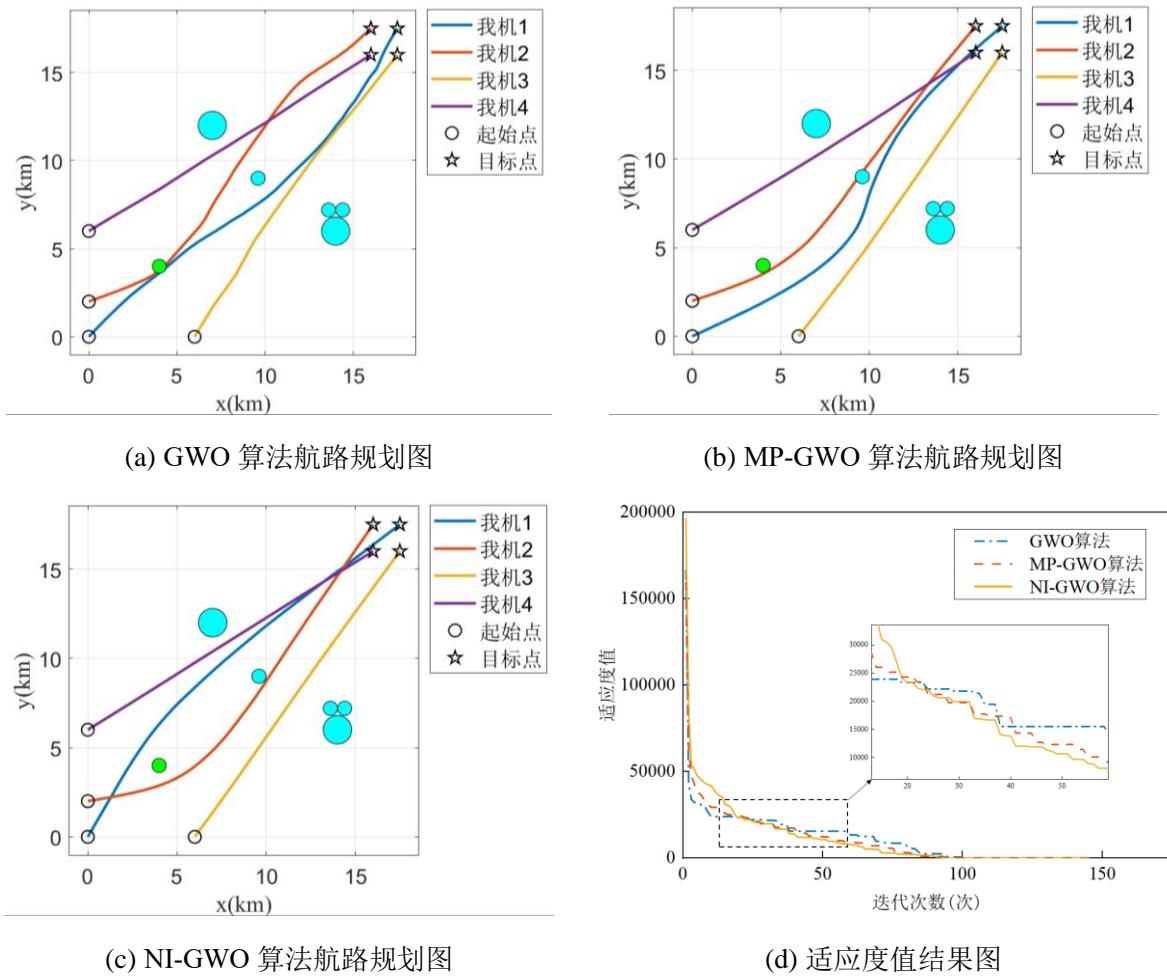


图 3-22 减少威胁区数量后的仿真结果图

如图 3-22 所示，在威胁区域发生变化后，GWO 和 MP-GWO 算法路径规划能力有所降低（多次穿越威胁区），而 NI-GWO 算法路径规划大致稳定。由上可以看出，相较于 GWO 和 MP-GWO 算法而言，本文所提的 NI-GWO 算法对威胁区场景的变化不敏感，可以适应复杂多变战场环境下的航路规划。为了进一步分析无人机路径规划效果，重复 500 次仿真实验，对实验过程中的数据进行统计如图 3-23 所示。

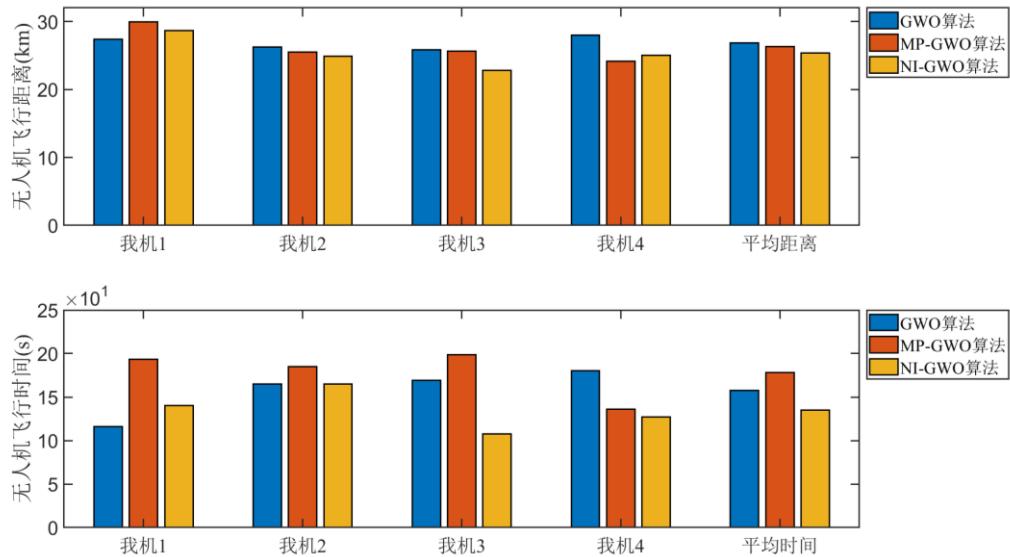


图 3-23 减少威胁区后航路规划飞行数据

如图 3-23 所示，减少威胁区数量后，三种算法的路径规划结果有所提升，但 GWO 算法和 MP-GWO 算法在路径规划中会存在穿越威胁区的情况，相对而言，NI-GWO 算法的稳定性较高。分别统计三种算法在新环境中运行的相关特征（最大航行距离、最小航行距离、最大航行时间、最小航行时间和适应度收敛值），如图 3-24 所示。

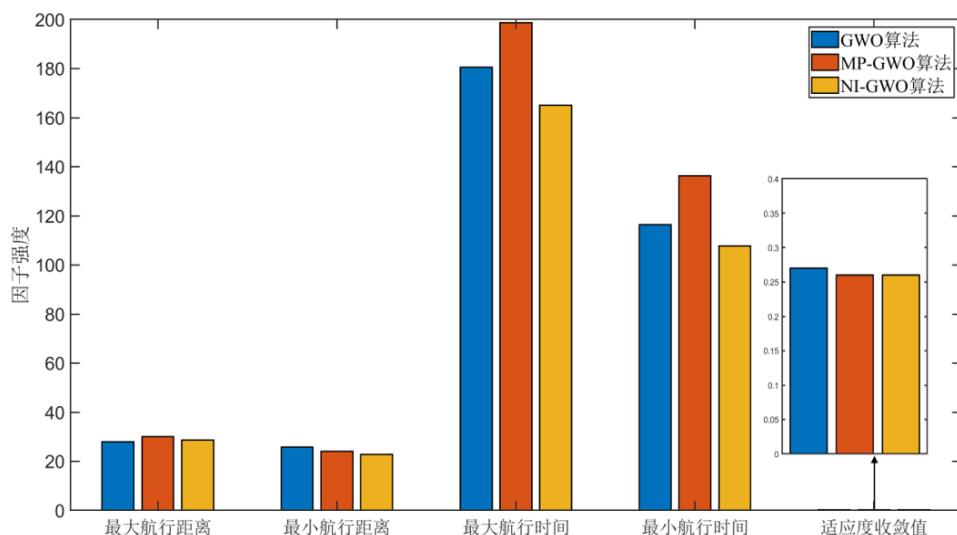


图 3-24 减少威胁区后因子强度图

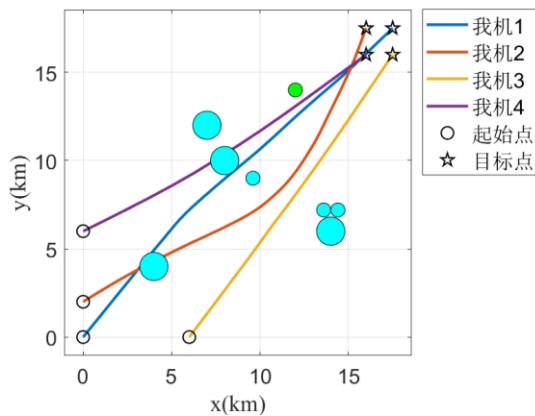
从仿真统计数据可以看出，在减少威胁区后 MP-GWO 算法的航行时间为三种算法中最长的；GWO 算法的最大航行距离与 NI-GWO 算法接近，但是最小航行距离最大；NI-GWO 算法的最小航行距离最小，且航行时间和航行速度均有着最优的效果。相比之下，本文所提的 NI-GWO 算法在所设置的新场景中的航路规划能力更优。

随后验证增加威胁区数量时三种算法对不同威胁区场景的鲁棒性。在如表 3-3 所示的威胁区基础上增加了 3 个火炮威胁区，增加后的威胁区信息如表 3-7 所示。

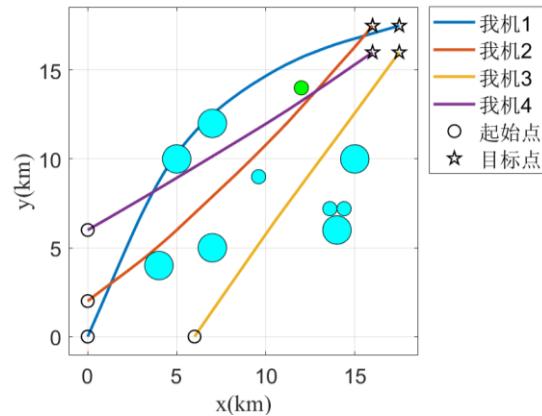
表 3-7 增加后的威胁区设置表

| 威胁 区中 心 | 雷达 威胁 区 1 | 雷达 威胁 区 2 | 火炮 威胁 区 1 | 火炮 威胁 区 2 | 火炮 威胁 区 3 | 火炮 威胁 区 4 | 火炮 威胁 区 5 | 火炮 威胁 区 6 | 火炮 威胁 区 7 | 火炮 威胁 区 8 | 火炮 威胁 区 9 |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 横坐 | | | | | | | | | | | |
| 标 (km) | 4 | 12 | 4 | 7 | 9.6 | 14 | 14.4 | 13.6 | 5 | 7 | 15 |
| 纵坐 | | | | | | | | | | | |
| 标 (km) | 4 | 14 | 4 | 12 | 9 | 6 | 7.2 | 7.2 | 10 | 5 | 10 |
| 半径 (km) | 0.4 | 0.4 | 0.8 | 0.8 | 0.4 | 0.8 | 0.4 | 0.4 | 0.8 | 0.8 | 0.8 |

不同算法下的路径规划仿真结果如图 3-25 所示。



(a) GWO 算法航路规划图



(b) MP-GWO 算法航路规划图

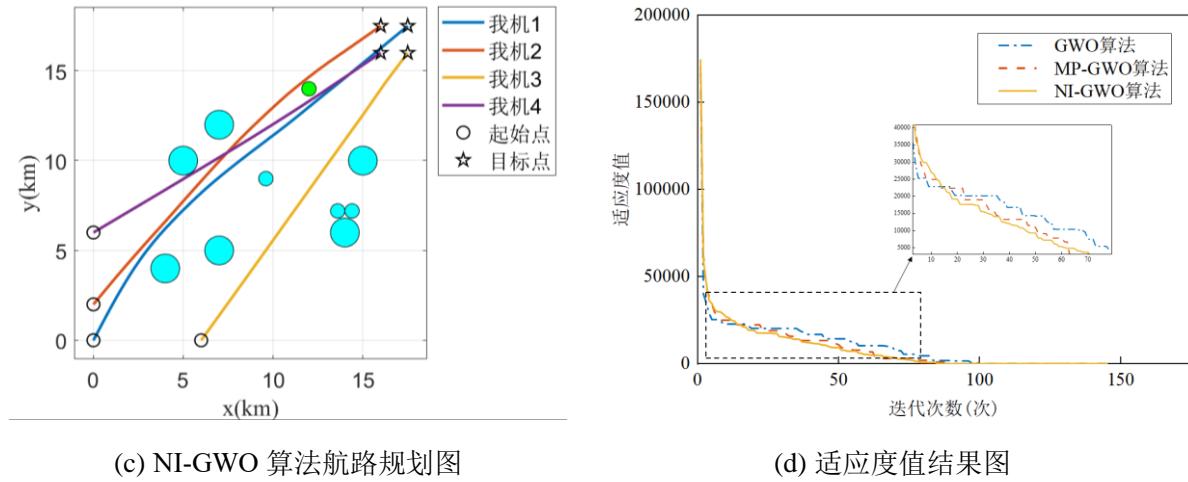


图 3-25 增加威胁区数量的仿真结果图

在图 3-25 中可以看出，新增威胁区域后，GWO 和 MP-GWO 算法出现多次穿越威胁区的情况，而 NI-GWO 算法没有出现穿越威胁区的情况。由上可以看出，相较于 GWO 和 MP-GWO 算法而言，本文所提的 NI-GWO 算法对更为复杂的作战场景有着相对优秀的路径规划能力。为了进一步分析无人机路径规划效果，重复 500 次仿真实验，对实验过程中的数据进行统计如图 3-26 所示。

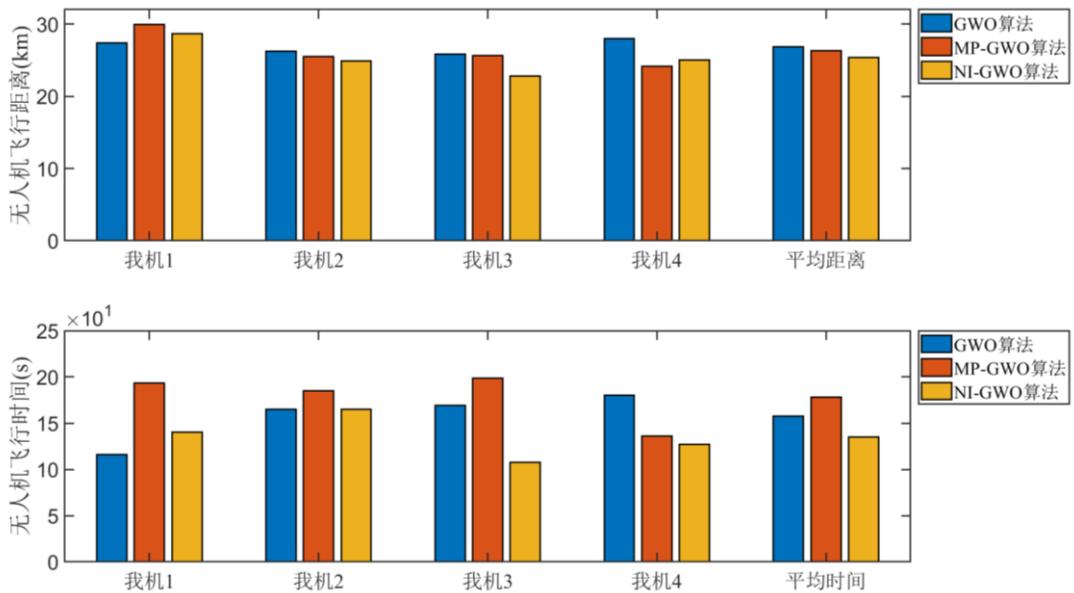


图 3-26 增加威胁区后航路规划飞行数据

如图 3-26 所示，在威胁区数量增加后，三种算法的路径规划结果均有所降低（航行距离增大、航行时间延长）。GWO 算法和 MP-GWO 算法在路径规划中会多次出现穿越威胁区的情况，而 NI-GWO 算法避免了穿越威胁区的情况，保证了路径规划的可靠性。最后，分别统计三种算法在新环境中运行的相关特征（最大航行距离、最小航行距离、最大航行时间、最小航行时间和适应度收敛值），其结果如图 3-27 所示。

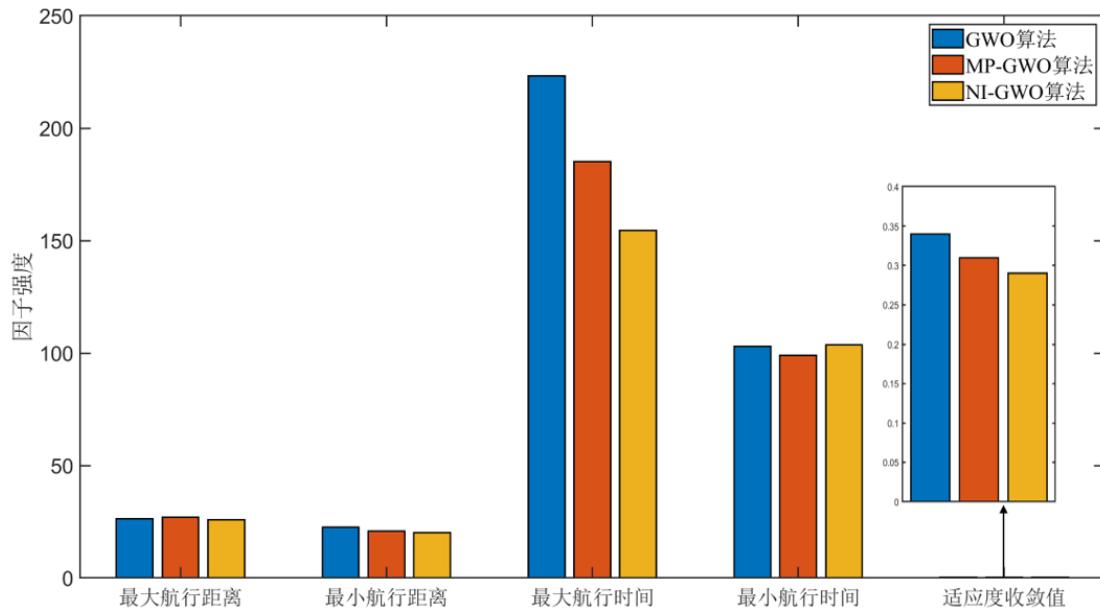


图 3-27 增加威胁区后因子强度图

如图 3-27 所示，在增加威胁区后 MP-GWO 算法的航行距离为三种算法中最长的；GWO 算法的航行时间时三种算法中最长的；相较于两种传统算法而言，尽管本文所提的 NI-GWO 算法并未做到所有性能均为最优，但其在多个性能评价指标中进行了均衡且在新场景下的路径规划中未穿越威胁区。相比与两种传统算法，本文所提的 NI-GWO 算法在全新的复杂场景下的路规划能力更优。

通过本文设置的三种仿真实验（不同架无人机航路规划分析、迭代次数鲁棒性分析以及新增威胁区域后航路规划的鲁棒性分析）分析结果表明，相较于两种传统算法 GWO 算法和 MP-GWO 算法，本章所提的 NI-GWO 算法在解决多机航路规划问题中拥有着卓越的表现，同时算法在迭代次数和作战场景上有较强鲁棒性。

3.5 本章小结

本章针对无人机编队航路规划问题，提出一种基于 NI-GWO 算法的多无人机航路规划方法。首先，根据多无人机协同路径规划问题，建立了多机协同航路规划模型；其次，对 GWO 算法的收敛因子和位置更新方式进行了改进，提出了 NI-GWO 算法模型；随后，分别设置两架、三架和四架无人机仿真场景，以验证 NI-GWO 算法对多无人机航路规划的有效性；最后，通过更改迭代次数和增减威胁区域以验证 NI-GWO 算法对多无人机航路规划的鲁棒性。

第4章 基于深度强化学习的无人机空战机动决策模型

在无人机航路规划引导进入作战区域后，各无人机需要完成相应的机动决策从而实现指定的战术目标。本章针对空战机动决策进行建模，从而为下一章中无人机编队协同策略模型提供机动决策元策略模型。首先，根据攻击区的解算模型，采用BP神经网络对导弹攻击区进行解算与改进；其次，针对SAC算法的收敛速度慢，可能会陷入局部最优的不足，设计了E-SAC算法，并根据算法模型和导弹攻击区终端约束选择了合适的状态空间和动作空间，建立了相应的奖励函数；最后，通过在不同场景下的仿真分析对算法有效性进行验证。

4.1 问题分析

进入作战区域后，各无人机需要完成对应的机动决策以实现作战任务。无人机机动决策应先解算出导弹的攻击区，随后引导无人机完成对应的机动决策，实现对目标敌机的追踪与打击任务。

本章从导弹攻击区解算和机动决策设计两方面对无人机编队机动决策进行研究。首先基于攻击区计算模型，设计了攻击区解算方法，通过黄金分割法解算出导弹攻击区数据完成BP神经网络对攻击区的拟合，再到对传统的BP神经网络进行改进以提升拟合的准确度，完成了对导弹攻击区解算的设计；其次根据解算出的导弹攻击区建立奖励函数和终端约束条件，并建立无人机空战机动决策的状态空间和动作空间，设计了基于E-SAC算法的无人机编队作战的机动决策模型；最后基于MaCA平台完成了仿真验证。本章具体研究内容如图4-1所示。

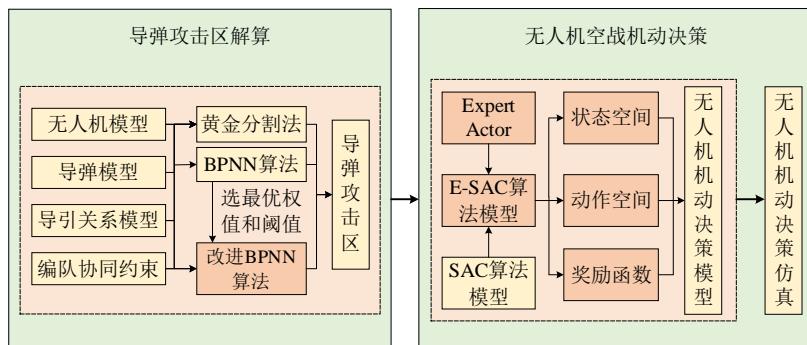


图 4-1 基于深度强化学习的无人机空战机动决策模型内容框图

4.2 导弹攻击区解算方法

4.2.1 导弹数学模型

根据2.1节的假设，导弹的三自由度质点模型如图4-2所示^[190]。

(1) 导弹动力学模型

导弹动力学模型描述了导弹机动特性的数学表达。根据 2.1 节的假设，在东北天惯性坐标系下建立了导弹的三自由度动力学方程组：

$$\begin{cases} \frac{dv_m}{dt} = \frac{F_m - F_a}{m_m} - g \sin \theta_m \\ \frac{d\theta_m}{dt} = \frac{g(n_{my} - \cos \theta_m)}{v_m} \\ \frac{d\varphi_m}{dt} = -\frac{n_{mz}g}{v_m \cos \theta_m} \end{cases} \quad (4-1)$$

其中， v_m 、 θ_m 、 φ_m 分别表示空空导弹的运动速度、弹道倾角、弹道偏角； F_m 和 F_a 分别表示空空导弹的发动机动力和飞行过程中所受的空气阻力； n_{my} 与 n_{mz} 分别表示空空导弹的侧向与法向过载。

(2) 导弹运动学模型

导弹运动学模型是以数学方式描述导弹运动规律的工具。在前面一系列假设的基础上，建立了导弹的三自由度运动学方程组：

$$\begin{cases} \frac{dx_m}{dt} = v_m \cos \theta_m \cos \varphi_m \\ \frac{dy_m}{dt} = v_m \sin \theta_m \\ \frac{dz_m}{dt} = -v_m \cos \theta_m \sin \varphi_m \end{cases} \quad (4-2)$$

其中， x_m 、 y_m 和 z_m 分别表示在 X 轴、Y 轴和 Z 轴正方向上的导弹位置坐标。

4.2.2 导弹运动与导引模型

我机发射导弹与敌机的弹目相对运动关系如图 4-2 所示。

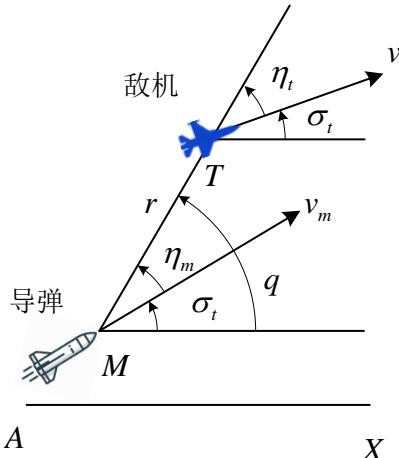


图 4-2 导弹与目标相对运动关系示意图

如图 4-2 所示，其中 M 表示导弹， T 表示目标敌机的位置， MT 代表目标线， AX

表示攻击平面内的水平参考线；导弹与目标之间的距离被表示为 r ，当导弹命中目标时， r 为零；导弹速度矢量和目标速度矢量分别由 v_m 和 v_t 表示，目标线方位角 q 是目标线与水平参考线之间的夹，导弹弹道角 σ_m 和目标航向角 σ_t 分别表示导弹速度矢量和目标速度矢量与水平参考线之间的夹角。导弹前置角 η_m 和目标前置角 η_t 则分别表示导弹速度矢量和目标速度矢量与目标线之间的夹角。

如上可得到导弹与目标之间的相对运动关系模型如式所示：

$$\begin{cases} \frac{dr}{dt} = v_t \cos \eta_t - v_m \cos \eta_m \\ r \frac{dq}{dt} = v_m \sin \eta_m - v_t \sin \eta_t \\ q = \sigma_m + \eta_m \\ q = \sigma_t + \eta_t \\ \eta_1 = 0 \end{cases} \quad (4-3)$$

其中， $\eta_1 = 0$ 表示导引关系控制方程由方程组。

当前导弹的主要制导技术包括追踪法、平行法和比例导引法。其中，比例导引法能够有效地利用导弹的机动性，被广泛用于多种类型的导弹制导系统中。该方法的核心原理如式(4-4)所示。

$$\eta_1 = \frac{d\sigma_m}{dt} - K \frac{dq}{dt} = 0 \quad (4-4)$$

其中， K 为比例系数。式(4-3)和式(4-4)结合构成以比例导引法的数学模型如式所示：

$$\begin{cases} \frac{dr}{dt} = v_t \cos \eta_t - v_m \cos \eta_m \\ r \frac{dq}{dt} = v_m \sin \eta_m - v_t \sin \eta_t \\ q = \sigma_m + \eta_m \\ q = \sigma_t + \eta_t \\ \frac{d\sigma_m}{dt} = K \frac{dq}{dt} \end{cases} \quad (4-5)$$

(1) 导弹攻击区解算数学模型

在复杂多变的空战环境中，需要预先获取导弹的最大攻击区、最小攻击区以及不可逃逸攻击区的数据信息。这能够给予无人机更多攻击位置的选择，在保证我方无人机安全的情况下，使得发射导弹命中目标的概率尽可能大。

(2) 空空导弹最大攻击区

导弹最大攻击区是指能够以一定概率命中目标的导弹最大发射区域。其解算过程中

的主要约束条件有以下四个方面：

1) 导弹最大攻击区条件下的雷达截获概率 P_{dr_max}

2) 导弹的能源工作时间 t_{energy}

3) 导弹在命中目标前的最小速度 v_{m_min}

4) 导弹在命中目标时导弹与目标的最小相对速度 v_{r_min}

综合以上四个影响因素可得，导弹最大攻击区的终端约束如式所示：

$$\begin{cases} P_{dr} \geq P_{dr_max} \\ t_e \leq t_{energy} \\ v_m \geq v_{m_min} \\ v_r \geq v_{r_min} \end{cases} \quad (4-6)$$

其中， P_{dr} 表示载机的雷达截获概率； t_m 表示空空导弹的飞行时间； v_m 表示在命中前，空空导弹的飞行速度； v_r 表示命中时的弹目相对速度。

(3) 空空导弹最小攻击区

导弹最小攻击区指的是在导弹能够以一定概率命中目标的条件下，无人机在保证自身安全的最小安全距离内发射导弹的区域。其解算过程中的主要约束条件包括以下三个方面：

1) 空空导弹的最大过载 n_{m_max}

2) 空空导弹引信解除保险时间 t_{fuze}

3) 无人机最小安全距离 d_{safe}

综合以上三个影响因素，空空导弹最小攻击区的约束条件如式所示：

$$\begin{cases} n_m \leq n_{m_min} \\ t_m \geq t_{fuze} \\ d_u \geq d_{safe} \end{cases} \quad (4-7)$$

其中， n_m 为空空导弹的机动过载； t_m 为空空导弹的飞行时间； d_u 为无人机与目标的距离。

(4) 空空导弹不可逃逸攻击区

空空导弹的不可逃逸攻击区是指无论目标做出任何机动，导弹都必定能够击毁目标的发射区域。除导弹雷达截获概率不同外，空空导弹不可逃逸攻击区与(2)节的约束条件相同，即如式所示：

$$\begin{cases} P_{dr} \geq P_{dr_ine} \\ t_e \leq t_{energy} \\ v_m \geq v_{m_min} \\ v_r \geq v_{r_min} \end{cases} \quad (4-8)$$

其中, P_{dr_ine} 为不可逃逸攻击区条件下的无人机雷达截获概率。

4.2.3 传统 BP 神经网络的导弹攻击区拟合方法

(1) 解算流程

黄金分割法是一种传统的导弹的攻击区拟合方法, 然而其计算速度缓慢, 尤其是在弹道仿真中需要解算高阶微分方程, 因此该方法难以适应空战场景中迅速变化的战况和对快速反应的需求。鉴于无人机和目标在空战中的高速移动和复杂多变的战场态势, 需要一种能够迅速响应的解算方法。

为此, 本节采用一种快速拟合攻击区的技术——反向传播神经网络(Backpropagation Neural Network, BNN), 即 BP 神经网络, 该算法具有强大的非线性映射能力和普遍适用性, 能够有效处理复杂的数据模式, 并且无需对内部结构和参数进行详细的建模。因此, 本节将黄金分割法计算出导弹的攻击区范围作为 BP 神经网络的训练样本, 使用合适配置的神经网络进行拟合, 以在实现攻击区域的精确预测, 同时满足无人机空战决策的实时性需求。

BP 神经网络是一种多层前馈神经网络, 它通过反向传播算法对网络中的权重和偏置进行调整, 以实现对输入数据的有效学习, 目前被广泛应用于空空导弹攻击区的解算和预测中, 其网络结构图如图 4-3 所示。

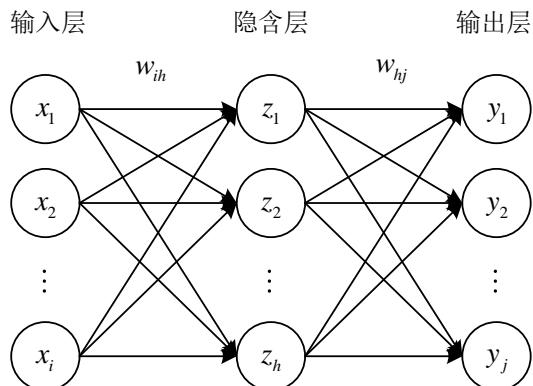


图 4-3 BP 神经网络结构图

BP 神经网络的工作原理基于两个关键步骤: 前向传播和反向传播。在前向传播中, 输入数据经过网络的多个层次, 每一层使用权重和偏置处理数据并应用激活函数, 以生成下一层的输入。当数据到达输出层时, 网络产生一个预测结果。接着, 在反向传播中, 这个预测结果与实际值比较, 计算误差。然后, 这个误差沿着网络向后传播, 途中根据误差的梯度调整每层的权重和偏置。通过这种方式, BP 神经网络在多次迭代中不断优化, 以减少输出结果和实际值之间的差异, 从而提高预测的准确性。其具体流程图如图 4-4 所示。

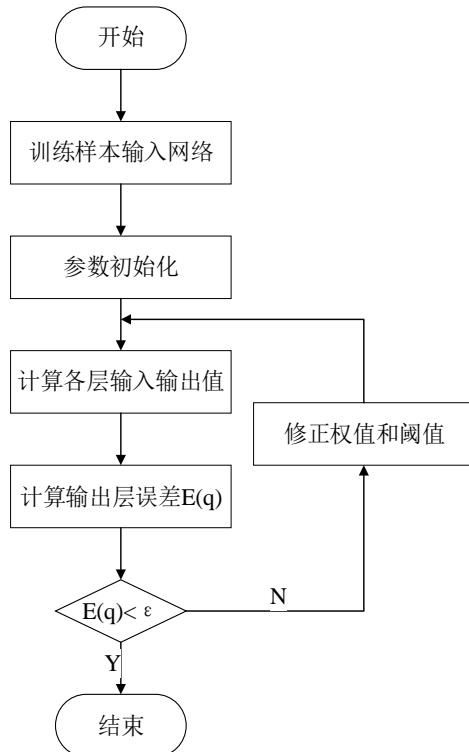


图 4-4 BP 神经网络算法流程图

为了优化 BP 神经网络的拟合准确度并提升其训练效率，本节将待训练的数据集根据目标线方位角的不同大小分为 18 个子集（设定方位角以每 20 度为一个区间划分数据集），通过对这些分组数据集分别使用 BP 神经网络进行训练，构建一个较为快速、准确的神经网络模型，满足实时性和精确度的需求。

在这个模型中，输入网络的训练集涵盖要素有：我机的飞行速度 v_u 、我机的飞行高度 h_u 、目标的飞行速度 v_t 和目标的飞行高度 h_t ，而输出则是攻击区边界的距离。

(2) 仿真结果与分析

本节首先设定了不同的初始条件并采用黄金分割法^[19]模拟出理论导弹攻击区数据，在仿真设置中，我机飞行角度和目标进入角度的变化范围为 $[0^\circ, 360^\circ]$ ，我机和目标的飞行高度的变化范围为 $2000m$ 到 $4000m$ ，我机和目标飞行速度变化范围为 $100m/s$ 到 $450m/s$ 。总共生成了 253400 条数据作为训练集。

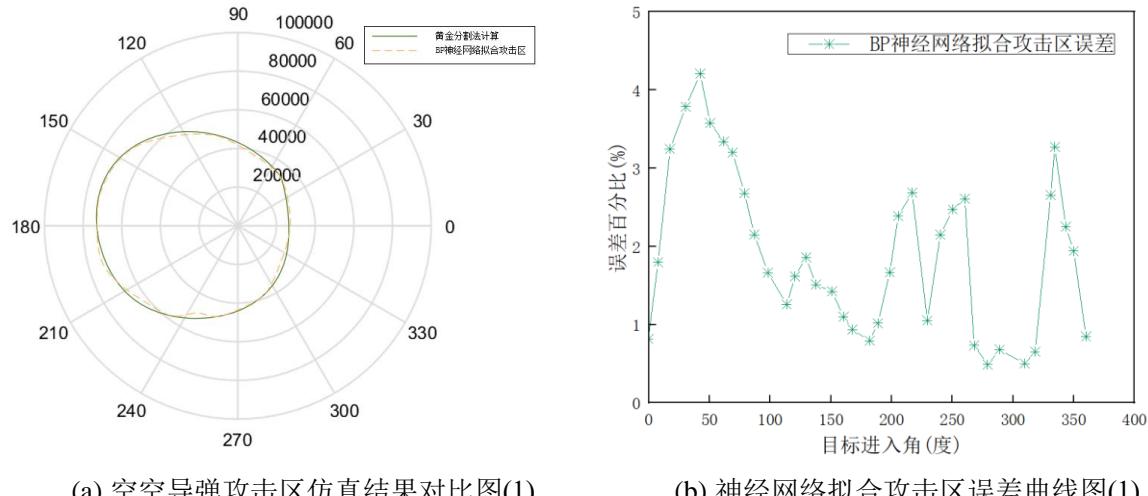
接着，利用 MATLAB 的神经网络工具箱 nntool，构建了 18 个结构一致的 BP 神经网络。这些网络根据各自的 18 份训练集进行独立训练，以确保各自的准确性和效率。网络的具体参数详如表 4-1 所示，基于 4.2.3 节的方法进行仿真。

表 4-1 BP 神经网络参数

| 参数名称 | 参数大小 |
|----------|---------|
| 输入层神经元个数 | 4 |
| 隐含层个数 | 1 |
| 隐含层神经元个数 | 16 |
| 隐含层传递函数 | tansig |
| 输出层神经元个数 | 1 |
| 输出层传递函数 | purelin |
| 最大学习轮回 | 40000 |

完成训练后，通过 BP 神经网络得到的空空导弹攻击区域结果与经典的黄金分割法得出的结果进行比较。

以最大攻击区为例，当本次实验的初始条件为：我机飞行速度 $v_u=150m/s$ 、飞行高度 $h_u=2000m$ ，以及目标的速度为 $v_t=200m/s$ 、高度 $h_t=2000m$ 时，两种算法拟合出的攻击区结果如图 4-5(a)所示，BP 神经网络计算出的攻击区域所需时间为 0.57 秒，其与黄金分割法的结果的误差分布如图 4-5(b)所示。



(a) 空空导弹攻击区仿真结果对比图(1)

(b) 神经网络拟合攻击区误差曲线图(1)

图 4-5 BP 神经网络仿真结果图(1)

当我机飞行速度 $v_u=300m/s$ 、飞行高度 $h_u=4000m$ ，以及目标的速度为 $v_t=250m/s$ 、高度 $h_t=3500m$ 时，两种算法拟合出的攻击区结果如图 4-6(a)所示，BP 神经网络计算出的攻击区域所需时间为 0.62 秒，其与黄金分割法的结果的误差分布如图 4-6(b)所示。

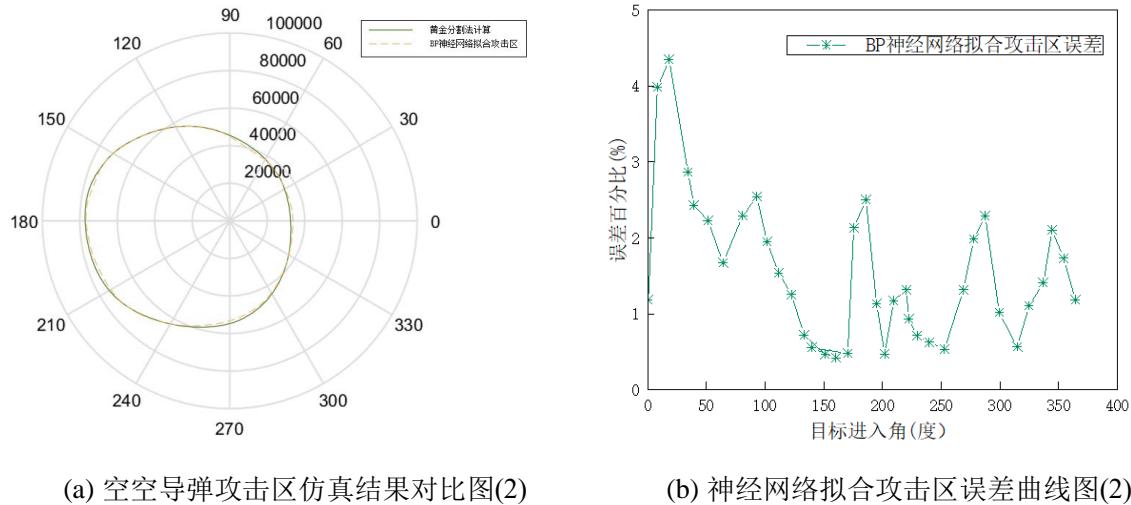


图 4-6 BP 神经网络仿真结果图(2)

由仿真结果可知，采用 BP 神经网路拟合得到的导弹攻击区与使用黄金分割法计算得到的攻击区之间的误差百分比保持在 5% 内，同时采用 BP 神经网络进行拟合的时间均小于 1 秒，远小于黄金分割法的时间，因此该算法的实时性能具有明显优势，同时也基本满足所需的准确度，但是从图中可以看出，其效果仍有不足。

4.2.4 基于改进神经网络的攻击区拟合方法

(1) 解算流程

空战背景中对无人机解算能力的准确度的要求较高，因此无人机必须准确地解算出攻击区以适应战场环境。然而，传统的空空导弹攻击区拟合方法在精度方面还存在提升空间，为了提升无人机在高速动态目标跟踪和精确打击方面的能力，本节提出了一种基于改进神经网络(Tuning of Backpropagation Neural Networks, TBPNN)的攻击区拟合方法，旨在通过对 BP 神经网络权重和阈值的优化，找到 BP 神经网络最优权值和阈值，从而提升解算的精度，增强无人机的作战效率^[192]。

1) 基于 TBPNN 算法的攻击区拟合步骤

首先输入 TBPNN 算法的权重和阈值作为配置参数，再由 BP 神经网络拟合出的攻击区预测输出值与黄金分割法所得到的期望输出值之间的误差绝对值和作为权重和阈值数据的评估分数，计算式如式所示：

$$F = k \left(\sum_{i=1}^n \text{abs}(y_i - o_i) \right) \quad (4-9)$$

其中， k 表示系数， n 表示该神经网络输出的节点个数， y_i 表示网络的第 i 个节点的期望输出值； o_i 表示第 i 个节点的网络预测输出值。

接下来对配置参数进行重组操作，随机生成一系列配置参数，某个配置参数被选中

进行重组的概率是基于其评估分数占所有配置参数评估分数总和的比例。这种方法确保了拥有较高评估分值的配置参数有更大的概率被选中参与下一次迭代。可以用如式所示描述该过程：

$$f_i = \frac{k}{F_i} \quad (4-10)$$

$$p_i = \frac{f_i}{\sum_{i=1}^N f_i} \quad (4-11)$$

其中， k 表示系数； F_i 表示当前权重和阈值的评价指标，该值越小代表配置参数的评估分数越高，因此在配置参数进行选择判断的时候将评估分数放入分母部分； p_i 表示第*i*个体的选择概率； N 表示所有配置参数数量。

将配置参数采用实数来进行编码，因此可以选择实数重组法增加配置参数的多样性。具体而言，所有配置参数中选取了第*k*个配置参数 a_k 与第*l*个配置参数 a_l 在*j*位进行重组，以产生新的配置参数，可以表示为如式所示：

$$\begin{cases} a_{kj} = a_{kj}(1-b) + a_{lj}b \\ a_{lj} = a_{lj}(1-b) + a_{kj}b \end{cases} \quad (4-12)$$

其中， b 为随机数且 $b \in [0,1]$ 。

随后，对第*i*个配置参数中的第*j*位 a_{ij} 进行改造操作，具体的操作方式如式所示：

$$a_{ij} = \begin{cases} a_{ij} + (a_{ij} - a_{\max}) * f(g) & r > 0.5 \\ a_{ij} + (a_{\min} - a_{ij}) * f(g) & r \leq 0.5 \end{cases} \quad (4-13)$$

$$f(g) = r_2 \left(1 - \frac{g}{G_{\max}} \right)^2 \quad (4-14)$$

其中， $a_{ij} \in [a_{\min}, a_{\max}]$ ， r_2 和 r 均为随机数且其中 $r \in [0,1]$ ， g 为当前的迭代次数， G_{\max} 为最大的迭代次数。

本节中，首先，通过反向传播算法调整输入层参数，构建 BP 神经网络算法的基本结构；接着，将 BP 神经网络算法中的权重和阈值输入到的 TBPNN 方法中计算其评估分数，其次，通过选择、重组和改造操作来选出最适合空空导弹攻击区神经网络的权重和阈值作为 BP 神经网络的初始权重和阈值，最后，对导弹攻击区进行预测计算，并通过训练输出结果。整个算法流程详如图 4-7 所示。

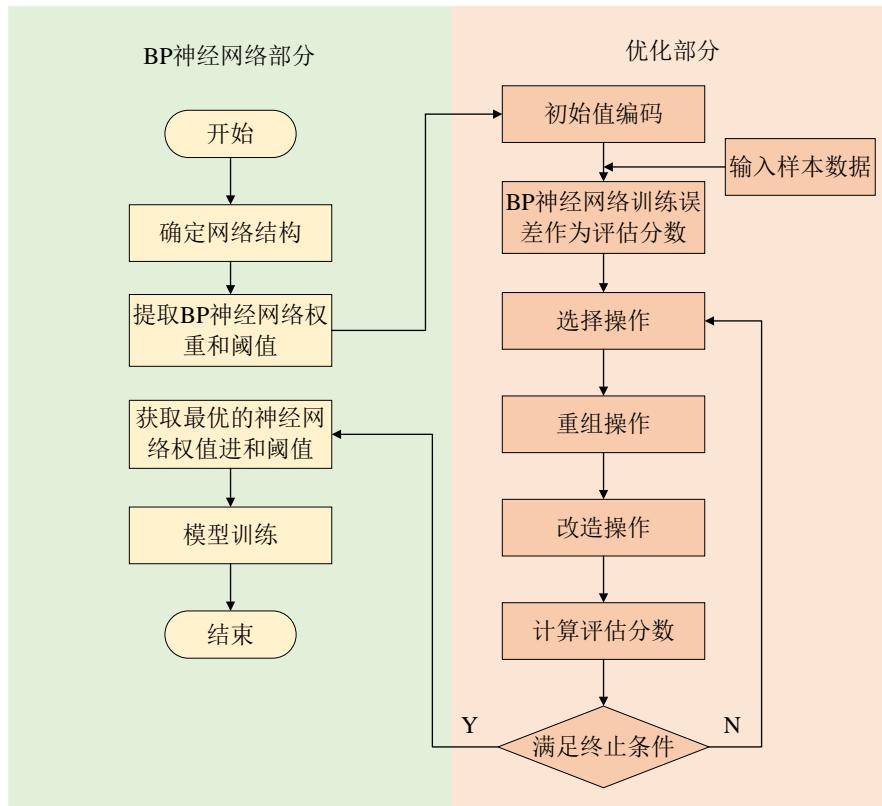


图 4-7 TBPNN 算法流程图

2) 基于 TBPNN 算法的无人机攻击区拟合模型

结合前述的 TBPNN 算法流程，可知该模型输入无人机和目标的实时状态信息，输出预测的最佳攻击区域，同时优化神经网络参数以提高预测的准确性和效率。其中优化过程分为评估分数、优化主循环两个主要模块。

a) 评估分数模块

该模块的主要用于评估每个输入进来的 BP 神经网络的阈值和权值在解算无人机攻击区精度方面的分数。

首先，输入一组神经网络的权重和阈值在 BP 神经网络中进行前向传播，然后，将神经网络的输出与实际的目标或期望值进行比较，计算误差，基于该误差确定评估分数，通常误差越小，评估分数越高，即神经网络预测的准确性更高。最后，输出每个权重和阈值的评估分数，以便用于后续的算法操作。其伪代码如表 4-2 所示。

表 4-2 评估分数模块伪代码

模块：评估分数

```

1: Function score = evaluateScore(configuration, y, o, k)
2:   ErrorSum = 0;
3:   For i = 1:length(y)
4:     ErrorSum = errorSum + abs(y(i) - o(i));
5:   End For

```

模块: 评估分数

```
6: score = k / (1 + errorSum); % 误差越小, 评估分数越高
7: Return score
```

b) 优化主循环模块

该模块输入配置参数及其评估分数、迭代次数、以及重组率和改造率，输出是经过一系列优化操作后形成的新配置参数，将用于下一轮迭代。其伪代码如表 4-3 所示。

表 4-3 优化主循环模块伪代码

模块: 优化主循环

```
1: 输入: 配置参数 i、配置参数 j 和重组系数 b
2: 输出: 新配置参数
3: For 每个参数 = 1:length do:
4:   生成随机数 r;
5:   If r < 0.5
6:     新配置参数 i= 配置参数 i* (1 - b) +配置参数 j * b;
7:     新配置参数 j =配置参数 j * (1 - b) +配置参数 i* b;
8:   End If
9: End For
10: Returen 新配置参数 i,新配置参数 j
11: 输入: 新配置参数 i 当前迭代次数 iteration, 最大迭代次数 max_iterations, 最小值 a_min 和
      最大值 a_max
12: 输出: 经过改造的新配置参数 i
13: For 每个参数 = 1:length do:
14:   生成随机数 r, f, g
15:   If r < 0.5
16:     新配置参数 i =新配置参数 i + (新配置参数 i - a_min) * f * (1 - iteration/max_iterations)^2;
17:   Else
18:     新配置参数 i =新配置参数 i+ (a_max -新配置参数 i) * f * (1 - iteration/max_iterations)^2;
19:   End If
20: End For
21: For 每次迭代 do:
22:   For i = 1:length(selected_configurations)/2
23:     配置参数 i= 被选择的配置参数 (i*2-1);
24:     配置参数 j = 被选择的配置参数 (j *2);
25:     b = rand(); % 用于重组的随机数 b
26:     If rand() < real_crossover_rate
27:       [新配置参数 i,新配置参数 j ] = performRealCrossover(新配置参数 i, 配置参数 j , b);
28:       更新配置参数=[更新配置参数, 新配置参数 i,新配置参数 j ];
29:     End If
30:   End For
31: End For
32: For j = 1:length(new_generation)
33:   新配置参数= 更新配置参 j ;
34:   If rand() < real_mutation_rate
35:     新配置参数= performRealMutation(新配置参数, iteration, max_iterations, a_min, a_max);
36:     更新配置参 j = 新配置参数;
37:   End If
38: End For
```

(2) 仿真结果与分析

本节同样设定了不同的初始条件，采用黄金分割法模拟出理论导弹攻击区数据，在仿真设置中，我机飞行角度和目标进入角度的变化范围为 $[0^\circ, 360^\circ]$ ，我机和目标的飞行高度的变化范围为 $2000m$ 到 $4000m$ ，我机和目标飞行速度变化范围为 $100m/s$ 到 $450m/s$ 。总共生成了232400条数据作为训练集。

设置BP神经网络的训练次数不超过1000次，学习速率为0.01，训练过程中误差不大于0.0001，初始参数规模 $N=10$ ，最大迭代数 $G_{\max} \leq 30$ ，重组概率和改造概率分别为0.8和0.2，基于4.2.4节的内容进行仿真。

以最大攻击区为例，当本次实验的初始条件为：我机飞行速度 $v_u=150m/s$ 、飞行高度 $h_u=2000m$ ，以及目标的速度为 $v_t=200m/s$ 、高度 $h_t=2000m$ 时，两种算法拟合出的攻击区结果如图4-8(a)所示，TBPNN神经网络计算出的攻击区域所需时间为0.18秒，其与黄金分割法的结果的误差分布如图4-8(b)所示。

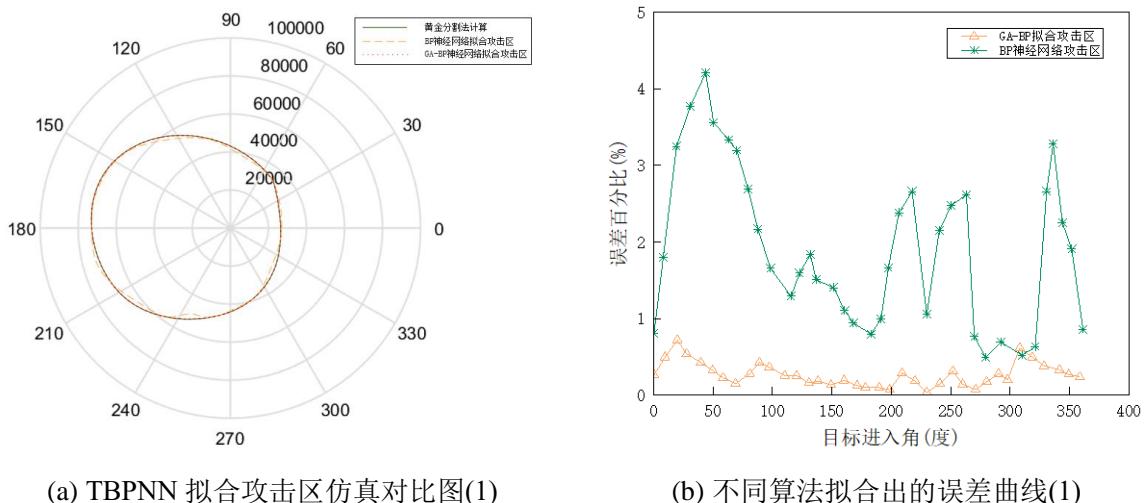


图 4-8 TBPNN 仿真结果图(1)

当我机飞行速度 $v_u=300m/s$ 、飞行高度 $h_u=4000m$ ，以及目标的速度为 $v_t=250m/s$ 、高度 $h_t=3500m$ 时，两种算法拟合出的攻击区结果如图4-9(a)所示，TBPNN神经网络计算出的攻击区域所需时间为0.23秒，其与黄金分割法的结果的误差分布如图4-9(b)所示。

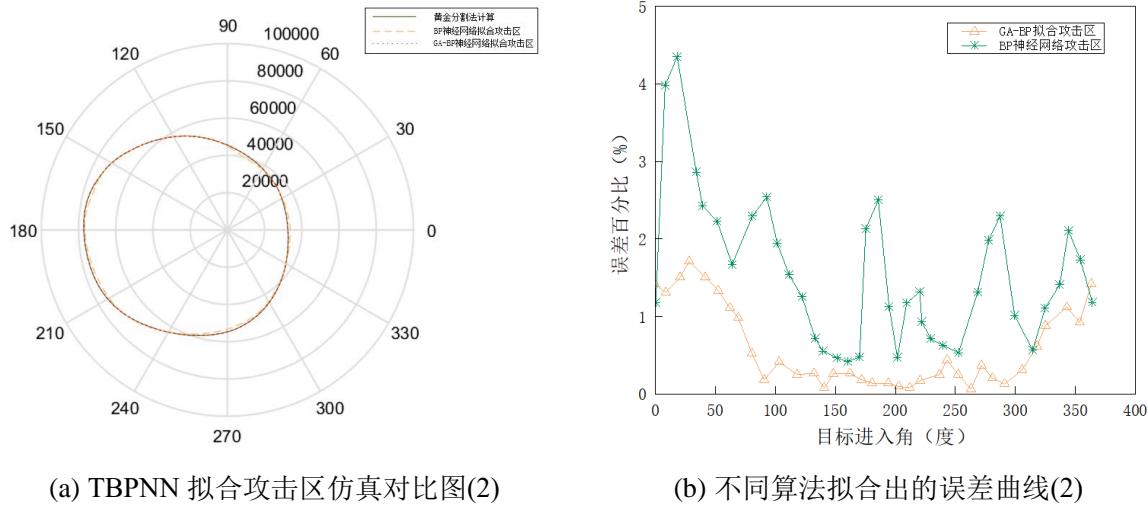


图 4-9 TBPNN 仿真结果图(2)

仿真结果表明, TBPNN 神经网络拟合的攻击区域结果比 BP 神经网络更接近期望值, 其误差百分比均在 2% 以内, 准确度显著高于 BP 神经网络; 在解算时间方面, 经 GA 优化后的 BP 神经网络的解算时间均低于 0.3 秒, 大幅提升了计算效率。因此, 采用 TBPNN 神经网络对攻击区进行拟合可以有效满足后续无人机编队协同空战战术决策的实时性和准确度需求。

4.3 基于改进 SAC 算法的无人机空战机动决策方法

4.3.1 状态空间

无人机机动决策模型的状态空间用于描述空中战斗的全部情况, 该情况在任何时候都可以通过无人机状态来确定, 无人机状态包含位置矢量 $\mathbf{R} = [x, y]$ 、速度 v 、航向角 φ 和距离 d 中的相对信息。为了统一每个变量的范围并提高网络学习的效率, 每个状态变量都被归一化为 $[0,1]$ 。状态空间定义为一个 6 元组 $S = [x, y, v, \varphi, d, q]$ 。具体空战态势示意图如图 4-10 所示。

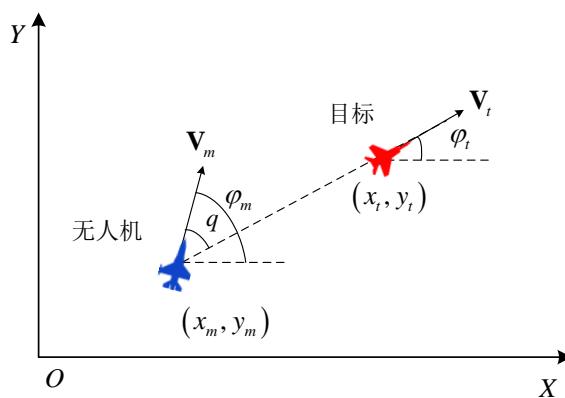


图 4-10 空战态势图

4.3.2 动作空间

无人机运动方程表明，在有效积分步长 dt 内设置 dv 、 $d\varphi$ ，无人机可以在空间中实现一系列的机动过程。因此，可以得到无人机飞行动作空间，即： $A = [dv, d\varphi]$ 。

4.3.3 奖励函数

在空中战斗中，无人机通过不断估计奖励完成相应的动作，以使敌方无人机进入导弹的攻击区域。战斗中的双方在 OXY 坐标系中建模，设定我机的速度矢量为 $\mathbf{V}_m = (v_{xm}, v_{ym})$ 、位置矢量为 $\mathbf{L}_m = (x_m, y_m)$ 。敌机的速度矢量为 $\mathbf{V}_t = (v_{xt}, v_{yt})$ 、位置矢量为 $\mathbf{L}_t = (x_t, y_t)$ 。双方的相对位置由 $\mathbf{D} = \mathbf{L}_m - \mathbf{L}_t$ 表示，双方的距离为 $d = \|\mathbf{D}\|_2$ 。 \mathbf{V}_m 和 \mathbf{D} 之间的角度是相对方位角为 q ，目标速度矢量与两者之间连线的夹角为 q_t 。无人机空战示意图如图 4-11 所示。

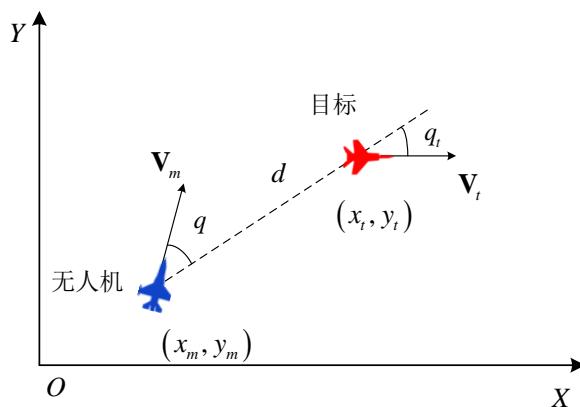


图 4-11 无人机空战示意图

机动决策的目标是无人机在空战中进行机动，使敌方无人机进入导弹的攻击区域。为了成功攻击敌人，我机应该尽量朝向敌人飞行，其相对方位角 q 应该最小化。随后定义无人机在空中战斗中的角度优势为：

$$\eta_A = -q / 180^\circ \quad (4-15)$$

其中， $q = \arccos((\mathbf{D} \times \mathbf{V}_m) / (\|\mathbf{D}\|_2 \cdot \|\mathbf{V}_m\|_2))$ 。 η_A 越小表示无人机在空中战斗中的优势越大。

距离在空中战斗中也是一个重要因素。导弹无法攻击超出其攻击区，因此在解算出最大攻击距离 D_{\max} 和最小攻击距离 D_{\min} 后无人机在空中战斗中的距离优势定义为：

$$\eta_D = -\frac{d}{(5 \cdot D_{\max})} + \delta \quad (4-16)$$

$$\delta = \begin{cases} -1, & \text{if } d < D_{\min} \\ 0, & \text{else} \end{cases} \quad (4-17)$$

结合角度优势和距离优势，无人机在空中战斗中的优势评估函数定义为：

$$\eta = \eta_A + \eta_D \quad (4-18)$$

本论文基于无人机在空战中的优势评估函数 η 设计了奖励函数，以无人机相对于敌机的战术位置和能力优势为依据，通过连续的负反馈激励无人机保持或提升其战术地位。在无人机处于明显优势的情况下，奖励函数会提供更大的正向激励，鼓励无人机继续执行和优化当前的机动决策。同时，为避免不必要的战术风险，当无人机越界或有撞击、摧毁友机的危险时，会施加相应的惩罚。奖励函数是对智能体在当前空战情况下行动的实时评估，它是基于无人机在空战中的优势评估函数设计的，该函数是一个负的连续函数，同时，当我机在空战中处于较大优势时，它会获得更大的激励以保留已经探索的策略。定义机动决策函数如下：

$$\eta_{ai} = \begin{cases} 3, & |q| < q_{\max} \text{ 或 } D_{\min} < |d| < D_{\max} \\ 0, & \text{其他} \end{cases} \quad (4-19)$$

设置我机超越环境边界时给予惩罚奖励：

$$r_{m,b} = \begin{cases} -5, & \text{我机超越边界} \\ 0, & \text{我机未超越边界} \end{cases} \quad (4-20)$$

撞击友机时给予惩罚奖励：

$$r_{m,f,s} = \begin{cases} -3, & \text{我机撞击友机} \\ 0, & \text{我机未撞击友机} \end{cases} \quad (4-21)$$

摧毁友机时给予惩罚奖励：

$$r_{m,f,d} = \begin{cases} -2, & \text{我机摧毁友方} \\ 0, & \text{我机未摧毁友方} \end{cases} \quad (4-22)$$

则总奖励为：

$$r = \eta + \eta_{ai} + r_{m,b} + r_{m,f,s} + r_{m,f,d} \quad (4-23)$$

其中， η 为无人机在空中战斗中的优势评估函数见式(4-18)。

发射条件只有在获得目标的位置和速度等数据、计算导弹发射元素并将数据加载到导弹上之后，火控系统检测到目标后才能达到。这意味着在发射空对空导弹之前需要锁定目标的一定时间 t_{\max} 。假设敌人连续处于导弹的攻击区域内的时间为 t_{in} ，当满足方程(4-24)时，便发射导弹。

$$\begin{cases} D_{\min} < D < D_{\max} \\ \eta_m < \eta_{m\max} \\ t_{in} > t_{\max} \end{cases} \quad (4-24)$$

4.3.4 E-SAC 算法

SAC 算法 (Soft Actor-Critic, SAC) [193] 是一种高效的基于 Actor-Critic 框架的强化学习方法。“Actor”代表一个策略函数，它负责根据当前的环境状态选择动作；“Critic”代表

一个价值函数，它评估在给定状态下采取特定动作的好坏，这种架构使模型能够同时学习最佳策略（即应采取哪些动作）和评估这些策略的效果（即这些动作的效果如何）。

SAC 算法作为一种深度强化学习算法，在处理连续动作空间和高维状态空间的问题方面表现出色，但也存在一些局限性，如训练收敛速度慢和对超参数的高敏感性。为了解决这些问题，本文提出了一种改进的 SAC 算法，称为 E-SAC(Expert-Soft Actor-Critic) 算法。E-SAC 算法通过引入 Expert Actor 来增强智能体的探索效率，并优化学习过程。

E-SAC 算法可以对空战中的无人机自主机动决策模型进行构建。它是一个带有策略网络 π_θ 、两个 soft-Q 网络 Q_{φ_1} 和 Q_{φ_2} 以及两个目标 soft-Q 网络 Q'_{φ_1} 和 Q'_{φ_2} 的演员-评论家架构，其中 θ , φ_1 , φ_2 , φ'_1 和 φ'_2 分别对应网络的参数。

在 E-SAC 算法中，策略 $\pi_\theta(\cdot | s_t)$ 服从某种分布，策略的随机性由策略的熵 $H(\pi_\theta(\cdot | s_t))$ 来衡量。为了在探索最优策略的同时最大化策略的探索性，策略的熵被引入到回报函数中。累积期望回报被定义为：

$$J(\pi) = \sum_{t=0}^T E_{(s_t, a_t) \sim \rho_\pi} [r(s_t, a_t) + \alpha H(\pi(\cdot | s_t))] \quad (4-25)$$

其中， α 是熵正则化系数，表示回报中熵的比例，控制策略的随机性。

动作由策略网络 π_θ 和噪声生成。噪声 τ 从标准正态分布中采样。因此，动作 a_t 的生成过程如方程(4-26)至(4-27)所示。

$$\mu, \sigma = \pi_\theta(s_t) \quad (4-26)$$

$$a_t = \tanh(\mu + \tau * \exp(\sigma)) \quad (4-27)$$

Soft-Q 网络的输入是状态 s_t 、动作 a_t ，输出是 Q_{soft} 。Soft-Q 函数定义如下：

$$\begin{aligned} Q_{\text{soft}}(s_t, a_t) &= E_{(s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) + \alpha \sum_{t=1}^{\infty} \gamma^t H(\pi(\cdot | s_t)) \right] \\ &= r(s_t, a_t) + \gamma E_{(s_{t+1}, a_{t+1})} \times \left[Q_{\text{soft}}(s_{t+1}, a_{t+1}) - \alpha \log(\pi(a_{t+1} | s_{t+1})) \right] \end{aligned} \quad (4-28)$$

该算法首先通过改变初始的空战场景设置不同的 Expert Actor 采样环境，以获取 Expert 经验样本。在某个环境中，无人机依据当前状态使用 Expert Actor 做出策略决策，获得相应的行动 a_t ，并通过执行该行动获取奖励 r_t 以及更新后的状态 s_{t+1} ，此过程持续直到成功击败敌机。在不同的空战环境中重复此过程，并将收集到的 Expert 经验样本 (s_t, r_t, s_{t+1}, a_t) 储存于 Expert 经验回放池中。从经验池中随机抽取样本对策略和 Soft-Q 网络进行训练。

策略网络是根据损失函数 $J_\pi(\theta)$ 进行更新的，如方程(4-29)所示：

$$J_\pi(\theta) = E_{s_t \sim R_{a_t} \sim \pi_\theta} [\log \pi_\theta(a_t | s_t) - Q_\varphi(s_t, a_t)] \quad (4-29)$$

Soft-Q 网络是根据损失函数 $J_Q(\varphi_i)$ 进行更新的，如方程(4-30)所示：

$$J_Q(\varphi_i) = E_{(s_t, a_t, s_{t+1}) \sim R, a_{t+1} \sim \pi_\theta} \left[\frac{1}{2} Q_{\varphi_i}(s_t, a_t) - (r(s_t, a_t) + \gamma (Q_{\varphi'}(s_{t+1}, a_{t+1}) - \alpha \log \pi_\theta(a_{t+1} | s_{t+1}))^2 \right] \quad (4-30)$$

其中， $Q_{\varphi'}(s_{t+1}, a_{t+1}) = \min(Q_{\varphi_1}(s_{t+1}, a_{t+1}), Q_{\varphi_2}(s_{t+1}, a_{t+1}))$ 。

在由深度强化学习支持的训练过程中，需要在早期阶段进行大量的探索，以确保策略获得不同的样本以进行优化。在后期稳定阶段，需要相对较小的探索来稳定策略。因此，根据训练过程动态调整 α ，具体如下：

$$J(\alpha) = E[-\alpha \log \pi_t(a_t | \pi_t) - \alpha H_0] \quad (4-31)$$

其中， H_0 是目标熵。

E-SAC 算法在训练初期通过将一定比例的 Expert Actor 经验样本混合到算法的经验回放缓冲区中，丰富了样本多样性，有助于指导智能体进行更全面的环境探索。为了减少专家演员在智能体后期决策过程中可能产生的不利影响，Expert Actor 经验样本与通过标准 SAC 算法探索获得的样本之间的比例会动态调整。通过这种方式，E-SAC 算法不仅促进了智能体的深度探索，还有效避免了算法过早收敛于局部最优解，同时加快了训练进程。E-SAC 算法的结构如图 4-12 所示。

在探索初期，E-SAC 算法通过将一定比例的 Expert 经验样本混合到算法的经验回放缓冲区，从而丰富了样本的多样性，并指导智能体进行全面的环境探索。同时，为减少 Expert Actor 对智能体后期决策过程可能产生的不利影响，Expert 样本与通过 SAC 算法探索得到的样本之间的比例将会动态调整。

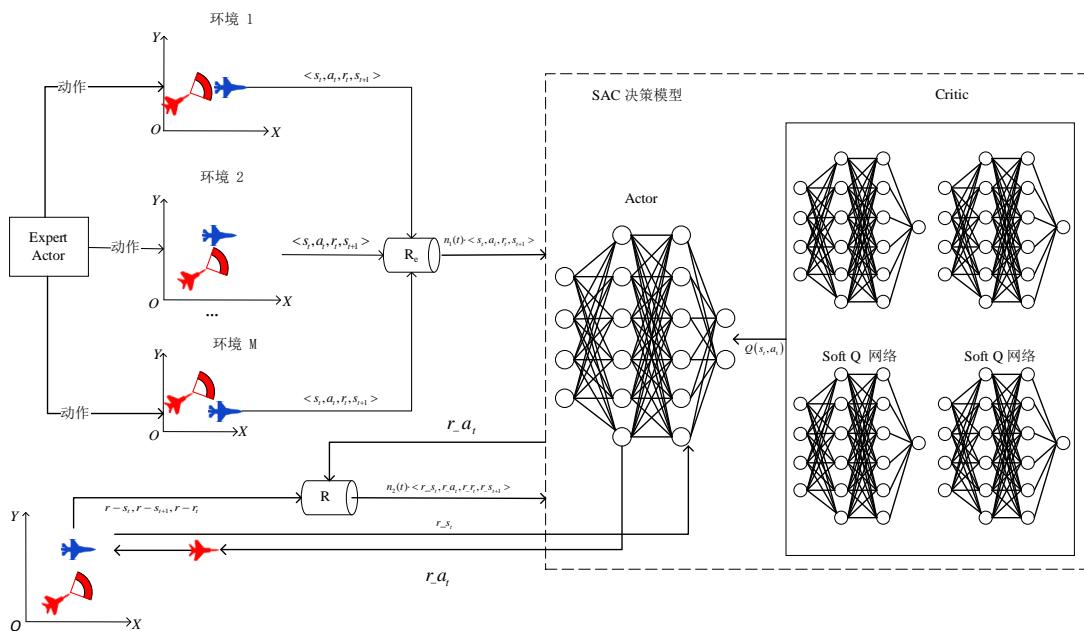


图 4-12 E-SAC 算法示意图

E-SAC 算法通过提高训练样本的多样性，不仅引导了智能体的深度探索，还有效预防了算法早熟收敛于局部最优解，同时显著加快了训练进程。该算法的训练流程详如表 4-4 所示。

表 4-4 基于 E-SAC 的无人机自主机动决策算法伪代码

算法: E-SAC

```

1: 初始化 E-SAC 算法的相关网络参数、空战环境、回放缓冲区 R、batch 大小 (batch_size)、以
   E-SAC 算法开始做出决策的步骤数 (start_size)
2: For episode = 1, N do
3:   获取无人机的初始状态  $s_t$ 
4:   For t = 1, T do:
5:     If 放入缓冲区 R 的长度小于 start_size
6:       随机选择一个动作  $a_t$ 
7:     Else 使用策略  $\pi_\theta(s_{a_t})$  来选择动作
8:     无人机执行动作  $a_t$ ，获取奖励和新状态  $s_{t+1}$ 
9:     将  $(s_t, r_t, s_{t+1}, a_t)$  存放到缓冲区 R 中
10:    If 回放缓冲区 R 的长度大于 batch_size:
11:      If 回放缓冲区 R 的长度小于 start_size
12:        从专家经验回放缓冲区 R 中取出 n 组样本。
13:      Else 从回放缓冲区 R 中取出  $n_2$  组样本与  $n_1$  合并
14:      If 回放缓冲区 R 的长度是偶数,
15:         $n_1 = n_1 - 1$ 
16:      End if
17:      更新 E-SAC 算法的神经网络
18:    End if
19:  End for
20: End for

```

关于 Expert-Actor 模型，其建立方法如下：当我机发射导弹攻击敌机时，相对方位角 q 和相对距离 d 应同时满足发射条件。当采用 Expert 方法控制无人机运动时，首先可以将 q 减少到一定范围内，以使我机处于面对敌机的正面或尾部攻击位置，然后，我机增加速度以迎接敌人，直到敌人进入之前所计算好的导弹攻击区范围内。基于我机运动模型的 Expert-Actor 的数学模型如式(4-32)-(4-36)所示：

$$\begin{cases} \Delta X = x_t - x_m \\ \Delta Y = y_t - y_m \end{cases} \quad (4-32)$$

$$D_\varphi = \begin{cases} 180^\circ + \arctan\left(\frac{\Delta X}{\Delta Y}\right)/\pi * 180^\circ, & \Delta Y < 0, \Delta X > 0 \\ \arctan\left(\frac{\Delta X}{\Delta Y}\right)/\pi * 180^\circ - 180^\circ, & \Delta Y < 0, \Delta X < 0 \\ \arctan\left(\frac{\Delta X}{\Delta Y}\right)/\pi * 180^\circ, & \text{else} \end{cases} \quad (4-33)$$

其中， ΔX 、 ΔY 分别代表敌人相对于无人机的位置向量分量， D_φ 代表两边相对位置向量 \mathbf{D} 与 X 轴之间的角度。

设 $\Delta v = v_{t+1} - v_t$ 、 $\Delta \varphi = D_\varphi - \varphi$ ，设我机速度和航向角变化需要在以下范围内进行控制。

$$\begin{cases} -\Delta v_0 \leq \Delta v \leq \Delta v_0 \\ -\Delta \varphi_0 \leq \Delta \varphi \leq \Delta \varphi_0 \end{cases} \quad (4-34)$$

无人机速度变化 dv 和航向角变化 $d\varphi$ 分别可以通过式(4-35)和式(4-36)算出。

$$dv = \begin{cases} 0, & \eta_m > 30^\circ \\ \Delta v_0, & \eta_m < 30^\circ \cap d > D_{\max} \cap v < v_{\max} \\ v_{\max} - v, & \eta_m < 30^\circ \cap d > D_{\max} \cap v > v_{\max} \\ -\Delta v_0, & \eta_m < 30^\circ \cap d < D_{\max} \cap \Delta v < -\Delta v_0 \\ \Delta v, & \eta_m < 30^\circ \cap d < D_{\max} \cap -\Delta v_0 < \Delta v < \Delta v_0 \\ \Delta v_0, & \eta_m < 30^\circ \cap d < D_{\max} \cap \Delta v > \Delta v_0 \end{cases} \quad (4-35)$$

$$d\varphi = \begin{cases} -\Delta \varphi_0, & \Delta \varphi < -\Delta \varphi_0 \\ \Delta \varphi, & -\Delta \varphi_0 < \Delta \varphi < \Delta \varphi_0 \\ \Delta \varphi_0, & \Delta \varphi > \Delta \varphi_0 \end{cases} \quad (4-36)$$

其中， v_{\max} 是我机的最大速度。

4.4 仿真结果与分析

4.4.1 实验环境设定

本次仿真采用 Python 语言，开发环境为 PyCharm，训练环境采取 1v1 方式，首先固定本机进行跟踪训练。输入状态维度为 2，动作维度为 1，操作系统为 Windows 10 系统，仿真环境基于 MaCA 平台，该平台可支持自定义作战场景、规模、智能体数量和种类等。

本文基于 MaCA 环境实现无人机智能作战算法的仿真，搭建仿真环境逻辑和运行流程如图 4-13 所示。

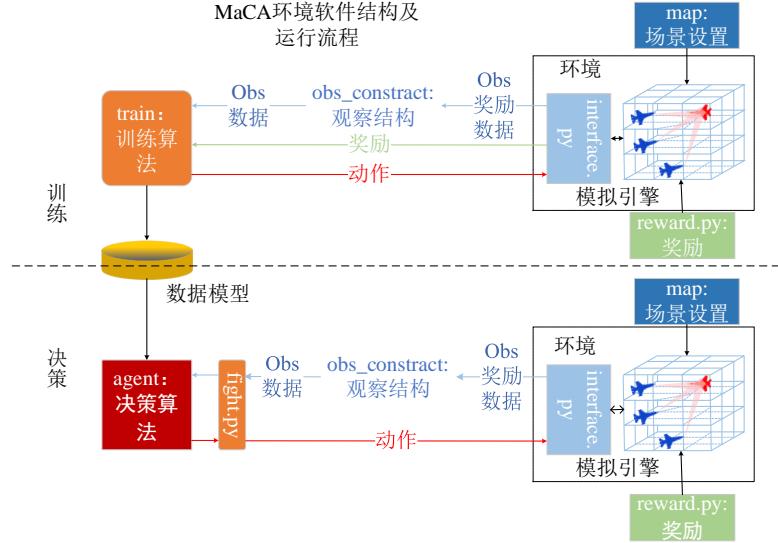


图 4-13 MaCA 环境软件结构及运行流程

设置地图尺寸 12000×12000 米，敌机策略采用 MaCA 平台中的 `fix_rule_no_att` 固有策略，我机采用本文建立的 E-SAC 算法，具体参数配置如表 4-5 所示。

表 4-5 无人机空战机动决策仿真参数表

| 参数名称 | 参数大小 |
|---------------------------------|------------|
| 最大离轴发射角 q_{\max} | 30° |
| 目标锁定时间 t_{\max} | 2s |
| 无人机最大速度 v_{\max} | 210m/s |
| 无人机最大变化速度 Δv_0 | 20 |
| 无人机航向角最大变化角度 $\Delta \varphi_0$ | 6° |
| 网络隐含层个数 | 2 |
| 网络隐含层神经元个数 | 256 |
| 学习速率 | 0.0003 |
| 累积回报折扣因子 | 0.99 |
| 目标熵 H_0 | -3 |
| 熵的正则化系数初始值 | 1 |
| 目标网络更新间隔步数 | 300 |
| 经验池容量 | 100,000 |
| 经验池提取样本数量 | 200 |
| 激活函数 | ReLU |
| 优化器 Adam 软更新系数 | 0.005 |

利用 4.2 节的方法对无人机空战攻击区实时解算并将解算结果传输至机动决策模型。为检验 E-SAC 算法在自主完成机动决策任务方面的有效性，本节设计了 4 种不同的模拟环境，以研究我机与敌机的初始距离和初始相对方位角的改变对不同算法训练过程的

影响。其中，在环境1和环境2中，我机与敌机有相同的初始距离，但初始相对方位角不同，在环境1和环境3中，我机与敌机有不同的初始距离，但初始相对方位角相同，在环境3和环境4中，我机与敌机有不同的初始距离，但初始相对方位角相同。基于设置的环境1-4进行分析，对比不同初始距离和不同初始方位角场景下对无人机机动决策的影响。具体模拟环境的空战初始态势如表4-6所示。

表 4-6 空战初始态势

| 环境编号 | 初始距离 (m) | 初始相对方位角 (°) |
|------|----------|-------------|
| 1 | 6000 | 90 |
| 2 | 6000 | 60 |
| 3 | 9000 | 30 |
| 4 | 9000 | 60 |

根据4.3节的模型，在4个环境中分别采用了SAC算法和增加了专家演员的E-SAC算法对无人机机动决策进行训练。在E-SAC算法的应用中，Expert Actor为SAC算法提供了从第1到256个训练回合的Expert样本，并逐渐降低了这些样本在总样本中的比例，直到在256个训练回合后完全停止使用Expert样本。

4.4.2 训练结果与分析

在实验过程中，记录每个回合我机所获得的总奖励和训练结果。为了更直观地对比两种算法的性能，对经过训练的我机（红色无人机），在环境1到环境4中进行测试，并详细记录了双方的战斗过程。作为对照，敌机（蓝色无人机）遵循MaCA平台中封装好的fix_rule_no_att算法策略进行机动，以提供一致的比较基准。通过这些实验设计和记录，分析和评估各种算法在复杂空战环境中的性能和适应性。

基于如式(4-20)-(4-23)所示设置的奖励函数，分别利用SAC和E-SAC算法的训练无人机机动决策的测试结果如表4-7至表4-10所示。

表 4-7 环境1训练结果

| 环境1 | SAC | E-SAC |
|---------|--------|--------|
| 最大奖励 | 518.36 | 584.34 |
| 奖励收敛回合数 | 384 | 365 |

表 4-8 环境2训练结果

| 环境2 | SAC | E-SAC |
|---------|--------|--------|
| 最大奖励 | 392.33 | 498.78 |
| 奖励收敛回合数 | 466 | 406 |

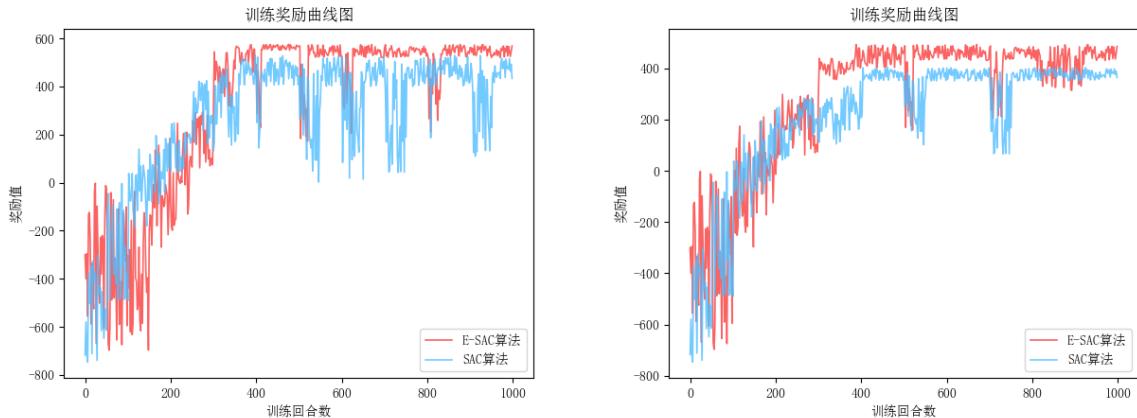
表 4-9 环境 3 训练结果

| 环境 3 | SAC | E-SAC |
|---------|--------|--------|
| 最大奖励 | 422.25 | 522.39 |
| 奖励收敛回合数 | 319 | 531 |

表 4-10 环境 4 训练结果

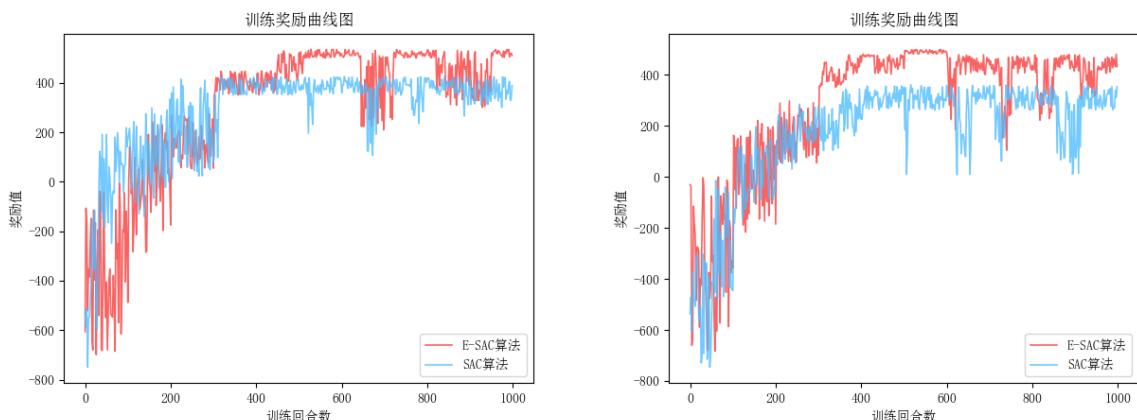
| 环境 4 | SAC | E-SAC |
|---------|--------|--------|
| 最大奖励 | 329.18 | 479.01 |
| 奖励收敛回合数 | 358.1 | 404 |

两种算法的训练奖励曲线图如图 4-14 所示。



(a) 环境 1 下我机的训练奖励曲线

(b) 环境 2 下我机的训练奖励曲线



(c) 环境 3 下我机的训练奖励曲线

(d) 环境 4 下我机的训练奖励曲线

图 4-14 训练奖励曲线

由实验结果可知，两种算法均可以在环境 1、环境 2、环境 3 和环境 4 中完成对目标的追踪与打击的机动决策训练任务。

将两种算法的训练结果在不同的环境下进行对比，在环境 1 实验结果中，两种算法

训练的奖励值较接近且具有相同的收敛趋势，而相较之传统的 SAC 算法，本文所提的 E-SAC 算法训练速度较快，提前了 19 个回合收敛，且 SAC 算法训练奖励函数曲线波动较大，表明了算法的不稳定性；在环境 2 中，本文所提的 E-SAC 算法最大奖励值优于传统的 SAC 算法，且 E-SAC 算法收敛更快，提前 60 个回合就达到了最大回报；环境 3 中的实验结果表明 SAC 算法在该环境下陷入局部最优解，只获得了 422.25 的奖励，而 E-SAC 算法获得了 522.39 的奖励；环境 4 中的我机与敌机的初始距离和相对方位角较大，SAC 算法只获得了 329.18 的奖励，而 E-SAC 算法在经过 404 个回合的训练后，获得了 479.01 的奖励。

综上分析可知，E-SAC 算法不仅可以获得更高的奖励，还能够用更少的步骤完成任务，即，本文所提的 E-SAC 算法针对无人机机动决策在训练效率、收敛速度和奖励方面比传统的 SAC 算法有显著提升。

下面对环境 1 和 4 中的仿真结果进行展示。首先，在环境 1 中，红方与蓝方的初始相对方位角为 30 度，初始距离为 6000 米。环境 1 的空战过程中无人机机动决策示意图如图 4-15 所示。

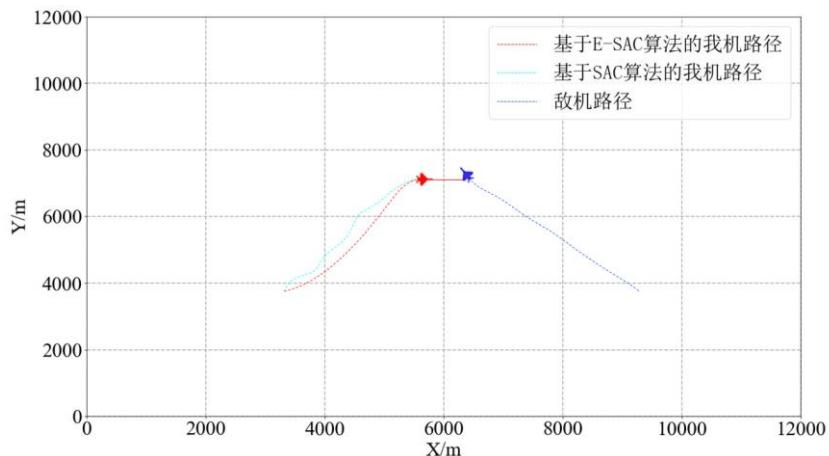


图 4-15 环境 1 中无人机一对一机动决策训练仿真过程

如图 4-15 所示，E-SAC 算法和 SAC 算法在决策过程中展现出相近的机动决策过程，可以看出我方无人机一开始便迅速调整航向，降低与敌机的相对方位角，以尾追攻击姿态使得我机进入之前所计算出的导弹攻击区，然后，迅速调整速度以缩短与敌机的距离。E-SAC 算法在减小相对方位角方面表现出了更高的稳定性和效率，显著优于传统的 SAC 算法，实验结果证明了 E-SAC 算法在训练机动决策时的有效性。

在环境 4 中，设置敌我双方的初始距离增加为 9000 米，角度增加为 60 度。环境 4 的空战过程中无人机机动决策示意图如图 4-16 所示。

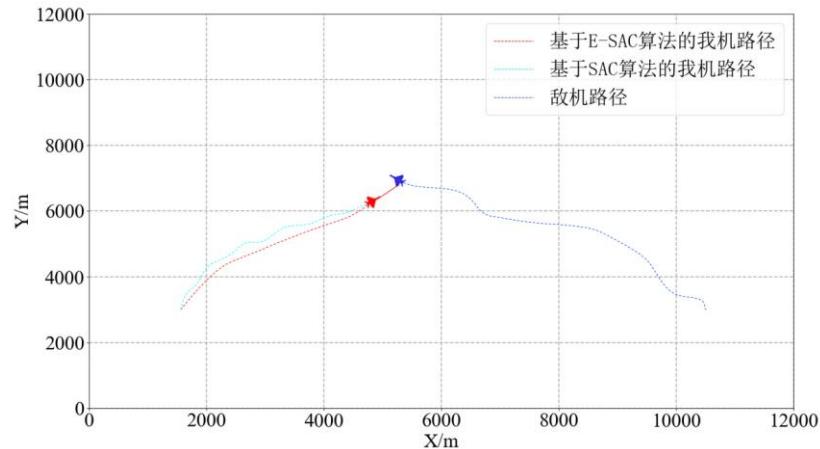


图 4-16 环境 4 中无人机一对一机动决策训练仿真过程

如图 4-16 所示, E-SAC 算法和 SAC 算法的决策过程均是首先改变方向, 减少相对方位角, 然后缩短距离, 并最终使到达攻击区范围内。仿真结果表明, 两种算法均能进行对敌机追踪与打击的机动决策。相对而言, 本文所提的 E-SAC 算法的无人机相对距离和方位角的变化较为稳定。

实验结果表明, E-SAC 算法可以增强探索过程, 并有效处理 SAC 算法可能面临的局部收敛问题, 从而快速获得无人机在空中作战过程中的最佳机动决策策略。

4.5 本章小结

本章针对无人机进入作战区域后的机动决策进行研究, 解决了无人机空战机动决策问题, 为下一章无人机编队协同策略模型提供可靠的机动决策元策略模型。首先, 根据攻击区解算的相关模型, 采用 BP 神经网络对攻击区进行解算; 其次, 将解算出的攻击区作为终端约束, 设计了基于 E-SAC 的无人机编队自主机动决策方法; 最后, 仿真结果表明, 本文所提出的 E-SAC 算法能够很好地引导无人机编队进行机动决策以追踪和击打敌机。

第5章 基于分层强化学习的编队协同策略模型

本章以分层强化学习为研究理论工具，研究了无人机编队在整个空战过程中的战术决策方法。首先，设计了编队协同空战过程元策略模型；其次，建立了基于分层强化学习算法的编队协同策略模型；最后，对无人机编队的空战战术协同策略模型进行仿真，验证算法的有效性。

5.1 问题分析

无人机编队协同策略是指无人机编队在空中作战时的决策选择和机动协调，需要设计合理的探测与攻击策略以实现高效的编队协同作战能力，从而提高无人机编队的任务执行效率和生存能力。

本章基于分层强化学习算法，对多无人机空战中的无人机编队协同策略模型进行研究。首先，对编队协同决策元策略模型进行设计，将第四章所设计的机动决策模型嵌入为编队协同策略模型中的机动决策元策略模型，并基于专家经验建立了雷达开关、主动干扰、主动探测和队形转换模型；其次，对分层强化学习中的 Option-Critic 算法进行改进，并选择合适的状态空间和动作空间；最后，对整个基于分层强化学习的编队协同智能空战策略模型进行仿真验证。本章具体研究内容如图 5-1 所示。

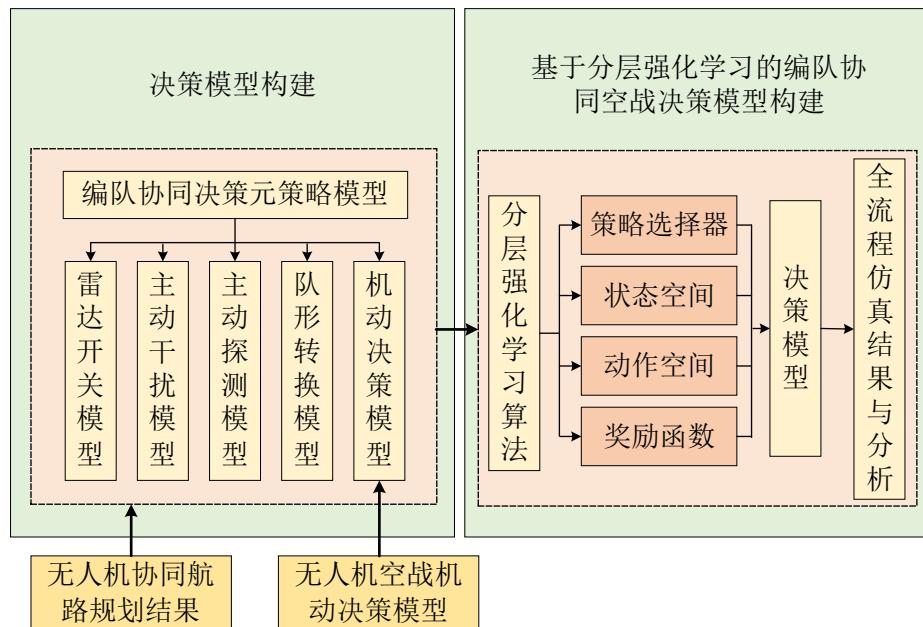


图 5-1 基于分层强化学习的编队协同策略模型研究内容

5.2 编队协同决策元策略模型

本文将无人机编队协同策略模型中的元策略定义为雷达开关策略、主动干扰策略、

主动探测策略、队形转换策略和机动决策策略这五个部分，无人机编队协同空战策略模型的分层结构如图 5-2 所示。

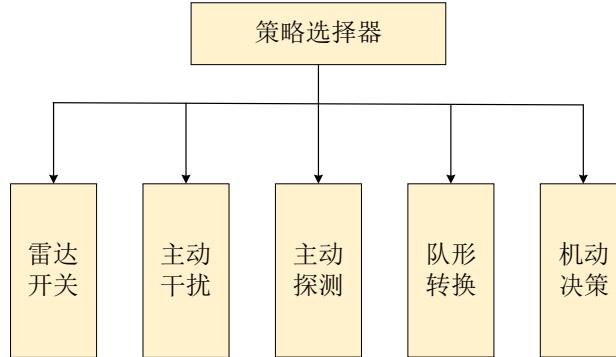


图 5-2 无人机编队协同空战策略模型的分层结构

在如图 5-2 所示的无人机编队协同空战策略模型的分层结构中，决策选择层作为策略选择器负责选择底层策略，初始编队和需要编队重组时选择队形转换策略；在雷达未发现目标阶段应选择目标探测策略对目标进行分布式搜索；搜索过程中要合理分配雷达资源选择雷达开关策略；发现目标后选择机动决策策略追踪目标并在目标进入攻击区时完成对敌打击，追踪目标过程中采取主动干扰策略干扰敌机雷达。

在分层强化学习的框架下，每个层级的策略可以独立学习并优化。顶层模型可以在更广泛的时间尺度上进行决策，而底层元策略模型则聚焦于更短时间尺度的决策。

5.2.1 雷达开关模型

雷达开关策略是无人机空战中雷达的使用策略^[194]，关键在于决定开启雷达进行主动探测的时机，以及关闭雷达以保持隐蔽。这一策略的核心目的是在确保有效地进行探测和信息收集的同时，最大限度地降低被敌方电子侦察系统发现的风险。

为了优化雷达的使用并减少资源消耗，同时降低飞机因雷达活动而被敌方探测到的潜在风险。本文提出了一种雷达开关策略模型，该模型重点分析雷达探测的重叠区域，以此为依据来决定雷达的开启或关闭。如图 5-3 所示，其中 θ 代表雷达的探测半角， r 为雷达探测距离。

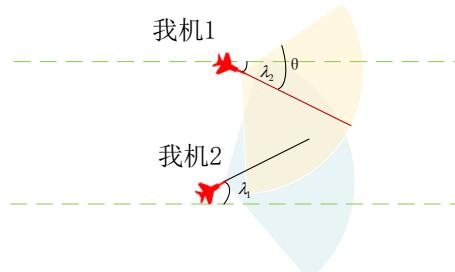


图 5-3 雷达探测重叠区域分析

当两架飞机的距离 d 和航向角 λ_1 和 λ_2 满足如式(5-1)所示的条件时，即视为进入了所谓的“判决区域”：

$$\begin{cases} d \leq r \sin \theta \\ \lambda_1 - \lambda_2 \leq \theta \end{cases} \quad (5-1)$$

在这种情况下，其中一架飞机的雷达会被关闭，以减少电磁信号的重叠和降低位置暴露风险。

5.2.2 主动干扰模型

主动干扰策略^[195]在于调整我机的干扰设备，使其工作频点与敌方雷达的频点相同或接近。这种策略能够有效干扰敌方雷达的正常运作，从而影响其探测和追踪能力。为了在电子战中有效地对敌机实施干扰，本文构建了一个主动干扰模型，目的是提升干扰的精度和效果。模型实施的具体步骤包括：

首先，在干扰操作开始前，需要确定被干扰目标的雷达频点(r_t)，并将此频点作为模型的输入参数。

目标雷达开机频点的观测信息需具备以下两个前提：

- 1) 目标必须位于我机雷达的有效探测范围之内。如果目标超出此范围，我机雷达则无法捕获到目标雷达的信息。
- 2) 目标雷达不应受到外部干扰。如果目标雷达受到电子对抗等干扰，那么我机雷达捕获的频点信息可能会不准确或无法获取。

在获取必要信息后，我机设置对应干扰频点。这一过程可用以下等式表示：

$$r_t = r_j \quad (5-2)$$

其中， r_j 是我机的干扰频点。

模型根据是否成功探测到目标雷达的频点，将产生不同的奖励值。如果未探测到任何目标雷达频点，奖励值设为 0；如果探测到 n 个目标的干扰频点，奖励值则设置为 n 。

该模型的输入参数为敌机雷达的频点，输出为我机实施干扰的雷达频点。

5.2.3 主动探测模型

雷达启动后，首先进入目标搜索阶段。在此阶段，雷达的主要任务是探测潜在目标。一旦探测到目标，雷达系统便进入目标确认过程，即对探测到的目标进行持续监测以确保其真实性^[196]。目标一旦被确认，雷达随即启动对目标的持续跟踪模式，持续监视直至完成对目标的打击任务。这个过程标志着雷达对目标的完整作战周期，如图 5-4 所示，从初始探测到最终的目标打击，展现了雷达系统在空战中的关键作用和战术价值。本文的主动探测策略主要针对发现敌机的过程，之后会开启对目标的持续锁定，即使切换到其他策略，雷达依然可以获取到目标的位置。

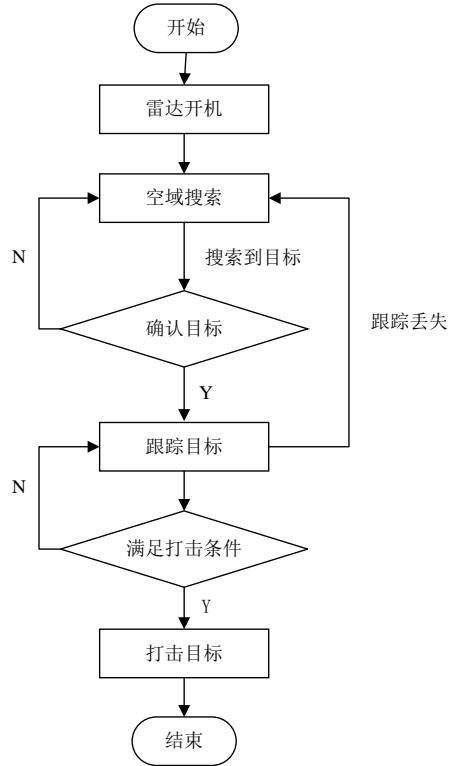


图 5-4 机载雷达作战过程示意图

开启机群分布式搜索主动探测雷达，以达到尽快发现所有敌机的目的。设置固定时间步长 τ_1 ，以在该步达到最大搜索面积。为了进一步确保编队分布搜索，利用人工势场的方法保持我方无人机之间的距离，如式(5-3)所示。通过定义势场函数，当友机间距离过近时，我机之间势场的斥力急剧增大；当友机间距超过指定值时，势场的斥力减少，引力增加。定义 $\rho(q)$ 为我机到其他友机的边界 $Q\mathcal{O}$ 的距离：

$$\rho(q) = \min_{q' \in \partial Q\mathcal{O}} \|q - q'\| \quad (5-3)$$

其中， $\partial Q\mathcal{O}$ 表示配置作战区域的边界。定义 ρ_0 为一个威胁区影响的距离，当我机 q 距离障碍（即友机）距离大于 ρ_0 时，不会排斥 q 。符合上述标准的势函数描述如式(5-4)所示：

$$U_{rep}(q) = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right)^2 & : \rho(q) \leq \rho_0 \\ 0 & : \rho(q) > \rho_0 \end{cases} \quad (5-4)$$

其中， η 是比例系数，斥力为 U_{rep} 的负梯度。当 $\rho(q) \leq \rho_0$ 时，斥力如式所示：

$$F_{rep}(q) = \eta \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q)} \nabla \rho(q) \quad (5-5)$$

当 $Q\mathcal{O}$ 是凸函数时， b 是 $Q\mathcal{O}$ 边界上最接近 q 的点，则 $\rho(q)$ 如式所示：

$$\rho(q) = \|q - b\| \quad (5-6)$$

其中， $\rho(q)$ 的梯度如式所示：

$$\nabla \rho(q) = \frac{q - b}{\|q - b\|} \quad (5-7)$$

其中，单位向量从 b 指向 q 。

5.2.4 队形转换模型

初始时刻我方机群两两以长机-僚机形式编队，长机执行搜索-攻击任务，僚机进行探测干扰任务，掩护长机完成作战任务。若长机被击毁，僚机将接替长机位置完成攻击与目标探测等任务。长机 id 记为 id_l ，僚机 id 记为 id_f 。构建编队列表与全局编队字典 $\{[id_l, id_f] \dots\}$ ，考虑到作战过程中因战损导致编队结构破坏的情况，需要在编队队形破坏后进行编队重组。可以通过判断编队列表，若编队中成员被击毁，响应 id 取反标记。例如，编队 1 长机被击毁，记 $[-id_l, id_f]$ 。若整队成员全部被击毁，将该编队列表移出全局编队字典。

编队重组通过遍历所有编队，根据编队列表中是否存在负值筛选不完整编队，不完整编队数量记作 N ，重组编队数记作 T ，有：

$$T = N \% 2 \quad (5-8)$$

无法重组编队数记为 L ，有：

$$L = N - 2T \quad (5-9)$$

重组的编队根据遍历顺序赋予长机或僚机职能，无法重组的单机单独完成作战任务。

5.2.5 机动决策模型

机动决策模型的训练方法采用第四章基于 E-SAC 算法的无人机空战机动决策的模型，状态空间、动作空间和奖励函数均见 4.3 节。

5.2.6 元策略模型封装

综合以上的元策略模型，包括雷达开关策略、主动干扰策略、主动探测策略、队形转换策略和机动决策策略。上述 5 种元策略构成了空战全流程作战决策模型，元策略封装流程按照如图 5-5 所示。

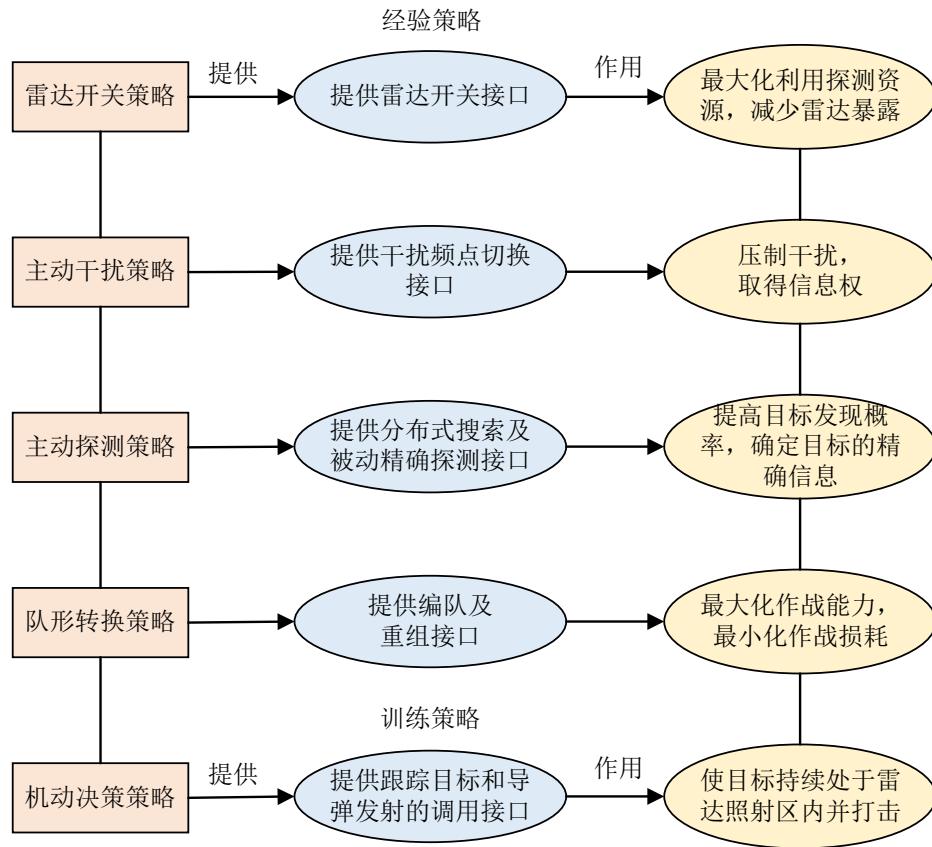


图 5-5 元策略封装结构图

如图 5-5 所示，元策略封装包括雷达开关策略、主动干扰策略、主动探测策略、队形转换策略和机动决策策略 5 种策略。5 种策略的封装内容及功能如下：

雷达开关策略封装了控制编队雷达开关机的接口，实现合理分配探测资源，减少自身暴露的风险。在全流程空战中，该接口应于目标探测时调用。

主动干扰策略封装了干扰频点跟踪接口，能够根据雷达探测到目标的频点进行干扰频点的切换。在全流程空战中，该接口应于发现目标后调用。

主动探测策略封装了基于人工势场的分布式搜索方案以及被动雷达编队搜索方案，通过调用接口可以执行分布式搜索，将观测值作为判据调度友机资源协同搜索，返回探测目标的详细信息。在全流程空战中，该接口应于编队后调用。

队形转化策略封装了编队以及编队重建的接口，调用该接口可实现机群编队，检测编队状态并重组编队的功能。该接口应于初始阶段及编队破坏时调用。

机动决策策略封装了导弹发射接口和追踪目标接口，若雷达发现目标，调用追踪目标接口可实现目标的自动追踪，目标追踪引导到攻击区后调用导弹发射接口。

本节中所提的 5 种基本元策略模型在空战全流程中的逻辑关系如图 5-6 所示。

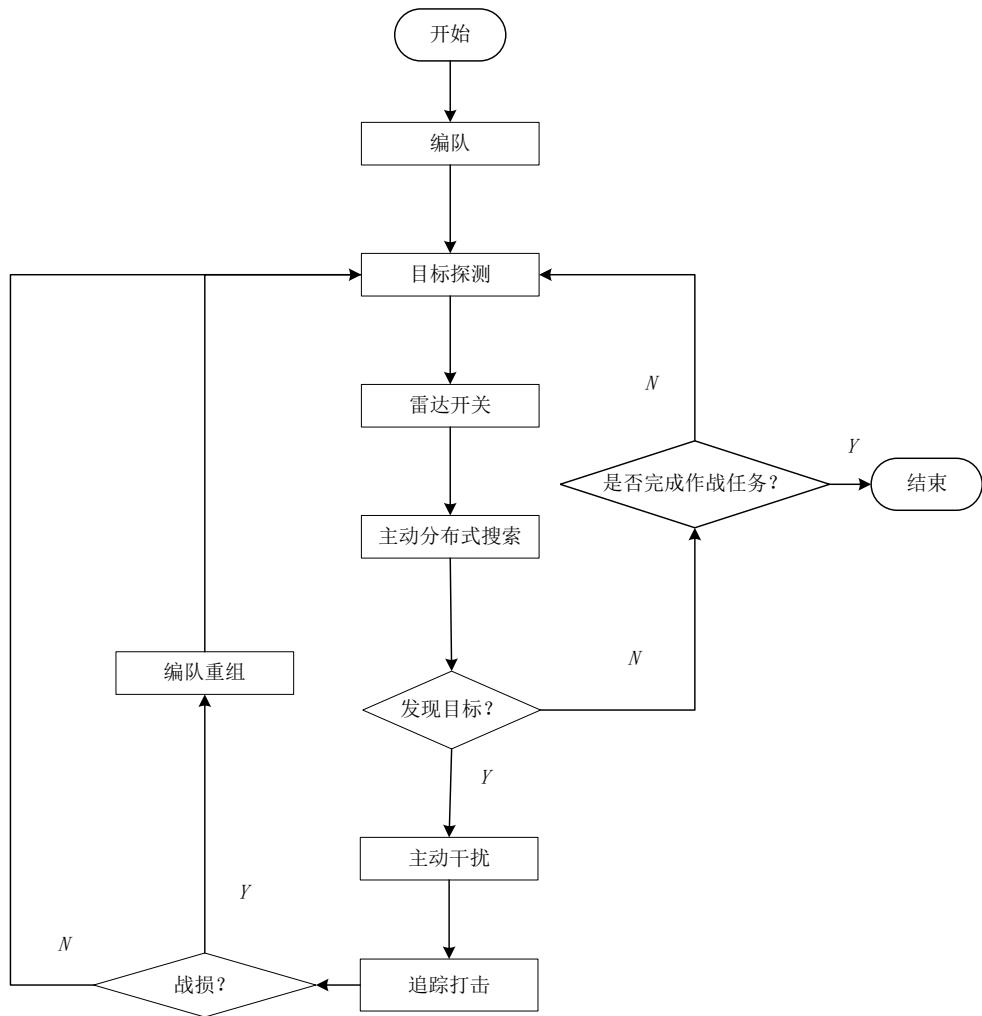


图 5-6 元策略与空战全流程的联系

5.3 改进分层强化学习算法战术决策模型

在分层强化学习算法中，Options 是一个重要的概念。作为一种分层强化学习策略，Options 的核心目的在于引入时间层次和行为层次，以此来应对复杂的决策过程。Options 通过整合单个动作为复杂的操作序列，在处理无人机编队协同空战战术决策时展现出其优势。此外，Options 的引入实现了在时间和空间维度上的抽象化，从而加速了学习过程，并促进了策略在不同任务间的迁移。基于此背景提出了 Option-Critic 架构，该架构在 Options 的基础上进一步发展，提供了一种用于自动化地学习和优化 Options 的机制。

5.3.1 元策略决策网络模型

上层策略选择一个 Option $\omega \in \Omega$ ，Option 包含三个部分：策略 $\pi_\omega(a|s)$ 表示 Option 中的策略，终止条件 β 表示状态 s 结束当前 Option 的概率为 $\beta_\omega(s)$ ，初始集 I_ω 表示 Option 的初始状态集合。

当终止函数返回 0，下一步还会由当前 Option 来控制；当终止函数返回 1 时，该

Option 的任务暂时完成，控制权交还给上层策略。把每个 Option 的行动策略和终止函数都用神经网络进行函数近似，即 $\pi_{\omega,\theta}(a|s)$ 和 $\beta_{\omega,g}(s)$ 。每个 Option 之间做选择的上层策略用 $\pi_\Omega(\omega|s)$ 表示，即在状态 s 时策略选择 Option ω 的概率。在此基础上，定义某状态下选择某个 Option 后产生的总收益，在状态 s 下选择某个 Option 时，采取某行动之后产生的总收益和在使用某 Option 时到达某状态之后产生的总收益。

这些价值函数都是相对于当前策略而言的，除了固定规则的变量之外，其余都服从当前策略运行。Option 内部的更新，其主要包含各个 Option 的策略 $\pi_{\omega,\theta}(a|s)$ 和其终止函数 $\beta_{\omega,g}(s)$ ，如式(5-10)所示。

$$\begin{aligned} \mu_\Omega(s, \omega | s_0, \omega_0) = & \sum_{t=0}^{\infty} \gamma^t P(s_t = s, \omega_t = \omega | s_0, \omega_0) \\ & - \sum_{s', \omega} \mu_\Omega(s', \omega | s_1, \omega_0) \sum_a \frac{\partial \beta_{\omega,\theta}(s')}{\partial \theta} A_\Omega(s', \omega) \end{aligned} \quad (5-10)$$

其中， $\mu_\Omega(s', \omega | s_1, \omega_0)$ 是一个 State-Option 二元组对于轨迹折扣的权重，从 (s_1, ω_0) 开始， $\mu_\Omega(s, \omega | s_1, \omega_0) = \sum_{t=0}^{\infty} \gamma^t P(s_{t+1} = s, \omega_t = \omega | s_1, \omega_0)$ 再使用时序差分方式学习到 Critic Q_U 和 A_Ω ，然后更新 Option 内的参数，从而进行 Option 的学习。对于上层策略，主要使用 $\pi_\Omega(\omega|s)$ 来描述，可认为它是一个 greedy 的策略，选择在所有 Option 中价值函数最大的 Option，即 $\max_\omega \sum_a \pi_{\omega,\theta}(a|s_{t+1}) Q_U(s_{t+1}, \omega, a)$ 。

基于 Option-Critic 的无人机战术决策算法伪代码如表 5-1 所示。

表 5-1 基于 Option-Critic 的算法伪代码

算法: Option-Critic 算法伪代码

- 1: 初始化 $s \leftarrow s_0$
 - 2: 根据选项上的 ε -软策略 $\pi_\Omega(s)$ 选择选项 ω
 - 3: For episode = 1, M do
 - 4: 根据选项 ω 的策略 $\pi_{\omega,\theta}(a|s)$ 选择动作 a
 - 5: 在状态 s 中采取动作 a ，观察新状态 s' 和奖励 r
 - 6: # 选项评估
 - 7: $\delta \leftarrow r - Q_U(s, \omega, a)$
 - 8: If s' 不是终止状态 Then
 - 9: $\delta \leftarrow \delta + \gamma(1 - \beta_{\omega,g}(s')) Q_\Omega(s', \omega) + \gamma \beta_{\omega,g}(s') \max_{\bar{\omega}} Q_\Omega(s', \bar{\omega})$
 - 10: End If
 - 11: $Q_U(s, \omega, a) \leftarrow Q_U(s, \omega, a) + \alpha \delta$
 - 12: # 选项改进
 - 13: $\theta \leftarrow \theta + \alpha_\theta \frac{\partial \log \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s, \omega, a)$
 - 14: $\vartheta \leftarrow \vartheta - \alpha_\vartheta \frac{\partial \beta_{\omega,g}(s')}{\partial \vartheta} (Q_\Omega(s', \omega) - V_\Omega(s'))$
 - 15: If $\beta_{\omega,g}$ 在 s' 终止 then
-

算法: Option-Critic 算法伪代码

```

16: 根据  $\epsilon$ -soft( $\pi_\Omega(s')$ ) 选择新的  $\omega$ 
17:  $s \leftarrow s'$ 
18: 直到  $s'$  是终止状态
19: End If

```

其结构如图 5-7 所示。

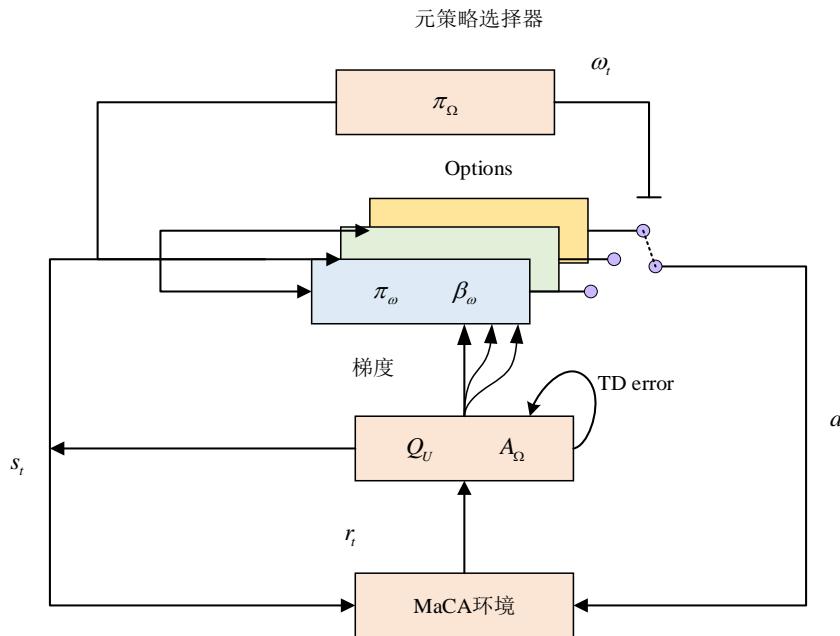


图 5-7 Option-Critic 结构

改进 Options 模型构建结构与一般 Option-Critic 结构不同，在训练时采取元策略模型，通过完成封装的元策略模型，输入环境状态返回动作。即放弃 Option-Critic 自动训练策略，这是因为采取 Option-Critic 方法训练的元策略无法将动作序列具体化为符合人类逻辑与经验的策略，且容易过拟合到某个动作，而非序列动作。改进后的方法只通过 Actor-Critic 网络训练得到终止函数，给出元策略在不同状态下何时完成元策略的执行。Critic 网络根据状态及奖励更新 Options 价值函数 Q 值以及 Options 间的优势函数，梯度反向传递 Actor 网络完成参数更新并给出终止函数。其构建流程如图 5-8 所示。

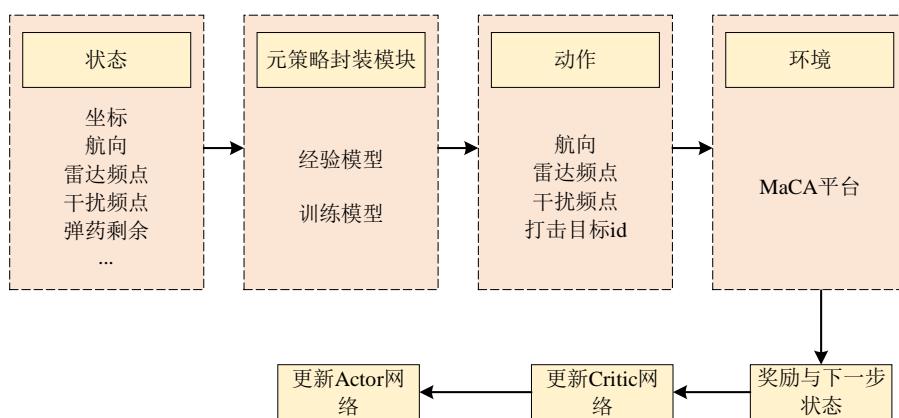


图 5-8 Options 模型构建及训练流程

为统一 Options 网络的状态输入，将全部原始观测值统一输入到各个网络，由各个网络对各自所需的观测值进行预处理。各个 Option 网络的输入和输出如下：

- (1) 针对雷达开关策略，输入为我机的坐标、航向、雷达开关状态，输出为本机的雷达开机频点。
- (2) 针对主动干扰策略，输入为敌机的开机频点，输出为我机的干扰开机频点。
- (3) 针对主动探测策略，输入为我机群的位置坐标、航向，输出为我机的航向。
- (4) 针对队形转换策略，输入为我机群的位置坐标、航向，我机的存活状态，输出为我机的航向。
- (5) 针对机动决策策略，输入为我机的位置坐标、航向、探测到的敌机位置坐标，输出为我机的航向、是否发射、发射情况下的打击目标 id 。

在上述策略中，探测到的敌机坐标来源主要分友机探测来源与我机探测来源两种。

Options 通过终止条件 $\beta(s)$ 以判断是否终止。策略选择器采用贪婪策略，则得出相应的一步离线策略更新目标 $g_t^{(1)}$ 如式所示：

$$\begin{aligned} g_t^{(1)} = & r_{t+1} + \gamma \left((1 - \beta_{\omega_t, g}(s_{t+1})) \sum_a \pi_{\omega_t, \theta}(a | s_{t+1}) Q_U(s_{t+1}, \omega_t, a) \right. \\ & \left. + \beta_{\omega_t, g}(s_{t+1}) \max_{\omega} \sum_a \pi_{\omega, \theta}(a | s_{t+1}) Q_U(s_{t+1}, \omega, a) \right) \end{aligned} \quad (5-11)$$

全流程作战策略由五部分元策略构成，一个训练策略：机动决策策略；四个固定策略：雷达开关、主动干扰、主动探测和队形转换策略。对于训练策略基于 E-SAC 算法框架分别构建执行和评估神经网络。具体的状态空间、动作空间和奖励函数设置如下。

(1) 状态空间

无人机机动决策模型的状态空间用于描述空中战斗的全部情况，该情况在任何时候都可以通过无人机状态来确定，无人机状态包含位置矢量 $\mathbf{R} = [x, y]$ 、速度 v 、航向角 φ 和距离 d 中的相对信息。为了统一每个变量的范围并提高网络学习的效率，每个状态变量都被归一化为 $[0, 1]$ 。状态空间定义为一个 6 元组 $S = [x, y, v, \varphi, d, q]$ 。

(2) 动作空间

无人机运动方程表明，在有效积分步长 dt 内设置 dv 、 $d\varphi$ ，无人机可以在空间中实现一系列的机动过程。因此，可以得到无人机飞行动作空间，即： $A = [dv, d\varphi]$ 。

(3) 奖励函数

为了确保空战模型的有效性，对于每个元策略，需要设计合适的奖励机制以激励最佳行为，并设定恰当的终止函数以在适当的时机结束当前策略，从而提高整体模型的性能。5 种元策略的奖励设置方案与顶层奖励函数如表 5-2 所示：

表 5-2 奖励设置

| 事件 | 奖励或惩罚 | 说明 |
|-------|-------|------------|
| 雷达开关 | +20 | 完成雷达开关切换动作 |
| 主动干扰 | +100 | 我机成功设定干扰频点 |
| 主动探测 | +10 | 我机成功发现目标 |
| 队形转换 | +30 | 我机完成队形转换 |
| 机动决策 | +100 | 我机追踪并打击目标 |
| 摧毁敌机 | +500 | 我机成功击杀对手 |
| 被敌机摧毁 | -500 | 我机被敌机击杀 |
| 超界 | -10 | 我机超出战场边界 |

5.3.2 顶层策略网络模型

顶层策略网络主要负责根据当前环境和状态选择合适的元策略，其中，策略选择器是实现这一目标的具体机制，本章策略选择器的模型采用传统的贪心策略模型，即在每个状态下选择预期回报最大的元策略。具体而言，元策略模型包括雷达开关、主动干扰、队形转换、主动探测和机动决策模型，策略选择器在计算期望回报时需要考虑相应终止函数选择概率。策略选择器的模型构建如图 5-9 所示。

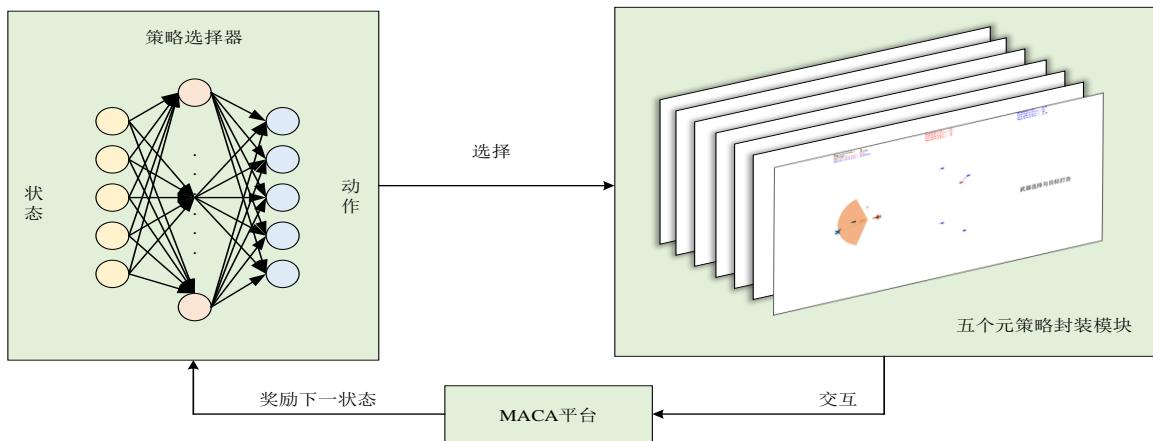


图 5-9 策略选择器模型构建

策略选择器将五个元策略作为五个动作值，以贪心策略选择当前环境状态的元策略，获取的奖励与下一时刻状态。重复以上步骤，训练策略选择器，直到能够选择合适的策略动作。

顶层策略需要以我方机群收到的总的外界奖励为指导，元策略由经验策略模型和训练策略模型组成，顶层策略和元策略与训练环境进行交互以训练策略选择器。整个作战过程策略选择器在不同状态下应具备：

- (1) 先敌发现

初始状态选择队形转换策略进行编队；随后采取目标探测策略展开搜索；搜索过程中执行雷达开关策略，分布式搜索执行该策略全部开启雷达，其余情况根据雷达判决区而定；能够在作战时采取主动干扰策略将干扰频点设置成观测到的敌方雷达频点。

(2) 先敌打击

发现目标后为避免目标丢失，应执行机动决策策略，锁定目标位置并引导到攻击区进行目标打击。

顶层策略选择器的训练流程如下：

- 1) 初始化：设置元策略训练网络和元策略选择器的网络参数；
- 2) 生成底层动作：根据元策略的 Actor 网络和固定策略的动作输出函数生成动作值；
- 3) 生成元策略选择动作：生成服从均匀分布的随机动作；
- 4) 与环境进行交互：将得到的动作集合在仿真环境中执行，得到奖励集合，下一状态集合和任务结束标志集合；
- 5) 保存经验：将经验分别存储到对应元策略的经验池中；
- 6) 测试：达标停止训练，否则循环步骤 1~5。

5.4 仿真结果与分析

5.4.1 实验环境设定

设置双方无人机数量为 4，每架无人机配备导弹各 8 枚，地图尺寸设置为 12000×12000 米，算法采用 MaCA 环境中的 fix_rule_no_att 黑盒算法；我机采用本文的改进分层强化学习算法。其具体参数如表 5-3 所示：

表 5-3 基于分层强化学习的编队协同智能空战战术决策模型仿真参数表

| 参数名称 | 参数大小 |
|----------------|--------------------|
| Actor 策略网络学习率 | 3×10^{-4} |
| Critic 策略网络学习率 | 3×10^{-3} |
| 温度参数 | 3×10^{-4} |
| 网络隐含层个数 | 2 |
| 网络隐含层神经元个数 | 512 |
| 学习速率 | 0.0003 |
| 累积回报折扣因子 | 0.99 |
| 目标熵 H_0 | -3 |
| 熵的正则化系数初始值 | 1 |
| 目标网络更新间隔步数 | 300 |
| 经验池容量 | 100,000 |
| 经验池提取样本数量 | 200 |
| 激活函数 | ReLU |
| 优化器 Adam 软更新系数 | 0.005 |

作战结果判定标准如下。

- 1) 完胜: 一方被全部击毁;
- 2) 完败: 失败;
- 3) 胜利: 双方导弹存量为 0, 存活数量多的获胜或达到最大 step, 剩余作战单元数量多的获胜;
- 4) 平局: 双方作战单元全都被击毁或双方导弹存量为 0 且双方存活作战单元数量相等或当达到最大 step 且双方存活作战单数量相等。

5.4.2 全流程仿真结果与分析

(1) 雷达开关策略有效性验证

雷达开关策略可以帮助我机降低暴露于敌机探测区域的风险。为了验证雷达开关策略的有效性, 本章首先不考虑雷达开关策略, 保持雷达常开的状态, 统计我机被敌机被动探测到的次数; 作为对照仿真开启雷达开关策略对全流程的空战进行模拟, 记录我机被敌机被动探测到的次数。一共进行 2000 个回合的空战对抗实验, 空战结果如表 5-4 所示。

表 5-4 雷达开关策略有效性验证

| | 关闭雷达开关策略 | 使用雷达开关策略 |
|------------|----------|----------|
| 平均被发现次数(次) | 21.5 | 15.7 |

由实验结果可知, 关闭雷达开关策略其平均每局被动探测 21.5 次, 而开启雷达开关策略其平均每局被动探测 15.7 次, 两者的结果对比可知, 雷达策略可以使得被敌机发现

的次数要平均少 5.8 次，可见使用雷达开关策略可以显著减少我机被敌机被动探测到的次数，这减小了我方飞机因被动探测而暴露位置的风险。该实验验证了分层强化学习空战智能决策中雷达开关策略的有效性。

(2) 主动干扰策略有效性验证

主动干扰的目的是通过发射干扰信号来影响敌机的雷达性能，提升己方攻击优势。为了验证主动干扰策略的有效性，采用基于改进分层强化学习算法对全流程空战进行模拟，首先关闭主动干扰策略，记录我机被敌机探测到的次数，然后开启主动干扰策略对全流程的空战进行模拟，记录我机被敌机探测到的次数。一共进行 2000 个回合的空战对抗实验，空战结果如表 5-5 所示。

表 5-5 雷达开关策略有效性验证

| | 关闭主动干扰策略 | 使用主动干扰策略 |
|-----------------|----------|----------|
| 敌机平均探测到的我机数量(架) | 2.3 | 1.1 |

由实验结果可知，该实验验证了分层强化学习空战智能决策中雷达开关策略的有效性。开启主动干扰策略后，敌机探测到我机的次数有所减少，从平均探测到 2.3 架变为平均探测到 1.1 架，这表明主动干扰策略能够有效地降低了敌机的探测能力。

(3) 主动探测策略有效性验证

目标探测策略中的分布式搜索采用了人工势场中的斥力场，能够维持友机间距，避免探索重复区域。我机分布式搜索策略如图 5-10 所示，当友机相互靠近时会采取相互远离的动作。

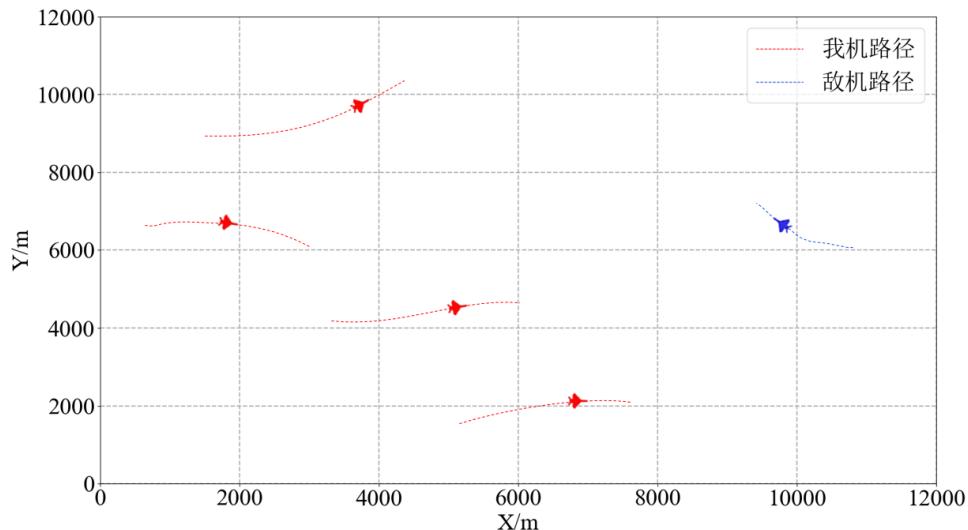


图 5-10 我机分布式搜索策略

(4) 队形转换策略有效性验证

空战场景初始化设置为长机-僚机编队形式，如图 5-11 所示。

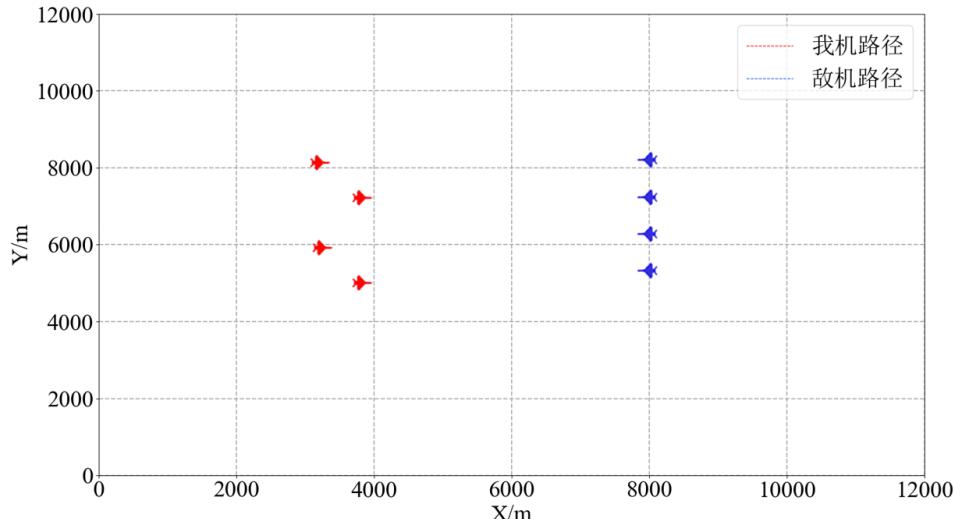


图 5-11 初始编队示意图

队形转换策略在作战过程中能够在我机队形被破坏的情况下完成编队重组，重组后以编队形式对目标进行压制干扰，有效避免了单独作战时的风险。为了验证队形转换策略的有效性，在进行回合数为 2000 的 4v4 空战全流程仿真时关闭队形转换策略，即初始的长机-僚机编队中有飞机战损，则编队中的无人机以单机的方式对探测目标并对其进行攻击，统计实验中 50 步内击败敌机的数量；然后，同样进行回合数为 2000 的 4v4 空战全流程仿真，此时启用队形转换策略，同样统计 50 步内击败敌机的数量。具体实验结果如表 5-6 所示。

表 5-6 队形转换策略有效性验证

| | 关闭队形转换策略 | 使用队形转换策略 |
|----------------|----------|----------|
| 50 步内打击敌机数量(架) | 0.5 | 1.1 |

由实验结果可知，在关闭队形转换策略时 2000 个回合中 50 步内平均打击敌机的数量为 0.5 架，但开启队形转换策略后其数量上升至 1.1 架，可以验证分层决策下队形转换策略的有效性。同时，保持长-僚机编队的策略能够有效地确保优先取得信息优势，实现对敌机的压制干扰，在此策略下，我方能够在无损毁的情况下歼灭敌机，并从初始阶段便掌握数量优势。

(5) 无人机编队全流程空战仿真

首先采用第四章所提的 E-SAC 算法对 4v4 场景下的机动决策元策略进行训练，奖励函数设置如表 5-2 所示，其训练奖励曲线图如图 5-12 所示。

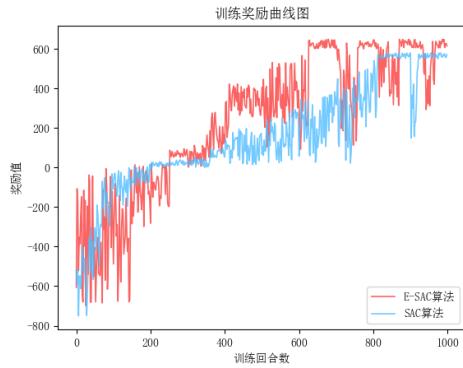


图 5-12 训练奖励曲线图

训练结束后与敌机对抗的仿真过程如图 5-13 所示。

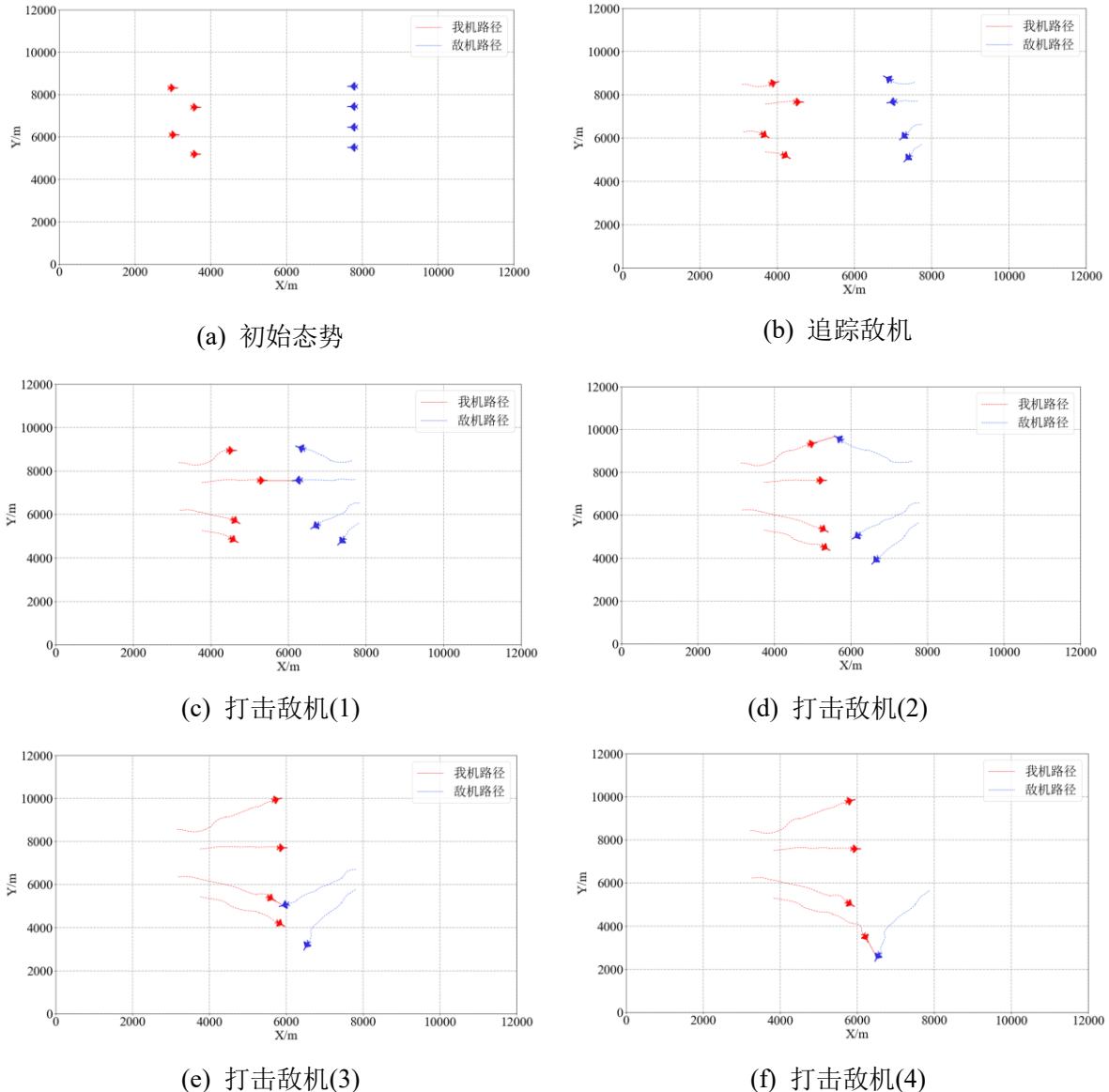


图 5-13 无人机编队四对四机动决策训练仿真过程

然后将训练完成的策略与敌机进行对抗仿真，采用本章所提的改进 Option-Critic 算法对 4v4 场景下的全流程进行仿真，其具体的训练奖励曲线如图 5-14 所示。

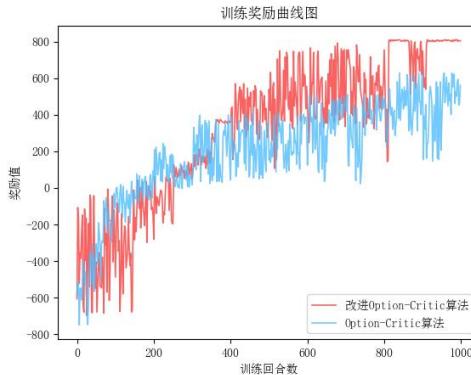


图 5-14 训练曲线图

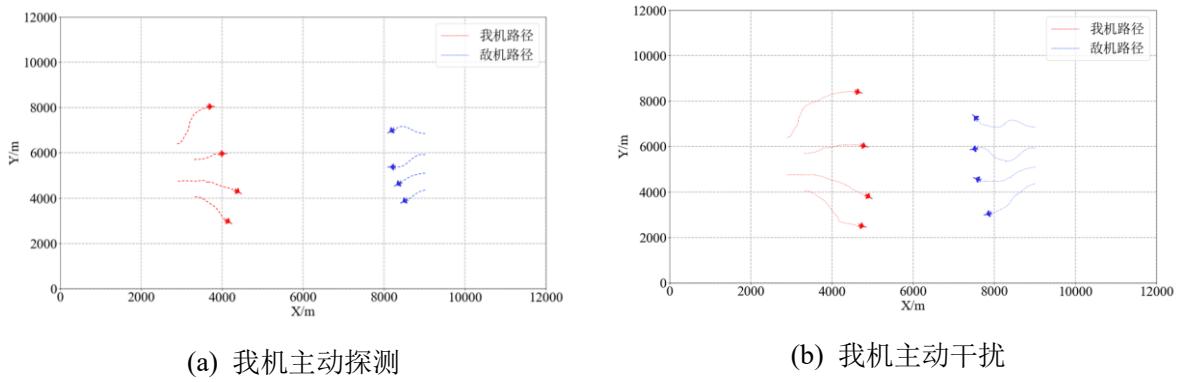
具体的训练结果数据如表 5-7 所示。

表 5-7 分层强化学习训练结果

| | Option-Critic | 改进 Option-Critic |
|---------|---------------|------------------|
| 最大奖励 | 635 | 808 |
| 奖励收敛回合数 | 627 | 811 |

从训练过程中的奖励曲线图和胜率图可以看出，Option-Critic 算法在训练过程中会出现陷入局部最优的情况，且其收敛效果较差，震荡幅度明显；本文所提的改进 Option-Critic 算法虽然在训练的过程中需要更多的回合数，但是其训练效果较好，可以获得 808 的奖励值，因此本文所提的算法具有有效性。

利用本文所提的改进分层强化学习对无人机编队协同决策仿真，仿真过程如图 5-15 所示。



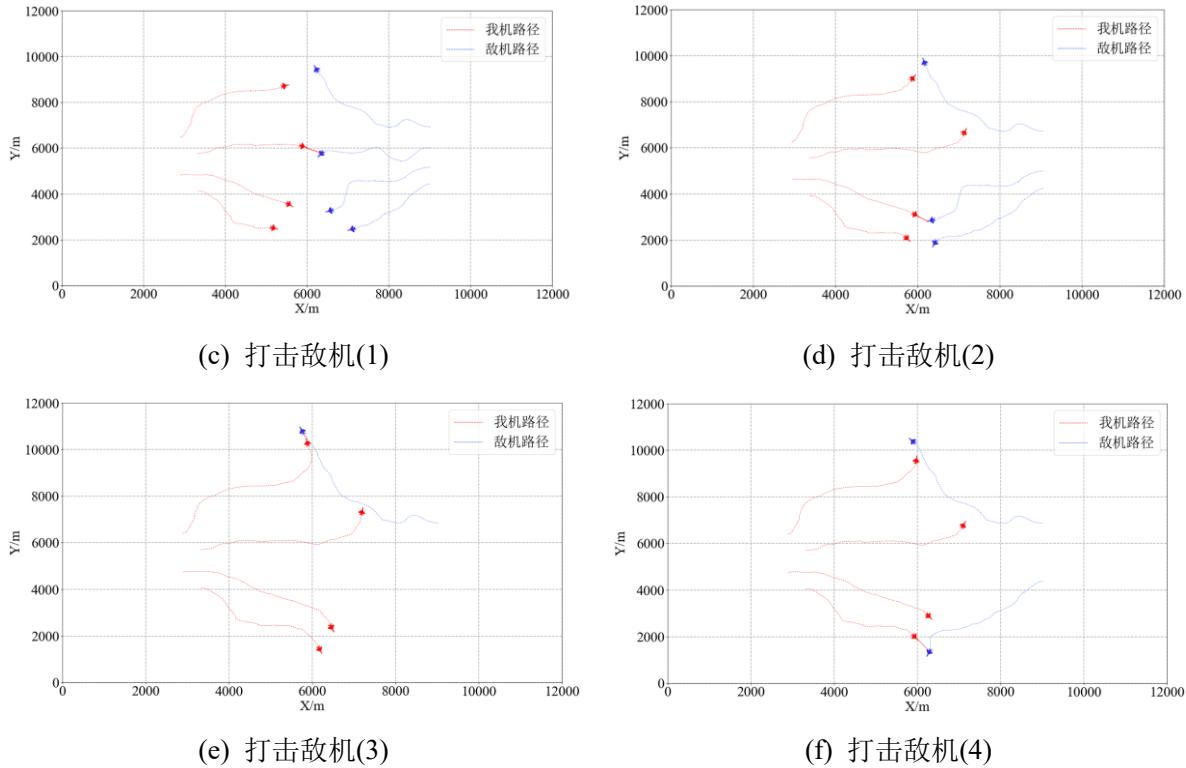


图 5-15 无人机编队四对四机动决策训练仿真过程

由仿真结果表明，我机能够在空战中进行有效的战术决策。在空战 4v4 场景中，我方无人机首先采用了主动探测策略对作战环境进行分布式搜索，在我机探测到敌机后选择了主动干扰策略对敌机进行干扰，随后采用了机动决策策略引导我机朝着敌机航行，并在到达导弹攻击区后对敌机进行攻击，最终将敌机全部歼灭，验证了本文所设计的无人机编队空战战术决策方法的有效性。

为了进一步分析本文所设计的基于编队协同分层强化学习空战决策模型的可靠性，本文引入了平均战损的概念，即平均每局对决的无人机损失数量，本次仿真对 50 场作战结果进行统计，统计作战胜率和平均战损，如表 5-8 所示。

表 5-8 分层强化学习仿真结果

| | 蓝方(敌机) | 红方(我机) |
|------|--------|--------|
| 胜率 | 0% | 100% |
| 平均战损 | 5.7 | 1.4 |

由实验结果可知，在 50 场空战对抗中，应用本文的编队协同智能空战战术决策方法均可以实现 100% 的胜率，其战损仅为 1.4 架战机，蓝方飞机胜率为 0 且平均每局损失高达 5.7 架战机，可见本论文提出的编队协同智能空战战术决策算法显著提升了无人机编队的作战能力，具备卓越的作战效能。

5.5 本章小结

本章针对无人机编队协同空战战术决策问题，提出了一种基于改进 Option-Critic 算法的编队协同策略模型。首先设计基于分层强化学习算法的无人机编队协同空战战术决策的元策略模型；其次，基于元策略选择器对分层强化学习战术决策模型进行改进；最后，设置仿真实验验证本文所提的分层强化学习算法在空战战术决策中的有效性，并将改进前后的算法进行对比。

第6章 总结与展望

6.1 论文总结

随着无人机技术的飞速发展和现代战争形态的不断演变，空中作战环境变得日益复杂和难以预测。在这样的背景下，提高无人机编队在战术决策上的自主性和效率，尤其是其智能化作战能力，成为了军事科技进步的重要方向。论文基于分层强化学习算法，针对无人机编队协同战术决策问题，从无人机编队航路规划、机动决策和编队协同战术决策三个方面展开研究。论文的主要研究工作如下：

(1) 针对多无人机航路规划问题，提出了一种改进的灰狼算法多无人机协同航路规划模型，提高了多无人机航路规划的能力和效率。

首先，构建了多无人机协同航路规划的模型；其次，针对灰狼算法收敛速度慢和容易陷入局部最优的问题，提出了一种利用非线性函数优化收敛系数的方法，并根据不同灰狼的适应度设计权重比例因子；最后，将传统灰狼优化算法与改进后的灰狼算法进行仿真对比，同时分析了算法的有效性。

(2) 针对无人机空战机动决策问题，将处理过程分为导弹攻击区解算和无人机机动决策两个步骤。根据空战环境中对无人机机动实时性和准确性要求高的特点，提出了一种基于 E-SAC(Expert-Soft ActorCritic, E-SAC)算法的机动决策模型，以增强智能体的探索效率和优化学习过程。

首先，针对黄金分割法解算实时性差和传统 BP 神经网络拟合精度差的问题，提出一种基于调优反向传播神经网络(Tuning of Backpropagation Neural Networks, TBPNN)算法的导弹攻击区的拟合方法；其次，根据解算出的攻击区设计基于深度强化学习算法的无人机机动决策的马尔可夫决策模型；然后，对 SAC(Soft Actor Critic, SAC)算法进行改进，提出了基于 E-SAC 算法的无人机自主机动决策方法，有效解决了 SAC 算法在处理无人机空战机动决策问题时的收敛速度慢和易陷入局部最优的不足；最后，仿真结果表明，论文提出方法在解决无人机机动决策问题上具有有效性和可行性，且在收敛速度和导引实时性方面优于传统 SAC 算法。

(3) 针对无人机编队协同空战全流程决策问题，基于分层强化学习算法，提出并构建了简易策略的经验模型与复杂策略的训练模型，指引无人机编队在空战中完成战术决策。

首先，根据无人机编队空战战术决策任务需求，建立了基于专家经验的雷达开关、主动干扰、主动探测和队形转换模型；其次，基于分层强化学习算法框架，将专家经验模型与基于 E-SAC 算法的机动决策模型相结合建立元策略，并采用 Actor-Critic 网络训

练习终止函数，解决了无人机编队空战战术决策中维数爆炸和训练不收敛的问题；最后，通过仿真验证了论文所提方法对解决无人机编队协同战术决策问题的可行性和优越性，且以全胜的战绩击败敌机策略。

6.2 后续展望

论文以分层强化学习方法作为核心研究理论，针对无人机编队协同智能空战战术决策问题，按照不同的作战任务进行划分，具体从雷达开关、主动干扰、主动探测、机动决策和战术防御这五个任务进行研究和仿真，从仿真结果可以看出，论文达到了预期的研究目标。结合论文研究内容和真实空战对无人机的需求，论文还可以在以下几个方面开展进一步的研究：

(1) 论文中对于与雷达使用相关的元策略的研究未涉及被动探测策略。

考虑到被动探测作为一种有效降低敌机发现我方飞行器概率的策略，在实际空战中扮演着至关重要的角色。因此，未来的研究应当考虑纳入被动探测策略，这不仅能增强无人机在敌对环境中的生存能力，还能提升其战术上的灵活性和适应性。

(2) 在论文中探讨无人机空战全流程决策对抗仿真的过程中，鉴于多智能体系统固有的复杂性，未来的工作将着重于探索更高效的协同协议和通信机制。

有效的协同协议对于确保多无人机系统在复杂空战环境中的高效协作至关重要。这包括但不限于，发展先进的任务分配算法、协同航路规划和战术决策同步机制。同时，鉴于通信延迟和中断可能对协同操作产生显著影响，改进通信机制，比如实现低延迟、高可靠性的数据传输，将是提高整体系统性能的关键。

参考文献

- [1] 祁圣君,井立,王亚龙.无人机系统及发展趋势综述[J].飞航导弹,2018(04):17-21.
- [2] 苗壮,孙盛智,段炼,别亚星.军用无人机关键技术发展应用及主要作战样式研究[J].飞航导弹,2020(09):52-56.
- [3] 段海滨,申燕凯,赵彦杰,范彦铭,王寅,牛轶峰,魏晨,罗德林.2019 年无人机热点回眸[J].科技导报,2020,38(01):170-187.
- [4] 魏齐. 里程碑!中国大型无人机编队飞行[N]. 环球时报,2023-06-30(008).
- [5] Sun W, Jiang N, Krishnamurthy A, et al. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches[C]. Conference on learning theory. PMLR, 2019: 2898-2933.
- [6] Lee C S, Wang M H, Chaslot G, et al. The computational intelligence of MoGo revealed in Taiwan's computer Go tournaments[J]. IEEE Transactions on Computational Intelligence and AI in games, 2009, 1(1): 73-89.
- [7] Lu F, Yamamoto K, Nomura L H, et al. Fighting game artificial intelligence competition platform[C]. 2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE). IEEE, 2013: 320-323.
- [8] Li Y, Han W, Wang Y. Deep reinforcement learning with application to air confrontation intelligent decision-making of manned/unmanned aerial vehicle cooperative system[J]. IEEE Access, 2020, 8: 67887-67898.
- [9] Yue L, Yang R, Zhang Y, et al. Deep reinforcement learning for UAV intelligent mission planning[J]. Complexity, 2022, 2022.
- [10] Wu Z S, Fu W P. A review of path planning method for mobile robot[J]. Advanced Materials Research, 2014, 1030: 1588-1591.
- [11] Zhu Z, Yin Y, Lyu H. Automatic collision avoidance algorithm based on route-plan-guided artificial potential field method[J]. Ocean Engineering, 2023, 271: 113737.
- [12] Bertsimas D, Tsitsiklis J. Simulated annealing[J]. Statistical science, 1993, 8(1): 10-15.
- [13] Choi K, Jang D H, Kang S I, et al. Hybrid algorithm combing genetic algorithm with evolution strategy for antenna design[J]. IEEE transactions on Magnetics, 2015, 52(3): 1-4.
- [14] Yu X, Li C, Zhou J F. A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios[J]. Knowledge-Based Systems, 2020, 204: 106209.

- [15] Alkhateeb F, Abed-algundi B H, Al-rousan M H. Discrete hybrid cuckoo search and simulated annealing algorithm for solving the job shop scheduling problem[J]. *The Journal of Supercomputing*, 2022: 1-28.
- [16] Khan M S A, Santhosh R. Task scheduling in cloud computing using hybrid optimization algorithm[J]. *Soft Computing*, 2022, 26(23): 13069-13079.
- [17] Praveen Kumar R, Raj J S, Smys S. Performance analysis of hybrid optimization algorithm for virtual head selection in wireless sensor networks[J]. *Wireless Personal Communications*, 2021: 1-16.
- [18] Chandrasekhar U, Khare N. An intelligent tutoring system for new student model using fuzzy soft set-based hybrid optimization algorithm[J]. *Soft Computing*, 2021, 25: 14979-14992.
- [19] Alkhateeb F, Abed-algundi B H, Al-rousan M H. Discrete hybrid cuckoo search and simulated annealing algorithm for solving the job shop scheduling problem[J]. *The Journal of Supercomputing*, 2022: 1-28.
- [20] Yan F. Gauss interference ant colony algorithm-based optimization of UAV mission planning[J]. *The Journal of Supercomputing*, 2020, 76(2): 1170-1179.
- [21] Lee J H, Song J Y, Kim D W, et al. Particle swarm optimization algorithm with intelligent particle number control for optimal design of electric machines[J]. *IEEE Transactions on Industrial Electronics*, 2017, 65(2): 1791-1798.
- [22] Li H, Baoyin H. Optimization of multiple debris removal missions using an evolving elitist club algorithm[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2019, 56(1): 773-784.
- [23] Cheng R, Song Y, Chen D, et al. Intelligent positioning approach for high speed trains based on ant colony optimization and machine learning algorithms[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2018, 20(10): 3737-3746.
- [24] Wang Y, Zhang Y, Xu D, et al. Improved whale optimization-based parameter identification algorithm for dynamic deformation of large ships[J]. *Ocean Engineering*, 2022, 245: 110392.
- [25] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. *Advances in engineering software*, 2014, 69: 46-61.
- [26] Daniel E, Anitha J, Gnanaraj J. Optimum laplacian wavelet mask based medical image using hybrid cuckoo search–grey wolf optimization algorithm[J]. *Knowledge-Based Systems*, 2017, 131: 58-69.

- [27] Guha D, Roy P K, Banerjee S. Load frequency control of interconnected power system using grey wolf optimization[J]. Swarm and Evolutionary computation, 2016, 27: 97-115.
- [28] Emary E, Zawbaa H M, Hassanien A E. Binary grey wolf optimization approaches for feature selection[J]. Neurocomputing, 2016, 172: 371-381.
- [29] Radmanesh M, Kumar M, Sarim M. Grey wolf optimization based sense and avoid algorithm in a Bayesian framework for multiple UAV path planning in an uncertain environment[J]. Aerospace Science and Technology, 2018, 77: 168-179.
- [30] Long W, Jiao J, Liang X, et al. Inspired grey wolf optimizer for solving large-scale function optimization problems[J]. Applied Mathematical Modelling, 2018, 60: 112-126.
- [31] Kumar V, Kumar D. An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems[J]. Advances in Engineering Software, 2017, 112: 231-254.
- [32] Long W, Jiao J, Liang X, et al. An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization[J]. Engineering Applications of Artificial Intelligence, 2018, 68: 63-80.
- [33] Zhang X, Wang X, Kang Q, et al. Hybrid grey wolf optimizer with artificial bee colony and its application to clustering optimization[J]. Acta Electronica Sinica, 2018, 46(10): 2430.
- [34] Lu F, Feng W, Gao M, et al. The fourth-party logistics routing problem using ant colony system-improved grey wolf optimization[J]. Journal of Advanced Transportation, 2020, 2020: 1-15.
- [35] Wu Y, Sun X, Dai B, et al. A transformer fault diagnosis method based on hybrid improved grey wolf optimization and least squares-support vector machine[J]. IET generation, transmission & distribution, 2022, 16(10): 1950-1963.
- [36] Khan I, Basuli K, Maiti M K. Multi-objective covering salesman problem: a decomposition approach using grey wolf optimization[J]. Knowledge and Information Systems, 2023, 65(1): 281-339.
- [37] Wang Y L, Wang T, Yao C. Gray wolf optimization algorithm based on Kent mapping and adaptive weight[J]. Application Research of Computers, 2020, 37(2): 37-40.
- [38] Ou Y, Yin P, Mo L. An Improved Grey Wolf Optimizer and Its Application in Robot Path Planning[J]. Biomimetics, 2023, 8(1): 84.
- [39] Liu Y, Jiang Y, Zhang X, et al. An improved grey wolf optimizer algorithm for identification and location of gas emission[J]. Journal of Loss Prevention in the Process

Industries, 2023, 82: 105003.

- [40] 张超然,吕余海.基于置信度神经网络的机载武器攻击区拟合算法[J].弹箭与制导学报,2021,41(04):120-124.
- [41] Austin F, Carbone G, Falco M, et al. Game theory for automated maneuvering during air-to-air combat[J]. Journal of Guidance, Control, and Dynamics, 1990, 13(6): 1143-1149.
- [42] Park H, Lee B Y, Tahk M J, et al. Differential game based air combat maneuver generation using scoring function matrix[J]. International Journal of Aeronautical and Space Sciences, 2016, 17(2): 204-213.
- [43] Virtanen K, Karelähti J, Raivio T. Modeling air combat by a moving horizon influence diagram game[J]. Journal of guidance, control, and dynamics, 2006, 29(5): 1080-1091.
- [44] Grimm W, Well K H. Modelling air combat as differential game recent approaches and future requirements[C].Differential Games—Developments in Modelling and Computation: Proceedings of the Fourth International Symposium on Differential Games and Applications August 9–10, 1990, Helsinki University of Technology, Finland. Springer Berlin Heidelberg, 1991: 1-13.
- [45] Lee B Y, Han S, Park H J, et al. One-versus-one air-to-air combat maneuver generation based on the differential game[C].Proceedings of the 2016 Congress of the International Council of the Aeronautical Sciences. 2016: 1-7.
- [46] McGrew J S, How J P, Williams B, et al. Air-combat strategy using approximate dynamic programming[J]. Journal of guidance, control, and dynamics, 2010, 33(5): 1641-1654.
- [47] 袁广进. 多无人机协同搜索算法研究与实现[D]. 南京邮电大学, 2021.
- [48] Zhang X, Liu G, Yang C, et al. Research on air confrontation maneuver decision-making method based on reinforcement learning[J]. Electronics, 2018, 7(11): 279.
- [49] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540): 529-533.
- [50] Fang J, Zhang L, Fang W and Xu T, "Approximate dynamic programming for CGF air combat maneuvering decision," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2016, pp. 1386-1390, doi: 10.1109/CompComm.2016.7924931.
- [51] Wang Y, Long Q, Wu M, Chen Y and Tuhiur R, "Research on Maneuvering Control Algorithm of Short-Range UAV Air Combat Based on Deep Reinforcement Learning," 2023 2nd International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM), Jiuzhaigou, China, 2023, pp. 83-90, doi:

- 10.1109/MLCCIM60412.2023.00018.
- [52] Isci H, Koyuncu E. Reinforcement Learning Based Autonomous Air Combat with Energy Budgets[C].AIAA SCITECH 2022 Forum. 2022: 0786.
- [53] Li Y, Lyu Y, Shi J, et al. Autonomous Maneuver Decision of Air Combat Based on Simulated Operation Command and FRV-DDPG Algorithm[J]. Aerospace, 2022, 9(11): 658.
- [54] Li L, Zhou Z, Chai J, Liu Z, Zhu Y and Yi J, "Learning Continuous 3-DoF Air-to-Air Close-in Combat Strategy using Proximal Policy Optimization," 2022 IEEE Conference on Games (CoG) , Beijing , China , 2022 , pp . 616 – 619 , doi : 10. 1109 / CoG 51982.2022. 9893690. Office of the Assistant Secretary of Defense for Acquisition Washington United States. Unmanned Systems Integrated Roadmap 2017-2042 [R] . 2018.
- [55] Xianyong J, Hou M, Wu G, et al. Research on maneuvering decision algorithm based on improved deep deterministic policy gradient[J]. IEEE Access, 2022, 10: 92426-92445.
- [56] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods[C].International conference on machine learning. PMLR, 2018: 1587-1596.
- [57] Zhang S, Li Y, Dong Q. Autonomous navigation of UAV in multi-obstacle environments based on a Deep Reinforcement Learning approach[J]. Applied Soft Computing, 2022, 115: 108194.
- [58] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C].International conference on machine learning. PMLR, 2018: 1861-1870.
- [59] Li B, Huang J, Bai S, et al. Autonomous air combat decision-making of UAV based on parallel self-play reinforcement learning[J]. CAAI Transactions on Intelligence Technology, 2023, 8(1): 64-81.
- [60] 刘承相,熊俊杰,魏秀玲,孙多.多无人机协同控制现状及发展前景探究[J].电子测试,2022,(16):110-112.
- [61] Gaerther U, Chung T H. UAV swarm tactics: an agent-based simulation and Markov process analysis[M]. Monterey, California: Naval Postgraduate School, 2013.
- [62] Ernest N, Carroll D, Schumacher C, et al. Genetic fuzzy based artificial intelligence for unmanned combat aerial vehicle control in simulated air combat missions[J]. Journal of Defense Management, 2016, 6(1): 2167-0374.
- [63] Ramirez-Atencia C, Bello-Orgaz G, R-Moreno M D, et al. Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms[J]. Soft Computing,

2017, 21: 4883-4900.

- [64] Zhen Z, Xing D, Gao C. Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm[J]. Aerospace Science and Technology, 2018, 76: 402-411.
- [65] Zhang G, Li Y, Xu X, et al. Multiagent reinforcement learning for swarm confrontation environments[C].Intelligent Robotics and Applications: 12th International Conference, ICIRA 2019, Shenyang, China, August 8–11, 2019, Proceedings, Part III 12. Springer International Publishing, 2019: 533-543.
- [66] Sun Z, Piao H, Yang Z, et al. Multi-agent hierarchical policy gradient for air combat tactics emergence via self-play[J]. Engineering Applications of Artificial Intelligence, 2021, 98: 104112.
- [67] Zhao W, Chu H, Miao X, et al. Research on the multiagent joint proximal policy optimization algorithm controlling cooperative fixed-wing UAV obstacle avoidance[J]. Sensors, 2020, 20(16): 4546.
- [68] Heidlauf P, Collins A, Bolender M, et al. Verification Challenges in F-16 Ground Collision Avoidance and Other Automated Maneuvers[C].ARCH@ ADHS. 2018: 208-217.
- [69] Wang M, Wang L, Yue T, et al. Influence of unmanned combat aerial vehicle agility on short-range aerial combat effectiveness[J]. Aerospace Science and Technology, 2020, 96: 105534.
- [70] Wei Y J, Zhang H P, Huang C Q. Maneuver Decision-Making For Autonomous Air Combat Through Curriculum Learning And Reinforcement Learning With Sparse Rewards[J]. arXiv preprint arXiv:2302.05838, 2023.
- [71] Hu Y, Wang W, Jia H, et al. Learning to utilize shaping rewards: A new approach of reward shaping[J]. Advances in Neural Information Processing Systems, 2020, 33: 15931-15941.
- [72] Piao H, Sun Z, Meng G, et al. Beyond-visual-range air combat tactics auto-generation by reinforcement learning[C].2020 international joint conference on neural networks (IJCNN). IEEE, 2020: 1-8.
- [73] Wang A, Zhao S, Shi Z, et al. Over-the-Horizon Air Combat Environment Modeling and Deep Reinforcement Learning Application[C].2022 4th International Conference on Data-driven Optimization of Complex Systems (DOCS). IEEE, 2022: 1-6.
- [74] Hu J, Wang L, Hu T, et al. Autonomous maneuver decision making of dual-UAV cooperative air combat based on deep reinforcement learning[J]. Electronics, 2022, 11(3): 467.

- [75] Zhan G, Zhang X, Li Z, et al. Multiple-uav reinforcement learning algorithm based on improved ppo in ray framework[J]. Drones, 2022, 6(7): 166.
- [76] Narvekar S, Sinapov J, Stone P. Autonomous Task Sequencing for Customized Curriculum Design in Reinforcement Learning[C].IJCAI. 2017: 2536-2542.
- [77] Liu X, Yin Y, Su Y, et al. A Multi-UCAV Cooperative Decision-Making Method Based on an MAPPO Algorithm for Beyond-Visual-Range Air Combat[J]. Aerospace, 2022, 9(10): 563.
- [78] Sun Z, Piao H, Yang Z, et al. Multi-agent hierarchical policy gradient for air combat tactics emergence via self-play[J]. Engineering Applications of Artificial Intelligence, 2021, 98: 104112.
- [79] Tobin J, Fong R, Ray A, et al. Domain randomization for transferring deep neural networks from simulation to the real world[C].2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2017: 23-30.
- [80] Comanici G, Precup D. Optimal policy switching algorithms for reinforcement learning[C].Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1. 2010: 709-714.
- [81] Rane S. Learning with curricula for sparse-reward tasks in deep reinforcement learning[D]. Massachusetts Institute of Technology, 2020.
- [82] Barto A G, Mahadevan S. Recent advances in hierarchical reinforcement learning[J]. Discrete event dynamic systems, 2003, 13(1-2): 41-77.
- [83] Pope A P, Ide J S, Mićović D, et al. Hierarchical reinforcement learning for air-to-air combat[C].2021 international conference on unmanned aircraft systems (ICUAS). IEEE, 2021: 275-284.
- [84] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C].International conference on machine learning. PMLR, 2018: 1861-1870.
- [85] Kong W, Zhou D, Du Y, et al. Hierarchical multi-agent reinforcement learning for multi-aircraft close-range air combat[J]. IET Control Theory & Applications, 2023, 17(13): 1840-1862.
- [86] Selmonaj A, Szehr O, Del Rio G, et al. Hierarchical Multi-Agent Reinforcement Learning for Air Combat Maneuvering[J]. arXiv preprint arXiv:2309.11247, 2023.
- [87] Kong W, Zhou D, Zhou Y, et al. Hierarchical reinforcement learning from competitive self-play for dual-aircraft formation air combat[J]. Journal of Computational Design and

- Engineering, 2023, 10(2): 830-859.
- [88] 高晓光. 航空军用飞行器导论[M]. 西北工业大学出版社,2004,12.
- [89] Afsar M M, Crump T, Far B. Reinforcement learning based recommender systems: A survey[J]. ACM Computing Surveys, 2022, 55(7): 1-38.
- [90] Song W, Duan Z, Yang Z, et al. Explainable knowledge graph-based recommendation via deep reinforcement learning[J]. arXiv, et al. "Explainable knowledge graph-based recommendation via deep reinforcement learning." arXiv preprint arXiv:1906.09506 13 (2019).
- [91] Wang X, Xu Y, He X, et al. Reinforced negative sampling over knowledge graph for recommendation[C].Proceedings of the web conference 2020. 2020: 99-109.
- [92] Xian Y, Fu Z, Muthukrishnan S, et al. Reinforcement knowledge graph reasoning for explainable recommendation[C].Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. 2019: 285-294.
- [93] Zhao K, Wang X, Zhang Y, et al. Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs[C].Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. 2020: 239-248.
- [94] Liu Q, Tong S, Liu C, et al. Exploiting cognitive structure for adaptive learning[C].Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 627-635.
- [95] Wang P, Fan Y, Xia L, et al. KERL: A knowledge-guided reinforcement learning model for sequential recommendation[C].Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020: 209-218.
- [96] Zhou S, Dai X, Chen H, et al. Interactive recommender system via knowledge graph-enhanced reinforcement learning[C].Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. 2020: 179-188.
- [97] Lei W, Zhang G, He X, et al. Interactive path reasoning on graph for conversational recommendation[C].Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020: 2073-2083.
- [98] Deng Y, Li Y, Sun F, et al. Unified conversational recommendation policy learning via graph-based reinforcement learning[C].Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021: 1431-1441.
- [99] Ammanabrolu P, Riedl M O. Playing text-adventure games with graph-based deep

- reinforcement learning[J]. arXiv preprint arXiv:1812.01628, 2018.
- [100] Hausknecht M, Ammanabrolu P, Côté M A, et al. Interactive fiction games: A colossal adventure[C].Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(05): 7903-7910.
- [101] Xu Y, Fang M, Chen L, et al. Deep reinforcement learning with stacked hierarchical attention for text-based games[J]. Advances in Neural Information Processing Systems, 2020, 33: 16495-16507.
- [102] Adhikari A, Yuan X, Côté M A, et al. Learning dynamic belief graphs to generalize on text-based games[J]. Advances in Neural Information Processing Systems, 2020, 33: 3045-3057.
- [103] Ammanabrolu P, Riedl M O. Transfer in deep reinforcement learning using knowledge graphs[J]. arXiv preprint arXiv:1908.06556, 2019.
- [104] Murugesan K, Atzeni M, Shukla P, et al. Enhancing text-based reinforcement learning agents with commonsense knowledge[J]. arXiv preprint arXiv:2005.00811, 2020.
- [105] Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature, 2019, 575(7782): 350-354.
- [106] Reis S, Reis L P, Lau N. Game adaptation by using reinforcement learning over meta games[J]. Group Decision and Negotiation, 2021, 30(2): 321-340.
- [107] You J, Liu B, Ying Z, et al. Graph convolutional policy network for goal-directed molecular graph generation[J]. Advances in neural information processing systems, 2018, 31.
- [108] Do K, Tran T, Venkatesh S. Graph transformation policy network for chemical reaction prediction[C].Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 750-760.
- [109] Wang S, Ren P, Chen Z, et al. Order-free medicine combination prediction with graph convolutional reinforcement learning[C].Proceedings of the 28th ACM international conference on information and knowledge management. 2019: 1623-1632.
- [110] Sun Z, Dong W, Shi J, et al. Interpretable disease prediction based on reinforcement path reasoning over knowledge graphs[J]. arXiv preprint arXiv:2010.08300, 2020.
- [111] Piplai A, Ranade P, Kotal A, et al. Using knowledge graphs and reinforcement learning for malware analysis[C].2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020: 2626-2633.
- [112] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving:

- A survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(6): 4909-4926.
- [113] Kober J, Bagnell J A, Peters J. Reinforcement learning in robotics: A survey[J]. The International Journal of Robotics Research, 2013, 32(11): 1238-1274.
- [114] Cui Y, Matsubara T, Sugimoto K. Pneumatic artificial muscle-driven robot control using local update reinforcement learning[J]. Advanced Robotics, 2017, 31(8): 397-412.
- [115] 许雅筑,武辉,游科友等.强化学习方法在自主水下机器人控制任务中的应用[J].中国科学:信息科学,2020,50(12):1798-1816.
- [116] Yahya A, Li A, Kalakrishnan M, et al. Collective robot reinforcement learning with distributed asynchronous guided policy search[C].2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 79-86.
- [117] Akila V, Christaline J A, Mani A J, et al. Reinforcement Learning For Walking Robot[C].IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021, 1070(1): 012075.
- [118] Uc-Cetina V, Navarro-Guerrero N, Martin-Gonzalez A, et al. Survey on reinforcement learning for language processing[J]. Artificial Intelligence Review, 2023, 56(2): 1543-1575.
- [119] 徐聪. 基于深度学习和强化学习的对话模型研究[D].北京科技大学,2020.
- [120] Kaiser M, Saha Roy R, Weikum G. Reinforcement learning from reformulations in conversational question answering over knowledge graphs[C].Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021: 459-469.
- [121] Christmann P, Saha Roy R, Abujabal A, et al. Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion[C].Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 729-738.
- [122] Kasim M F. Playing the game of Congklak with reinforcement learning[C].2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE, 2016: 1-5.
- [123] Hu H, et al. "Rationalizing translation elongation by reinforcement learning." bioRxiv (2018): 463976.
- [124] Mahadevan S, Marchalleck N, Das T K, et al. Self-improving factory simulation using continuous-time average-reward reinforcement learning[C].MACHINE LEARNING-

INTERNATIONAL WORKSHOP THEN CONFERENCE-. MORGAN KAUFMANN PUBLISHERS, INC., 1997: 202-210.

- [125] 张磊. 基于强化学习的多无人机协同控制算法研究[D].中国科学院大学(中国科学院长春光学精密机械与物理研究所),2023.
- [126] 马昂,于艳华,杨胜利等.基于强化学习的知识图谱综述[J].计算机研究与发展,2022,59(08):1694-1722.
- [127] 严浙平,杨泽文,王璐等.马尔可夫理论在无人系统中的研究现状[J].中国舰船研究,2018,13(06):9-18.DOI:10.19693/j.issn.1673-3185.01199.
- [128] Ji X, Li J. Online motion planning for UAV under uncertain environment[C].2015 8th International Symposium on Computational Intelligence and Design (ISCID). IEEE, 2015, 2: 514-517.
- [129] Ure N K, Chowdhary G, How J P, et al. Health aware planning under uncertainty for UAV missions with heterogeneous teams[C].2013 European Control Conference (ECC). IEEE, 2013: 3312-3319.
- [130] Kiam J J, Schulte A. Multilateral quality mission planning for solar-powered long-endurance UAV[C].2017 IEEE Aerospace Conference. IEEE, 2017: 1-10.
- [131] 陈少飞. 无人机集群系统侦察监视任务规划方法[D].国防科学技术大学,2017.
- [132] 李月娟,吕永健,常迁臻等.基于 MMDP 的无人作战飞机任务分配模型研究[J].计算机应用与软件,2013,30(07):276-279+286.
- [133] Jeong B M, Ha J S, Choi H L. MDP-based mission planning for multi-UAV persistent surveillance[C].2014 14th International Conference on Control, Automation and Systems (ICCAS 2014). IEEE, 2014: 831-834.
- [134] 霍璐. 旋翼无人机对地目标检测与态势估计方法研究[D].中国民航大学,2017.
- [135] Baek S S, Kwon H, Yoder J A, et al. Optimal path planning of a target-following fixed-wing UAV using sequential decision processes[C].2013 IEEE/RSJ International conference on intelligent robots and systems. IEEE, 2013: 2955-2962.
- [136] Ragi S, Chong E K P. UAV path planning in a dynamic environment via partially observable Markov decision process[J]. IEEE Transactions on Aerospace and Electronic Systems, 2013, 49(4): 2397-2412.
- [137] Vanegas F, Campbell D, Eich M, et al. UAV based target finding and tracking in GPS-denied and cluttered environments[C].2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016: 2307-2313.
- [138] Vanegas F, Campbell D, Roy N, et al. UAV tracking and following a ground target under

- motion and localisation uncertainty[C].2017 IEEE Aerospace Conference. IEEE, 2017: 1-10.
- [139] Vanegas F, Gonzalez F. Uncertainty based online planning for UAV target finding in cluttered and GPS-denied environments[C].2016 IEEE Aerospace Conference. IEEE, 2016: 1-9.
- [140] 郭军,朱凡,刘远飞.基于马尔可夫链预测的多无人机协同搜索控制[J].弹箭与制导学报,2007(05):315-318.
- [141] 左青海,潘卫军,卓星宇等.无人机感知与避撞方法研究[J].装备制造技术,2016(11):65-69.
- [142] Fu Y, Yu X, Zhang Y. Sense and collision avoidance of unmanned aerial vehicles using Markov decision process and flatness approach[C].2015 IEEE International Conference on Information and Automation. IEEE, 2015: 714-719.
- [143] Krishnamoorthy K, Pachter M, Chandler P, et al. UAV perimeter patrol operations optimization using efficient dynamic programming[C].Proceedings of the 2011 American Control Conference. IEEE, 2011: 462-467.
- [144] 陈占海,祝小平.无人作战飞机数据链延时对攻击决策的影响及其补偿[J].科学技术与工程,2011,11(09):2043-2047.
- [145] Vaca F E. National Highway Traffic Safety Administration (NHTSA) notes[J]. Annals of Emergency Medicine, 2004, 43(2): 275-277.
- [146] 侯海晶. 高速公路驾驶人换道意图识别方法研究[D]. 长春: 吉林大学, 2013.
- [147] 高振海. 驾驶员无意识车道偏离识别方法[J]. 吉林大学学报 (工学版), 2017, 47(3): 709-716. Zeng X, Fu H, Dai B, et al. A new approach for dense depth image recovery[C].2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics. IEEE, 2015, 2: 491-496.
- [148] 谷明琴. 复杂环境中交通标识识别与状态跟踪估计算法研究[D]. 长沙: 中南大学, 2013.
- [149] Cuenca Á, Ojha U, Salt J, et al. A non-uniform multi-rate control strategy for a Markov chain-driven Networked Control System[J]. Information Sciences, 2015, 321: 31-47.
- [150] Thulasiraman P, Sagir Y. Dynamic bandwidth provisioning using Markov chain based RSVP for unmanned ground networks[C].2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA). IEEE, 2014: 130-136.
- [151] Thulasiraman P, Clark G A, Beach T M. Mobility estimation using an extended Kalman

- filter for unmanned ground vehicle networks[C].2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA). IEEE, 2014: 223-229.
- [152] Kawano H. Real-time obstacle avoidance for underactuated autonomous underwater vehicles in unknown vortex sea flow by the mdp approach[C].2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2006: 3024-3031.
- [153] Teck T Y, Chitre M. Single beacon cooperative path planning using cross-entropy method[C].OCEANS'11 MTS/IEEE KONA. IEEE, 2011: 1-6.
- [154] Tan Y T, Gao R, Chitre M. Cooperative path planning for range-only localization using a single moving beacon[J]. IEEE Journal of Oceanic Engineering, 2014, 39(2): 371-385.
- [155] 温涛, 许枫, 杨娟, 等. 连续隐马尔可夫模型在多基地目标识别中的应用[J]. 应用声学, 2017, 36(6): 512-520.
- [156] 吴小勇. 反潜体系的搜索能力优化方法研究[D]. 长沙: 国防科学技术大学, 2012.
- [157] Myers V, Williams D P. A POMDP for multi-view target classification with an autonomous underwater vehicle[C].OCEANS 2010 MTS/IEEE SEATTLE. IEEE, 2010: 1-5.
- [158] Myers V, Williams D P. Adaptive multiview target classification in synthetic aperture sonar images using a partially observable Markov decision process[J]. IEEE Journal of Oceanic Engineering, 2011, 37(1): 45-55.
- [159] 詹艳梅, 孙进才, 胡友峰. 纯方位目标运动分析的自适应算法[J]. 西北工业大学学报, 2002, 20(4): 647-650.
- [160] 王彪, 曾庆军, 夏捷. 一种有效的水下目标运动分析算法与仿真研究[J]. 系统仿真学报, 2012, 24(11): 2410-2413.
- [161] 陈钊, 刘郑国, 王海燕, 等. 基于 RJMCMC 方法的水下被动目标声源数和方位联合估计[J]. 鱼雷技术, 2011, 19(6): 415-422.
- [162] 梁庆卫, 孙天元, 蒋姗姗. 基于马尔可夫链的水下移动网络可靠性研究[J]. 鱼雷技术, 2014, 22(3): 165-168.
- [163] 刘友永. 水声差分跳频通信关键技术研究[D]. 哈尔滨: 哈尔滨工程大学, 2009.
- [164] 徐君锋. 无线传感器网络中通信可靠性和能量有效性的机制研究[D]. 大连: 大连理工大学, 2011.
- [165] 钟贞魁. 基于优先级 QoS 选择策略的水下网络中继算法[J]. 计算机仿真, 2016 (1): 317-320.
- [166] 苏毅珊, 金志刚, 窦飞. 高性能水下传感器网络多信道 MAC 协议[J]. 哈尔滨工程

- 大学学报, 2015, 36(7): 987-991.
- [167] 张艳莉, 黄军辉. 一种基于马尔可夫链的水下频谱预测方法[J]. 广东轻工职业技术学院学报, 2012, 11(4): 15-19.
- [168] 迟凤阳. 水下航行器组合导航定位技术研究[D]. 哈尔滨: 哈尔滨工程大学, 2015.
- [169] 陈鹏云. 多传感器条件下的 AUV 海底地形匹配导航研究[D]. 哈尔滨: 哈尔滨工程大学, 2016.
- [170] Bellman R E. A Markov decision progress[J]. Journal of Mathematical Mechanics, 1957, 6(5):679-684.
- [171] 张忠铝. 基于 SAC 算法的 AUV 航路规划与跟踪方法研究[D]. 山东大学, 2023. DOI : 10.27272/d.cnki.gshdu.2023.006138
- [172] Doya K. Reinforcement learning in continuous time and space[J]. Neural computation, 2000, 12(1): 219-245.
- [173] Sutton R S, McAllester D, Singh S, et al. Policy gradient methods for reinforcement learning with function approximation[J]. Advances in neural information processing systems, 1999, 12.
- [174] Konda V, Tsitsiklis J. Actor-critic algorithms[J]. Advances in neural information processing systems, 1999, 12.
- [175] Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial intelligence, 112(1-2), 181-211.
- [176] Parr, R., & Russell, S. J. (1998). Reinforcement learning with hierarchies of machines. In Advances in neural information processing systems (pp. 1043-1049).
- [177] Dietterich, T. G. (2000). Hierarchical reinforcement learning with the MAXQ value function decomposition. J. Artif. Intell. Res.(JAIR), 13, 227-303.
- [178] 周欢, 赵辉, 韩统等. 基于规则的无人机集群飞行与规避协同控制[J]. 系统工程与电子技术, 2016, 38(06):1374-1382.
- [179] 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.
- [180] Bellingham J, Tillerson M, Richards A, et al. Multi-task allocation and path planning for cooperating UAVs[J]. Cooperative control: models, applications and algorithms, 2003: 23-41.
- [181] 胡中华, 赵敏, 姚敏等. 一种改进蚂蚁算法的无人机多目标三维航迹规划[J]. 沈阳工业大学学报, 2011, 33(05):570-575.
- [182] Swartzentruber L, Foo J L, Winer E. Multi-objective UAV path planning with refined

- reconnaissance and threat formulations[C].51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 18th AIAA/ASME/AHS Adaptive Structures Conference 12th. 2010: 2758.
- [183] 史志富.基于贝叶斯网络的 UCAV 编队对地攻击智能决策研究[D].西北工业大学,2007.
- [184] Gupta S, Deep K. A novel random walk grey wolf optimizer[J]. Swarm and evolutionary computation, 2019, 44: 101-112.
- [185] Rodríguez L, Castillo O, Soria J, et al. A fuzzy hierarchical operator in the grey wolf optimizer algorithm[J]. Applied Soft Computing, 2017, 57: 315-328.
- [186] Daniel E. Optimum wavelet-based homomorphic medical image fusion using hybrid genetic–grey wolf optimization algorithm[J]. IEEE Sensors Journal, 2018, 18(16): 6804-6811.
- [187] Jayabarathi T, Raghunathan T, Adarsh B R, et al. Economic dispatch using hybrid grey wolf optimizer[J]. Energy, 2016, 111: 630-641.
- [188] Wang N, He Q. Seagull optimization algorithm combining golden sine and sigmoid continuity[J]. Appl. Res. Comput, 2022, 39: 157-162.
- [189] 周瑞,黄长强,魏政磊,等. MP-GWO 算法在多 UCAV 协同航迹规划中的应用[J]. 空军工程大学学报(自然科学版), 2017, 18(05): 24-29.
- [190] 郑志伟. 空空导弹系统概论 [M]. 北京:兵器工业出版社, 1997.
- [191] 徐啟云,王洁,罗畅等.无人机对地自主攻击方法仿真与研究[J].计算机仿真,2015,32(12):55-59+106.
- [192] 罗广彬,洪成雨,程志良等.基于 BP 和 TBPNN 神经网络的混凝土抗压强度预测研究 [J].混凝土,2023,(03):37-41.
- [193] Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., & Levine, S. (2021). How to train your robot with deep reinforcement learning: lessons we have learned. The International Journal of Robotics Research, 40(4-5), 698-721.
- [194] 蒲小勃, 缪炜星. 超视距空战中机载雷达的使用策略研究[J]. 电光与控制, 2012, 19(6): 1-4.
- [195] 张建东, 王鼎涵, 杨啟明, 等. 基于分层强化学习的无人机空战多维决策[J]. 兵工学报, 2023, 44(6): 1547.
- [196] 王翔.猝发通信技术在超短波通信系统中的应用[J].电子制作 2013(11):116

致 谢

岁月流转，时光飞逝，不觉间我就要迎来属于我的研究生毕业季了，在西北工业大学的硕士学习生活中，我收获颇丰，不仅在专业领域得到了深入的学习和研究，还拓宽了视野，领略了不同的风光，更重要的是，我学到了许多宝贵的人生经验和道理。在这个特别的时刻，我想对所有在我成长路上给予帮助和关爱的人表示深深的感谢。

我衷心感谢我的导师 XXX 副教授。回想研究生初期，XXX 老师的关怀和支持给了我巨大的帮助。在整个研究生阶段，XXX 老师以其深厚的学识指导我解决学术问题，严谨的态度教导我科研方法，踏实的工作风格帮助我不断进步。特别是在我迷茫时，老师用温柔的话语为我指引方向，用无尽的耐心引领我成长。此外，感谢 XX 教授，在我申请博士需要推荐信时，尽管 XX 教授工作繁忙，他仍然不遗余力地抽出时间，为我撰写了这封至关重要的信件，并确保及时递交到我心仪的学校，XX 教授的这种关怀让我深受感动。感谢 XXX 副教授，他在我遇到困难时总是耐心聆听，用细腻的关怀和智慧的话语安抚我的不安，并帮助我找到解决问题的途径。这种关心不仅体现在学术指导下，更是一种精神上的支撑，让我感受到了教师这一职业的温暖和伟大。感谢 XXX 副教授，在我撰写论文的过程中，老师不厌其烦地带着我修改论文中的每个细节，老师一丝不苟的态度以及对学术严谨的坚持都让我深感敬佩。我还要向 XXX 老师表达特别的感激之情，在我的学术旅程中，他一直是我的引路人，带领我不断前行。XXX 老师以其温和的性格和鼓舞人心的言语，经常为我注入力量，并激发我不断提升自己努力前行。在此，我衷心祝愿所有老师身体健康、工作顺利。

感谢硕士期间与我一起成长的伙伴们。感谢 430 教研室的 XXX、XXX、XXX、XXX、XXX、XXX 等同学对于我学业、科研和生活上的帮助，感谢你们为我的教研室生活增添了许多欢乐。特别感激 XXX 师弟，在我面对学术难题时，他总是慷慨地伸出援手。他不仅帮助我解决了许多棘手的问题，为我提供宝贵的指导和建议。感谢 XXX 寝室的室友们——XXX、XXX、XXX，她们对我所展现的关爱和包容让我深感温暖，与她们共度的每一刻，我总是感到无比的快乐和放松，尽管我们相识的时间不足三年，但我们之间的友谊却仿佛经历了十几年的深厚积累，一见如故，想到不久后即将与她们告别，我的心中充满了深深的不舍，非常感谢她们给我的研究生生活留下的许多美好回忆。我还要向我的朋友 XXX 表达感激，在你的陪伴下，我的申博之路变得不再孤独，你的引导和支持使我变得更加坚强和坚定，在我面对难关时，你总是在我身边，提供帮助和宝贵的建议，我非常庆幸能与你成为朋友。在此，愿你们云程发轫，万里可期。

我非常感激始终支持我的父母，他们不仅无条件地为我提供经济和物质上的支持，还给予了我深厚的情感依靠和心灵的慰藉。在漫长的学习旅程中，每次与他们的通电话

都如同温暖的阳光，照亮我前行的道路，给我带来无尽的爱和温暖。我衷心希望自己能成为一个让他们引以为傲的女儿，用我的努力和成就来回报他们无私的爱和付出。

最后，我要向西北工业大学表达我的深深感谢。在这里我参与过丰富多彩的活动、见证了校园四季的美景：春日的玉兰，夏夜的晚霞，秋季的银杏，以及冬日的飘雪。这些美好的回忆将永远珍藏在我的心中。我衷心祝愿学校未来发展得更加繁荣昌盛。

行文至此，感慨万分。虽纸张有限，但对于各位的感激无限。希望以上每一位，在未来的岁月里幸福、顺遂。

在学期间发表的学术成果和参加科研情况

发表学术论文:

- [1] ***. Evaluation of Autonomous Capability of Ground Attack UAV Based on Hierarchical Analysis Method[C]. International Conference on Autonomous Unmanned Systems. Singapore: Springer Nature Singapore, 2022, 9:1072-1081. (一作, EI Index : 20231313822517)
- [2] ***. Research on threat assessment method of formation cooperative combat in a complex environment (已录用, 本人第四作者)

发明专利:

- [1] ***. 无人机自主对地攻击能力评价方法[P].中国专利.申请号: ZL202205096301

参与科研项目:

- [1] 中国试飞院, 无人机作战自主对地攻击评价体系研究。2021.05-2021.12.
- [2] 中航 601 所, 所综合通信识别仿真系统。2022.11-2023.04.
- [3] 中航 611 所, 编队协同智能战术决策模型和互操作设计。2023.04-2023.09.

所获荣誉奖项:

- [1] 西北工业大学研究生二等奖学金, 2021 年;
- [2] 西北工业大学研究生二等奖学金, 2022 年;
- [3] 西北工业大学优秀研究生荣誉称号, 2022 年;
- [4] 西北工业大学研究生二等奖学金, 2023 年;
- [5] 西北工业大学优秀研究生荣誉称号, 2023 年。

西北工业大学

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西北工业大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文被查阅和借阅。学校可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西北工业大学。

保密论文待解密后适用本声明。

学位论文作者签名: _____

指导教师签名: _____

年 月 日

年 月 日

西北工业大学

学位论文原创性声明

秉承学校严谨的学风和优良的科学道德，本人郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容和致谢的地方外，本论文不包含任何其他个人或集体已经公开发表或撰写过的研究成果，不包含本人或其他已申请学位或其他用途使用过的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式表明。

本人学位论文与资料若有不实，愿意承担一切相关的法律责任。

学位论文作者签名: _____

年 月 日