



同 济 大 学

第十八届“挑战杯”全国大学生课外学术科技作品竞赛
“揭榜挂帅”专项赛

复杂海域下多无人船覆盖探查系统
设计报告

参 赛 学 校	同济大学
选题发榜单位	中国船舶集团有限公司系统工程研究院 中船智海创新研究院有限公司
题 目 名 称	“区域信息采集多智能体控制模型”比赛方案
作 品 名 称	复杂海域下多无人船覆盖探查系统
申报者姓名	娄抒瀚、邓立原、赵浪舟、赵茜妍、 张洪菁、王劲博、柳深
指 导 老 师	龚炜、许佳、李莉

2023 年 9 月

摘要

海上区域信息采集是在海洋环境中获取各种信息的过程,当前面对复杂多变的海洋环境,需要进行多智能体协作的复杂海域内近距离目标探查工作。针对海上区域信息采集需要完成的复杂工作和相关指标要求,本工作建立了一个区域信息采集多智能体控制模型,将复杂的海上区域信息采集问题划分为区域覆盖、协同探查、智能体避障三个子问题,实现了复杂海域下多无人船覆盖探查系统。

针对区域覆盖问题,结合复杂海面信息采集任务场景建立了一个基于全区域有效覆盖和覆盖时间的多目标优化模型,并通过基于蜂窝离散分解将区域全覆盖问题简化为离散点序列遍历问题,然后基于启发式模拟退火算法进行最优区域分配及覆盖路径优化。经仿真测试,最终区域有效覆盖率达 99.67%。

针对协同探查问题,首先建立了多智能体同时进行区域覆盖和静目标探查的综合优化模型,然后设计了一种改进贪婪策略解决智能体内部决策及智能体间的调度问题,动态地对智能体进行动作决策与路径规划。经仿真测试,最终静目标有效探查指标和静目标正确探查指标均能达到 100%。

针对智能体避障问题,建立了速度坐标下的碰撞区域表示与避碰速度选取模型,采用分布式底层避碰算法 ORCA 将动态避碰问题转化为二次线性规划问题,并通过分层的责任分配机制实现了智能体对动态障碍与静态障碍的避碰能力,智能体可以无需通信与调度实现自主避障。经仿真测试,协同失败总次数为 0,即智能体不会进入其他智能体或静目标的威胁半径。

经仿真测试和五项指标计算,区域有效覆盖率得分 9.94 分;静目标有效探查指标得分 10 分;静目标正确探查指标得分 10 分;协同有效性指标得分 10 分;任务完成时长为 85 分 27 秒。

在实际工程应用中,本文提出的区域覆盖与协同探查模型与算法可以应用于不同形状和面积的未知海域的探查与覆盖问题,并且能够保证多智能体协同工作的负载均衡与时间效率,分布式的避障算法能够较好地推广到需要考虑船舶动力学约束的场景中。综上,本文所提出的模型与算法不仅在仿真环境中有较好性能,同时也具有较好的工程实现性和可扩展性。

目录

设计说明	4
1 国内外发展现状.....	4
1.1 区域覆盖路径规划.....	4
1.2 多智能体任务分配.....	6
1.3 多智能体协同避障.....	7
2 模型设计思路.....	8
2.1 问题的重述与分析.....	8
2.2 模型准备.....	9
2.2.1 模型假设.....	9
2.2.2 符号说明.....	9
2.3 模型建立.....	10
2.3.1 区域覆盖模型.....	10
2.3.2 协同探查模型.....	11
2.3.3 智能体避障模型.....	12
3 技术路线.....	14
3.1 总技术路线.....	14
3.2 区域覆盖策略.....	16
3.3 协同探查策略.....	21
3.4 智能体避障策略.....	22
4 模型的工程应用方法.....	24
4.1 模型的推广.....	24
4.2 工程部署方法.....	24
自测报告	26
1 计分方式.....	26
1.1 区域有效覆盖率.....	26
1.2 静目标有效探查指标.....	26
1.3 静目标正确探查指标.....	27
1.4 协同有效性指标.....	28
2 测试结果.....	28
2.1 区域有效覆盖率.....	29
2.2 静目标有效探查指标.....	29
2.3 静目标正确探查指标.....	30
2.4 协同有效性指标.....	30
2.5 全区域有效覆盖任务时长指标.....	30
2.6 小结.....	30
附录：参考文献	31

设计说明

1 国内外发展现状

海上区域信息采集是在海洋环境中获取各种信息的过程,包括海洋物理、化学、生物等方面的信息,对于海洋资源的开发利用和海洋环境的保护具有重要的意义^[1]。现代海洋信息采集技术已经发展到了一个比较成熟的阶段,包括海洋观测船、卫星遥感、无人机等。但是,海洋环境复杂多变,受海浪、海流、潮汐、气象条件、水温、盐度、海洋生物等多种因素相互作用的影响,海洋环境愈发复杂。此外,海底地形、海底沉积物、海底生物、海洋污染等因素也会造成海洋环境的变化,对传统平台探测载荷的探查精度造成较大影响。为应对复杂多变的海洋环境,一方面,通过近距离对目标进行观测能够获得较高的目标定位精度;另一方面,利用多个智能体协同完成指定复杂海域的目标探查工作可 effectively 提高探查精度、降低安全风险。

本文针对指定的不规则封闭海域进行多智能体的动态规划,要求各智能体实现全区域的有效覆盖,且保证能够探查到区域内所有静目标,并且在发现和探查不同属性的多个静目标过程中,能够以指定角度稳速稳向控制单个或多个智能体抵近静目标完成观测动作。针对该问题,需要实现多智能体的区域覆盖路径规划、探查任务分配以及协同避障三大主要任务。

1.1 区域覆盖路径规划

区域覆盖路径规划指利用移动智能体,在其物理接触范围或者在其传感器感知范围内遍历目标环境区域,并尽可能地满足任务完成时间短、重复路径少或未遍历区域小等优化目标。智能体全覆盖应用出现在军事、农业、工业、商业、灾难救援、城市生活等各个方面,例如自动排雷、作物收割、空中交通巡查等。多智能体的覆盖路径规划与单智能体相比,除了考虑遍历和路径的无重复外,还要考虑任务分配均衡性、资源冲突的挑战、智能体数目和任务环境的不确定性、智能体之间的协商和冲突、先验信息依赖性、算法自适应性和可扩展性等。Hazon 等^[2]将单智能体的螺旋生成树覆盖算法扩展到多智能体,并提出了带回溯优化的多智能体生成树覆盖(MSTC),提高了智能体覆盖的效率和鲁棒性。目前多智能体覆盖路径规划算法可分为3类:无规划、集中规划和离散规划。

随机路径规划算法^[3]无需地图信息,智能体采用随机反应式运动规划。取得理想的覆盖效果要以牺牲时间或投入大量的智能体为代价。集中规划算法在获得全局环境信息的条件下进行区域分配,智能体各自完成分配区域内的覆盖任务,可以产生更高效的路径,大大缩短

完成覆盖任务的持续时间，然而在实际应用中许多局限性。离散规划算法没有集中的任务分配机制，各智能体通过显式或隐式通信交换信息来协作完成覆盖。离散规划算法是一种在线的、实时的规划算法，可以适应未知环境下的覆盖，但其效果与算法本身关系密切。

许多工作致力于提高区域覆盖路径规划的性能效果。Mitra 等^[4]为了利用集中式框架的强大功能，提出了一种基于目标分配的后退地平线集中式在线同质多智能体规划器，确保未知工作空间的完全覆盖并在覆盖完成时间方面具有优势。Tang 等^[5]提出了一种新的多智能体持续覆盖策略，旨在优化覆盖性能，为每个智能体获得更平等的覆盖路径，同时保证避障和覆盖周期的最小化。Collins 等^[6]提供了一种新的多智能体覆盖路径规划算法（SCoPP），通过考虑智能体的初始状态来优化智能体之间的负载平衡，同时最大限度地减少非凸区域监测的完成时间。Li 等^[7]提出了一种在连续时间内制定的 RA-MCPP 路径规划器，以提高区域覆盖的可靠性概率度量。Gao 等^[8]提出了一种基于能量补偿和障碍物矢量化（ECOV）的完全覆盖路径规划（CCPP）算法。该算法具有路径覆盖重复率低、覆盖率高等优点，可适应复杂的地图模型。Vandermeulen 等^[9]提出了基于迭代启发式算法的覆盖方法，以最小化覆盖线的数量，从而减少路径中的转弯数。Gosrich 等^[10]提出了一种基于图形神经网络的多智能体分散控制策略，利用智能体之间的通信来显示共享相邻智能体间信息来进行控制决策，实现了更高的覆盖质量。

由于区域覆盖路径规划的强任务实践性，许多工作针对具体任务或场景提出算法。许多工作^{[11][12]}研究了自主或异质无人机在有限区域上的覆盖路径规划问题。Xie 等^[13]针对为一组有限电源约束的无人机找到最优路径，以完全覆盖多个不重叠的凸多边形区域的任务，开发了基于 Branch-and-Bound（BnB）的方法找到（接近）最优路径以及基于遗传算法的方法解决在不同目标下有效地大规模问题。Jiao 等^[14]提出一种新的多智能体覆盖路径规划算法 MACPP-MPC。该算法可以通过智能体之间的交互和协作，以提高系统在未知复杂环境中的适应性和鲁棒性。Jing 等^[15]提出了一个多 UAV 覆盖路径规划（CPP）框架，用于检测大规模、复杂三维结构。针对域上的密度函数最初未知时多智能体空间全覆盖的问题，Nakamura 等^[16]让一组移动智能体同时学习一个域上的空间场，并在空间上分布自己以优化覆盖路径。

多智能体区域覆盖需要面对智能体协商和冲突、覆盖性能的优化、算力与时间成本高等挑战。针对本赛题海域场景，本工作对多边形待探查海域采用了基于蜂窝离散分解和模拟退火算法（SA）的最优区域分配及覆盖路径优化。在初始阶段，智能体同时进行覆盖与探查的工作。直至覆盖任务完成后，所有智能体切换至探查工作模式。该路径规划策略优化了任务的实现效果，能完成基本全覆盖并在覆盖完成时间等指标上具有一定的优势。

1.2 多智能体任务分配

多智能体系统处理全覆盖行动的过程中,需要经过若干个阶段,包括任务分解、任务指派、任务调度等。这三个阶段统称为任务分配。多智能体任务分配(也称多机器人任务分配, Multi-Robot Task Allocation, MRTA)在满足各种约束条件的情况下,为系统中各个智能体规划出最佳的任务执行序列,在系统执行代价最小的情况下完成系统所需执行的全部任务,本质上是一个组合优化问题。MRTA 任务分配方案应该应对加速比、资源冲突性(例如路径冲突)、智能体数目不确定性(例如任务完成时间具有截止时限)或者环境的不确定性(例如环境概率性的转移)这三个挑战。目前解决多智能体任务分配的主要方法有:线性规划方法、基于市场机制的方法、启发式搜索算法等。

基于数学规划的分配方法通过线性规划算法可以将多智能体任务规划问题转化为数学运算^[17],虽能求得最优解,但具有较高的时间和空间复杂度,当问题规模增大时,求解难度也急剧增加,时间耗费呈指数增长。启发式分配方法^[18]通过对一个初始解决方案进行不断迭代从而逐步逼近最优解,具备适应性强、搜寻速度快等优点,但一般求到的是局部最优解。基于市场机制的方法^[19]是依据市场情况让多个智能体进行竞拍或谈判来实现任务的分配,主要包括拍卖算法和合同网算法。总体而言,采用基于启发式算法的任务分配策略对于复杂环境下的 MRTA 问题求解效果较好,是应用于实际任务分配的最佳策略。

Ponda 等^[20]针对多无人机协同搜救问题,考虑执行任务的无人机无法与控制中心建立连接的问题,通过预测网络拓扑,设置中继任务进行分配修复连接,确保任务执行时网络连通,结合拍卖算法实现了实时任务分配。Lee 等^[21]针对通信范围受限、资源受限情况下的 MRTA 问题,采用基于多跳网络结合分布式拍卖的求解策略,解决了资源和电量消耗下的实时最优分配问题。Luo 等^[22]针对任务存在执行时间和时间窗的 MRTA 问题,每个智能体受电池寿命和执行任务上限决定,提出一种基于动态规划结合分布式拍卖算法,实现多智能体分布式实时任务分配。针对动态扰动或故障,Neille 等^[23]引入了动态迭代任务分配图搜索(DITAGS),以在涉及异质团队的动态环境中实现韧性任务调度和运动规划问题。Luo 等^[24]则考虑了将任务分配给异构移动智能体团队的问题。Al-Hussaini 等^[25]提出了基于启发式的方法生成任务重新分配建议,以处理突发事件,并通过在具有挑战性的环境中运行的多智能体团队更快地完成任务。Bai 等^[26]设计了两个基于组的分布式拍卖算法实现了多智能体动态运输任务分配问题,量化了算法在所用智能体数量和智能体容量方面的潜力。Nie 等^[27]创新了实时多智能体任务分配,通过提出超图 MRS 模型和超图搜索算法,以协调异构的多智能

体系统,从而满足动态生产需求带来的频繁重构。分析和解决了频繁生产流程调整带来的挑战。**Park** 等^[28]提出了一个博弈论多智能体任务分配框架,使一大群智能体能够在动态变化的环境中优化分配任务。该算法使智能体能够学习并渐进地实现最优平稳任务分配。

由于群智任务与协同智能体的复杂性与不确定性,任务分配机制大多存在适用范围窄、匹配准确度不高、分配效率低等问题。本工作针对任务分配调度的效率最优化问题,设计与距离相关的代价函数,利用改进贪婪算法设计任务分配策略分阶段动态的将探查任务分配给各智能体,即分配给其最近的任务点,以缩短任务分配时间和整体运行时间。

1.3 多智能体协同避障

避障是多智能体覆盖路径规划的核心技术之一,即在起点与终点之间规划处一条无碰的且代价最小的路径。智能体除了需要高效完成任务,更要遵守外界的约束条件,保证导航过程的安全性,不与其他智能体或各类障碍物发生碰撞而造成不可估量的损失。在研究的早期,大多数学者致力于静态的,单个障碍物的简单情形,并取得了不错的效果,然而现实的环境是复杂的,动态的,甚至未知的,如何在动态复杂,未知的环境中高效避障与路径规划,至今仍是一个研究热点与难点。避障问题的研究主要集中在以下几个方面:①不同智能体速度与大小的避障;②规则与不规则障碍物的处理:规则障碍物一般是指可以用数学模型建模的障碍物,不规则障碍物由于无法直接建模,一般通过顶点近似的方法来处理;③静态和动态环境中的避障;④单智能体和多智能体:单个智能体不用考虑智能体之间的交互,多智能体系统需要考虑环境非平稳性的问题。其可分为集中式和分布式,集中式系统通过一个中央控制器实时统一协调,而分布式系统的每个智能体基于机载传感器进行独立决策,适合以较低的计算预算部署大量智能体。

传统避障方法有人工势场法、最优交互避碰法、蚁群算法等。人工势场法基于目标产生引力、障碍产生斥力的原则,通过对环境建立势场,将目标与障碍产生的合力作用于智能体,从而引导智能体运动状态^[29]。**Lazarowska**^[30]提出了离散人工势场算法(DAPF),通过构造离散势场实现动态方法,算法规划的路径长度更为合理,并且缩短了运行时间,但该算法需要在全局地图已知的情况下才能使用。最优交互避碰法(ORCA)将动态避碰问题转化为二次线性规划问题,在速度平面凸区域内求解^[31]。**Fulgenzi** 等^[32]将概率速度障碍(PVO)方法应用于动态占用网格,提出了一种估计碰撞概率的方法,其中障碍物的位置、形状和速度的不确定性、遮挡和有限的传感器范围直接影响计算。实验证明智能体能以恒定的线速度在障碍物中安全导航,但缺乏对非线性速度障碍物的处理。这些算法可以处理很多避障问题,但是

可能陷入局部最优，且在适应性、计算效率、自主性、稳定性等方面仍需进一步加强。近年来，人工智能算法也被广泛应用于多智能体环境的协同避障问题。基于深度神经网络的强大环境感知能力，监督学习首先得到了应用。许多工作利用机载传感器将运动数据输入神经网络，常用卷积神经网络提取环境特征，训练输出简单的避障指令^[33]。此外，深度强化学习（Deep Reinforcement Learning, DRL）也受到了很多关注^{[34][35]}，开始被广泛应用于避障的领域中。Llewyn 等^[36]以蝗虫的一种识别神经元为灵感，提出了一种人工神经网络模型并优化了模型参数，增强了模型的自适应性，将该模型与视觉传感器结合实现了智能体的动态避障。

本工作采用最优交互速度障碍法处理避障任务。在最优交互速度障碍法中，智能体各自承担二分之一的避障责任，因此只适用于同质的智能体避障。面对异质智能体、静止障碍物、智能体不对称等情况，均分的责任分配机制则无法灵活应对，且限制了导航性能。本工作针对此不足做出了改进。针对不同的避障情况，基于优先度分级灵活分配避障责任，并保证每对碰撞体的责任之和为 1。此改进拓展了动态避障与静态避障两种情况，实现了分层责任分配机制的优化。

2 模型设计思路

2.1 问题的重述与分析

海洋环境复杂多变，对传统平台探测载荷的探查精度影响较大。通常情况下，近距离进行目标观测能够获得较高的目标定位精度，在发现和探查不同属性的多个静目标过程中，需要以指定角度稳速稳向控制单个或多个智能体抵近静目标完成观测动作，确保可以获得目标的高精度位置、速度、图像或视频信息。利用智能体协助完成指定复杂海域的目标探查工作可有效提高探查精度、降低安全风险。

在一个指定的不规则封闭海域内进行区域全覆盖和区域内目标探查，需要解决以下三个问题：

区域覆盖（问题 1）： 需要利用多智能体完成在不规则封闭海域内全区域的有效覆盖，以发现所有的静目标，且智能体不能驶入禁航区；

协同探查（问题 2）： 智能体需要以指定方式完成所有静目标的探查任务，其中静目标的属性会动态改变；

智能体避障（问题 3）： 智能体需要保持航行安全，不能进入其他智能体或静目标的威

胁半径，需要进行碰撞规避。

对于上述问题，需要设计一个区域信息采集多智能体控制模型，通过模型能够对多智能体进行动态任务分配与路径规划，使得多智能体协同完成区域覆盖和静目标探查的任务。对于其中的三个子问题做如下的具体分析。

对于问题 1，将船舶视为有一定探查半径和探查角度范围的质点（仅在覆盖路径规划时视作质点），将静目标及其威胁半径内的区域视为障碍物，可以把问题被转化一个已知封闭平面区域的多智能体协同全覆盖路径规划问题。具体而言，该问题的目标是一个极小化极大问题，使得探查时间最长的智能体耗时最短，即任务完成时间最短。对于该问题，需要设计一种区域分配及路径规划算法，使得智能体能以尽可能短的工作时间来完成整个二维区域的覆盖任务。

对于问题 2，将静目标的探查视为一项任务，需要设计一种智能体内部决策算法，来使得智能体在覆盖任务、探查任务中进行选择。进一步考虑共享观测条件下的多智能体场景，还需要设计一个多智能体的调度算法。通过智能体内部决策算法和调度算法，智能体可以尽可能地完成覆盖任务及探查任务。

对于问题 3，将静目标及其他智能体视为静态障碍物和动态障碍物，需要设计一种避障算法，保证智能体在航行的过程中尽量保持安全。对于智能体而言，自主避障应是一项基本能力，因此避障算法应是分布式的，无需中心服务器调度。

2.2 模型准备

2.2.1 模型假设

为了方便对问题进行建模与求解，基于复审的赛题要求，对本模型的部分细节如下假设：

（1）智能体共享全局观测信息且信息的同步是无延时的，如只要静目标进入一个智能体的最大探测目标距离，该静目标的属性就可以立即同步到所有智能体，对所有智能体均是可知的。

（2）假设智能体在航行和执行探查的过程中无任何意外情况，并且能够按照的设定状态和速度运行。

2.2.2 符号说明

为方便模型的表述，现作符号说明如表 1 所示。

表 1 主要符号说明

符号	含义
S	任务区域（去除禁航区）
S_i	任务子区域
N_A	智能体数量， $N_A = 8$
N_O	静目标数量
$A = \{A_1, A_2, \dots, A_i, \dots, A_{N_A}\}$	智能体集合
$O = \{O_1, O_2, \dots, O_i, \dots, O_{N_O}\}$	静目标集合
v_{sail}	智能体属性：最大航行速度
v_{detect}	智能体属性：最大探查速度
r_{detect}	智能体属性：智能体探测目标距离
r_{threat}	智能体属性：智能体威胁范围
t_{detect}	同一智能体完成连续探查同一目标的等候时间
P_i	智能体 A_i 在任务区域的航行路径，由一系列路径点组成

*为简化表示，异构的智能体属性参数不含下标。

*其余符号将在下文出现时解释。

2.3 模型建立

2.3.1 区域覆盖模型

针对未知海域区域覆盖问题，由于任务区域边界已知，可以将多智能体的区域覆盖划分为两个子问题，即区域划分问题和单智能体覆盖路径问题。首先通过合理的区域划分将任务区域分配给各个智能体，每个智能体负责一个海域，在海域内完成单智能体的覆盖路径求解。结合两个子问题的解，能够解决多智能体在不规则封闭海域内完成全区域的有效覆盖问题。

在区域划分时，需要将任务区域 S 划分为子区域 S_i ：

$$S = \bigcup_{i=1}^{N_A} S_i。$$

假设将任务区域 S_i 分配给智能体 A_i 进行覆盖，则智能体 A_i 的路径 $P_i = \{P_i^j, j=1,2,\dots\}$ 应满足下式：

$$S_i = \bigcup_j D(P_i^j, r_{\text{detect}})，$$

式中 P_i^j 是构成路径 P_i 的点， $D(P, r) = \{Q | \|Q - P\| < r\}$ 表示一个以点 P 为圆心， r 为半径的圆形区域。

由于在开始覆盖前，任务区域内部的信息是未知的，因此在划分区域时无法提前考虑对区域内静目标的避障、以及边覆盖边探测的情况。为了达到高效的区域覆盖，划分的区域在分配给单智能体进行覆盖时，各智能体都要规划尽量短的路径。

综上，多智能体区域覆盖问题可以表述为一个带约束的优化问题：

$$\begin{aligned} & \min \max(|P_1|, |P_2|, \dots, |P_{N_A}|) \\ \text{s.t. } & S = \bigcup_{i=1}^{N_A} S_i = \bigcup_{i=1}^{N_A} \bigcup_j D(P_i^j, r_{\text{detect}}) \end{aligned}。$$

但是，考虑到实际应用中，全区域有效覆盖和覆盖的时间往往要进行平衡和取舍，且全覆盖并非强约束，故该问题用如下多目标优化模型表示更合理：

$$\begin{aligned} & \min \max(|P_1|, |P_2|, \dots, |P_{N_A}|) \\ & \max \left| \bigcup_{i=1}^{N_A} S_i \right| = \left| \bigcup_{i=1}^{N_A} \bigcup_j D(P_i^j, r_{\text{detect}}) \right| \end{aligned}。$$

2.3.2 协同探查模型

首先，假设智能体已经知道任务区域内的所有静目标属性，所有智能体只执行探查任务，则对于这样的问题，目标是在更短的时间内完成更多的有效探查。

由于一个静目标可能有多次探查任务，记任务区域中全部的待探查任务为 $TASK = \{TASK_1, TASK_2, \dots, TASK_j, \dots\}$ ，共包含 $\sum_i^{N_O} n_i$ 个任务。记智能体 A_i 完成的探查任务集为 $task_i$ 。

完成探查花费的时间包括智能体从自身当前位置前往探查起始点所需航行的时间，以及同一智能体连续探查同一静目标的等待时间 t_{detect} （如有）。则该问题的时间目标可以表示为：

$$\begin{cases} \min \max \left(\frac{|P_i|}{v_{\text{sail}}} + t_{\text{detect}} \right), i = 1, 2, \dots, N_A \\ \text{s.t. } \bigcup_{i=1}^{N_A} \text{task}_i = \text{TASK} \end{cases} \quad \circ$$

但是，考虑到实际应用中，对所有目标的完全探查和探查时间往往要进行平衡和取舍，且全部探查并非强约束，该问题用如下多目标优化模型表示更合理：

$$\begin{cases} \min \max \left(\frac{|P_i|}{v_{\text{sail}}} + t_{\text{detect}} \right), i = 1, 2, \dots, N_A \\ \max \left| \bigcup_{i=1}^{N_A} \text{task}_i \right| \end{cases} \quad \circ$$

上述模型的前提是智能体分别完成区域的覆盖和静目标探查，实际环境中，智能体在航行过程中可能同时会进行探查，即多智能体在位置的环境下航行，环境随时间变得部分可知，在此情况下，多智能体要同时继续进行区域覆盖和静目标探查。这个综合问题的优化模型可建立为：

$$\begin{cases} \min \max \left(\frac{|P_i|}{v_{\text{sail}}} + t_{\text{detect}} \right), i = 1, 2, \dots, N_A \\ \max \left| \bigcup_{i=1}^{N_A} \text{task}_i \right| \\ \max \left| \bigcup_{i=1}^{N_A} S_i \right| = \left| \bigcup_{i=1}^{N_A} \bigcup_j D(P_i^j, r_{\text{detect}}) \right| \end{cases} \quad \circ$$

解决该问题的关键是对智能体 A_i ，求解并控制其运行路径 P_i ，达到上述综合目标。

2.3.3 智能体避障模型

由于当前多智能体是在大范围海域的场景中执行任务，需要重点考虑的问题是在整体任务区域上的高效区域信息采集（覆盖和探查）多智能体控制方法，因此不将智能体避障考虑进上两节的优化模型中，仅将其视为智能体求解出路径后需要进行的局部调整与规划。下面对智能体避障子问题进行问题建模。

首先对于两个智能体，不妨记为 A_1, A_2 ，它们的位置可以分别表示为坐标 p_1, p_2 ，速度分别表示为 v_1, v_2 。

为了表示碰撞，将空间坐标转换为速度坐标（图 1a 和 1b）。对于智能体 A_1 ，将其视为质点，并作为速度坐标的原点。如果智能体 A_2 静止，那么在时间 τ 内，如果 $v_1 \cdot \tau$ 到达红色弧

线与两射线所夹范围时，两智能体就会碰撞。所以速度上限是 $\frac{v_{\text{弧线上的点}}}{\tau}$ ，即绿色弧线，即智能体 A_1 的速度只要不取到阴影范围区域 VO_{A_1/A_2}^τ ，在智能体 A_2 静止的情况下，两者在时间 τ 内不会发生碰撞。VO 区域是速度障碍，即 Velocity Obstacle，指的是智能体速度的一阶近似值，在给定时间范围内以这些速度运动，未来某个时间会导致与障碍物发生碰撞。

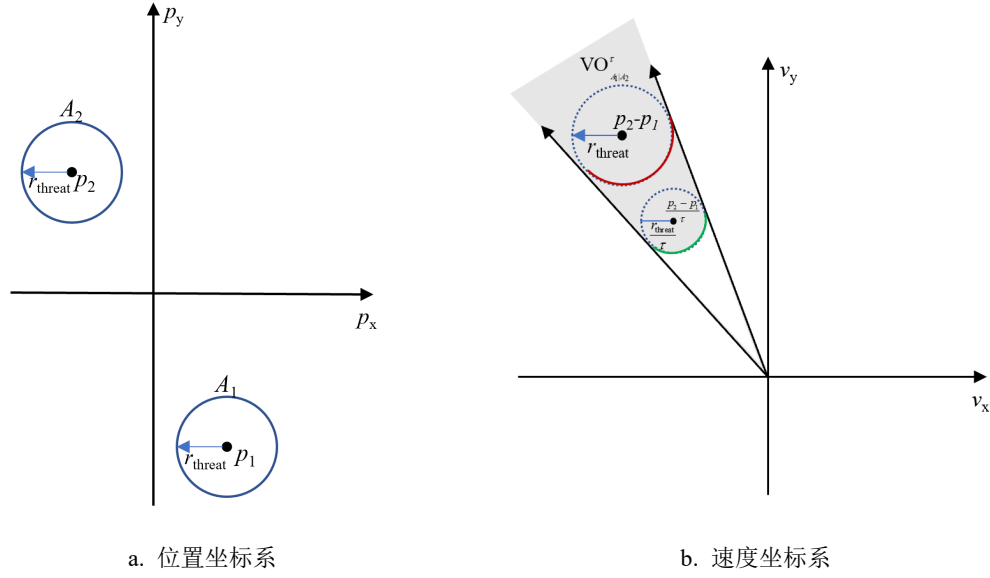


图 1 碰撞速度的二维平面表示

现考虑如果智能体 A_2 有速度，在速度坐标系下，智能体 A_2 可能的速度可以表示为圆形区域 $D_{v_2} = D((0,0), v_{sail})$ ，将 D_{v_2} 与 VO_{A_1/A_2}^τ 作闵氏和 $D_{v_2} \oplus VO_{A_1/A_2}^\tau = \{x + y \mid x \in D_{v_2}, y \in VO_{A_1/A_2}^\tau\}$ ，表示智能体 A_1 在考虑了智能体 A_2 可能的速度后，为了避免碰撞而不能选择的速度范围，在此范围外任选速度均可以避免碰撞。

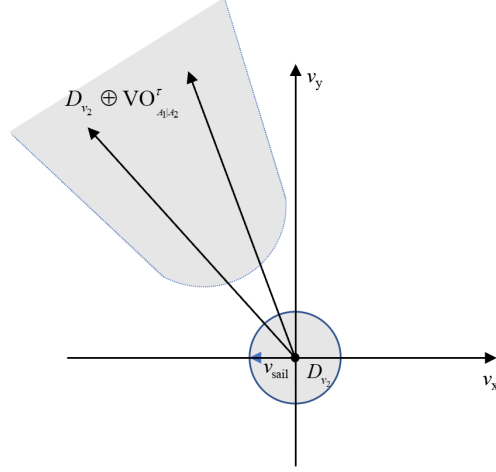


图 2 会碰撞的速度区域（闵氏和）

对于智能体 A_1 ，在考虑避障时，需要调整其速度 v_1 ，该速度不能偏离原有速度 v_1^- 太大，而且要满足速度在二维平面表示中，不落在上述描述的可能发生碰撞的速度范围区域内，可以表示为：

$$\begin{cases} \min |v_1 - v_1^-| \\ \text{s.t. } v_1 \notin \bigcup_i^{N_A} (D_{v_i} \oplus \text{VO}_{A_1|A_2}^r) \\ v_1 \in D((0,0), v_{\text{sail}}) \end{cases}$$

对于任意智能体 A_i 均可用上述方式选择避障速度。而对于静目标的避障，只需将静目标视为不具有速度的智能体即可，智能体 A_i 在避障的速度选择时加入约束 $v_i \notin \bigcup_j^{N_O} \text{VO}_{A_i|O_j}$ 即可。

3 技术路线

3.1 总技术路线

本工作将问题分解成三个子问题，并分别采用了区域覆盖策略、协同探查策略、智能体协同避碰策略，共同构成一个复杂海域下多无人船覆盖探查系统，如图 3 所示。整体算法流程如图 4 所示。

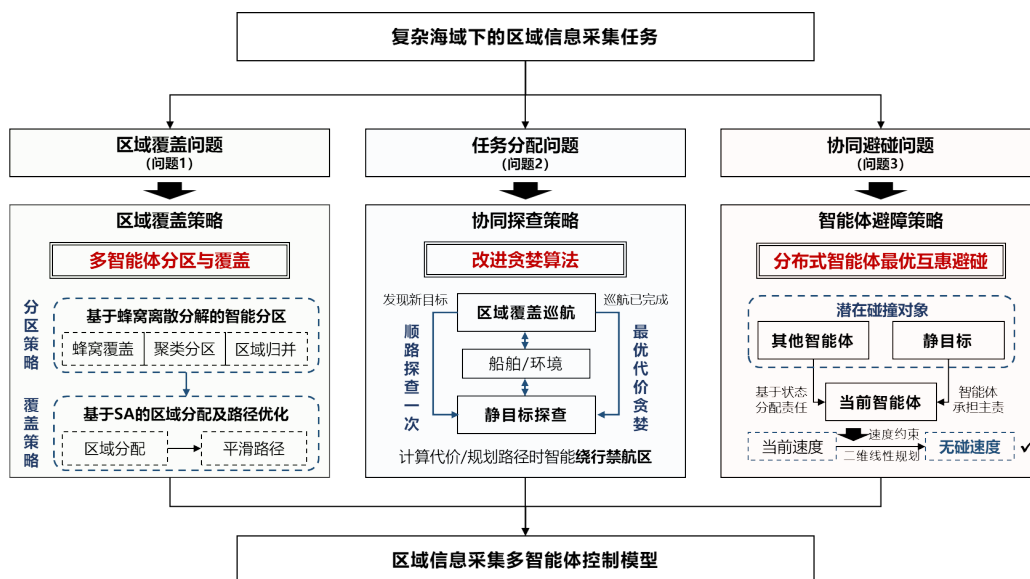


图3 复杂海域下多无人船覆盖探查系统总技术路线

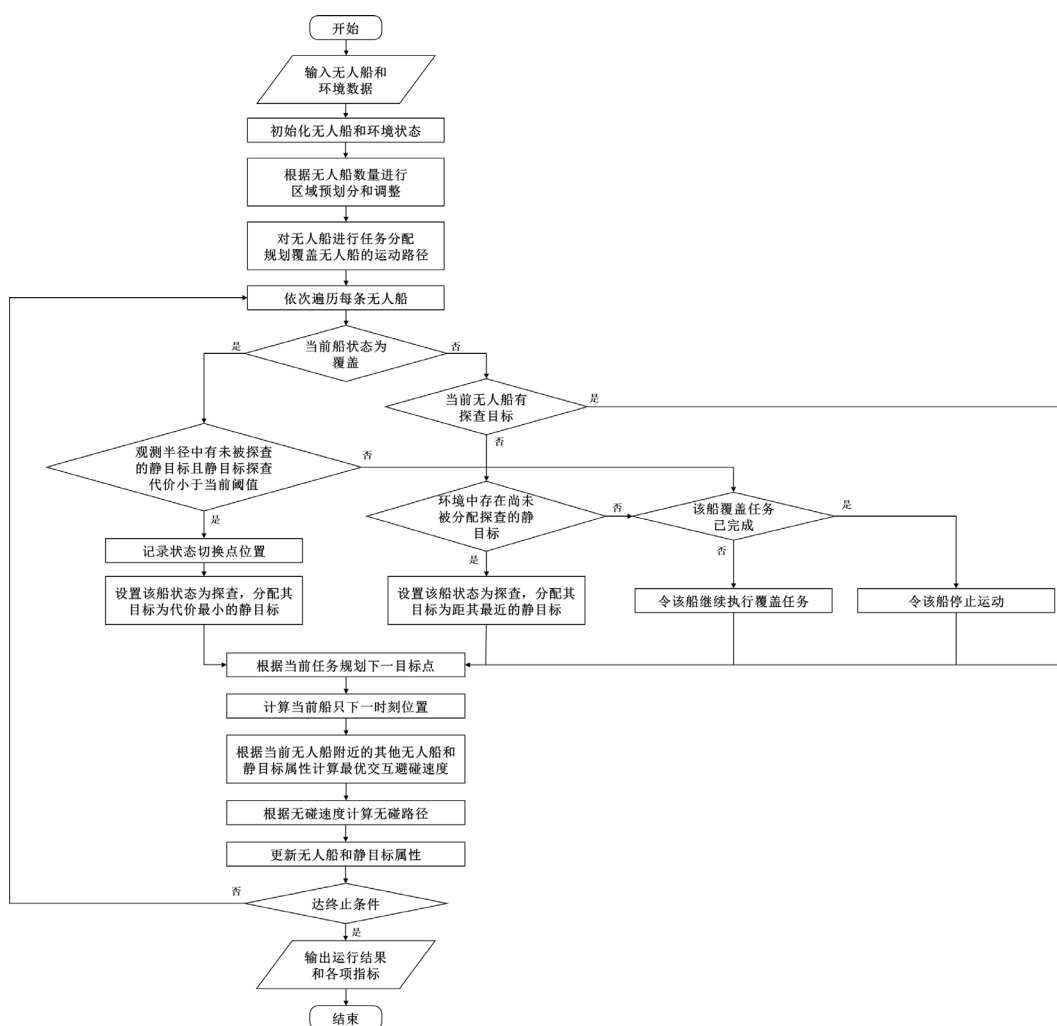


图4 模型算法流程图

3.2 区域覆盖策略

对于区域划分问题,采用了基于蜂窝离散分解的子区域划分方法。根据 2.3.1 节的分析,区域的覆盖可以用下式表示:

$$S = \bigcup_{i=1}^{N_A} S_i = \bigcup_{i=1}^{N_A} \bigcup_j D(P_i^j, r_{\text{detect}})。$$

于是,可以在任务区域中,确定一系列点,并以这些点以一定半径作圆,这样可以使得任务区域被一系列圆完全覆盖。文献[37]提出了一种最小圆的完全覆盖方法,相邻圆心之间的距离为 $\sqrt{3}r$,圆心相接后构成六边形网状结构。图 5 展示了最少圆覆盖矩形区域的示意图。

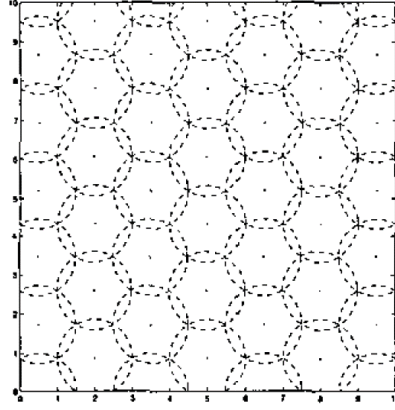


图 5 最少圆覆盖矩形区域

因此,利用此思想,可以将区域进行离散化,找出一系列点,并以这些点为圆心,以最小的智能体探测半径 $r = \min\{r_{\text{detect}}\}$ 作为半径作圆使得任务区域被覆盖,则只要所有的圆心被智能体经过,就可以保证任务区域被全覆盖。这样就可以将覆盖问题就能大幅简化,覆盖可以由“几何面积完全重叠”转化为“点序列的遍历”。由于圆的半径取最小的智能体探测半径,且将圆心连接构成路径后实际覆盖面积有覆盖冗余,因此在离散化时可以适当增大相邻圆心之间的距离进行蜂窝点离散,最大不超过 $2r$,如图 6 所示。

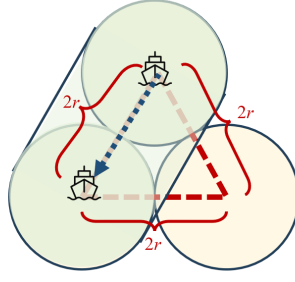


图6 蜂窝点离散化区域后覆盖示意图

考虑到边界处的圆心可能在界外，若将其忽视则会损失覆盖率。为了防止智能体出界或进入禁航区的同时保证覆盖率，在计算蜂窝点时对处于边界的点进行额外的增减。对凸多边形区域完全离散化如图7所示。

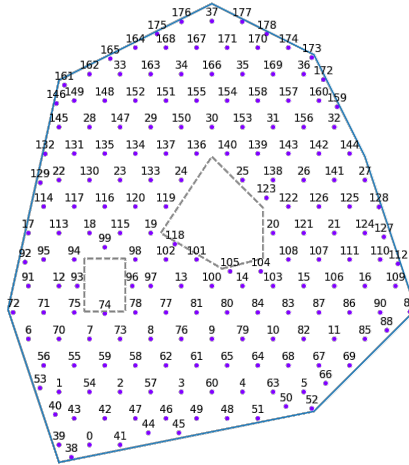


图7 凸多边形任务区域离散化示意图

对于覆盖路径问题，采用了基于模拟退火算法(SA)的最优区域分配及覆盖路径优化。考虑智能体的初始位置不同、速度不同，每个智能体需要负责覆盖的面积大小应该根据一定最优准则进行分配。于是，考虑将智能体从各自起点开始，遍历完所有蜂窝点的时间最短作为目标，使用启发式算法进行每个智能体的点分配和路径规划。

具体实现上，采用了模拟退火算法，将解定义为 N_A 个智能体的途径蜂窝点的序列，序列元素为各个蜂窝点的编号，每个序列的元素总数为对应智能体所要经过的蜂窝点，序列内元素的顺序表示路径上途径蜂窝点的先后顺序。通过 SA 优化，最终得到一个使目标值最小的解，即每个智能体的最优覆盖路径。

算法 1: 模拟退火算法 (SA)

- 1 初始化。设置初始温度 T_{start} 、终止温度 T_{end} 、温度冷却速率 α 。初始化当前解，并计算解的能量（代价）。
 - 2 根据当前解生成一个随机的邻近解，如交换任意两个序列元素的位置，并计算其能量。
 - 3 若邻近解能量 < 当前解能量，则接受并更新解：当前解 = 邻近解，当前能量 = 邻近能量；
若邻近解能量不小于当前解能量，也以一定概率接受差解并更新。
 - 4 降温，更新温度为当前温度 * 冷却速率 α
 - 5 重复 2-4 步，直到终止温度 T_{end} ，得到最终解。
-

这是一个近似多旅行商问题，复杂度随着问题规模的增加而呈指数级增长，在本问题的规模下，精确得出最优解十分非常困难。因此对问题进行进一步分解，分解为两步小规模 SA 优化，分解流程如下：

算法 2: 基于 SA 的最优区域分配及覆盖路径优化分步分解算法

- 1 对所有蜂窝点进行 k-means 聚类，聚类生成的每一簇的点构成一个子区域。
 - 2 第一步 SA 优化：
将所有子区域，使用 SA 算法分配给各个智能体，优化目标为：各个智能体遍历完所有子区域内的点的时间最小。得到的（近似）最优解为：各个智能体负责遍历的子区域。
 - 3 第二步 SA 优化：
对于智能体 A_i ，只考虑其负责的子区域的蜂窝点，使用 SA 算法生成一条路径点序列，使得遍历所有蜂窝点的时间最短。
 - 4 对于每个智能体使用第二步 SA 算法得到各自路径，最终所有智能体执行路径即可实现较优时间的全覆盖。
-

如图 8 所示，对所有蜂窝点进行聚类，每一个子区域内的点用同种颜色表示，可以使问

题规模（需考虑的点数量）从数百级别降低到只有数十级别。

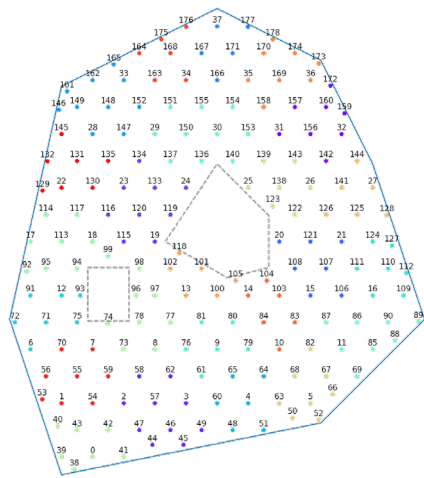


图 8 蜂窝点聚类结果

在第一步 SA 优化中，该步要求解的是每个智能体负责覆盖的子区域的分配，计算代价可以考虑每个智能体的速度异构和其实位置不同，最小化最长一个智能体完成覆盖负责区域所需要的时间，通过 SA 算法得到符合该 MIN-MAX 优化目标的智能体子区域分配的解。

在第二步 SA 优化中，由于分配问题已通过第一步 SA 优化解决，故仅需考虑单个智能体的具体路径问题，考虑现实船舶运行时不宜有过多的转弯，因此在 SA 算法的代价计算中，还添加了转弯惩罚因子。最终的路径能够满足转弯次数较少，路径更平滑，更具有工程现实意义。图 9 是各智能体最终路径规划的示意图。

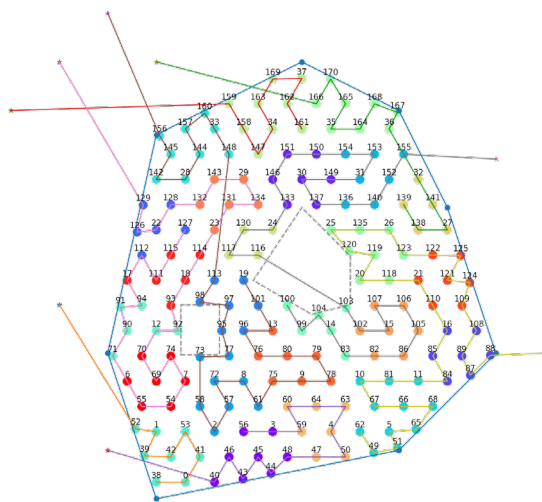


图 9 各智能体路径规划（初步规划，仅考虑点序）

若上述步骤规划的路径穿越了凸多边形组成的禁航区，则会根据几何方法对路径进行调整，使其绕开禁航区。具体流程如算法 2 所示。

算法 3：基于几何方法的禁航区绕行策略

输入 路径点集 P 、禁航区凸多边形区域集合 $F=(F_1, \dots, F_n)$

输出 绕行路径点集 P'

- 1 建立新的路径点集 $P'=P$;
 - 2 令对每一个凸多边形禁航区 $F_i \in F$:
 - 3 根据膨胀半径，将 F_i 膨胀为 F_i' ;
 - 4 对于每一个路径点 $p_j \in P'$ (不含终点):
 - 5 若 p_j 在 F_i' 内部，计算 p_j 和 p_{j+1} 构成线段与 F_i' 的最近交点 p_j' ; $p_j \leftarrow p_j'$;
 - 6 若终点在膨胀区内部，根据其与此前一点连线调整至多边形边界出;
 - 6 遍历 P' ，删除 F_i 内的所有内部点;
 - 7 对于每一个路径点 $p_j \in P'$ (不含终点):
 - 8 若 p_j 在 F_i' 外部且 p_{j+1} 在 F_i' 外部且 p_j 和 p_{j+1} 构成线段与 F_i' 相交:
分别 p_j 考虑按顺时针和逆时针绕过 F_i' 到达 p_{j+1} 的路径，取长度更短的一条作为 P' 。
-

图 10a 与图 10b 展示了利用本算法的路径调整结果。其中图 10a 考虑了原始规划路径点在膨胀区域内部的情形，图 10b 考虑了原始规划路径点在膨胀区域外部的情形。从图中可以看出，本算法面对实际工程场景中的临界情形（膨胀区域中出现航路点）以及多个禁航区的情形，均能正确处理，找到最佳绕行路线。

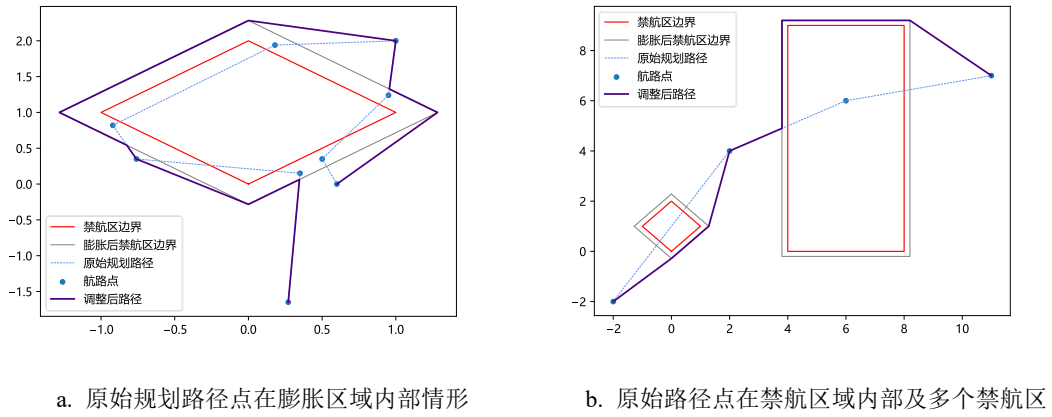


图 10 基于几何方法的禁航区绕行路径调整

3.3 协同探查策略

对于静目标的协同探查问题，将其分解为了智能体的内部决策及智能体间的调度两个问题。智能体的内部决策问题指的是：智能体在执行覆盖任务的同时需要在本地进行决策，决定是否需要就近完成观测到的静目标的探查任务。由于复审环节智能体探查同一静目标需要有时间间隔，因此在覆盖阶段每发现一个新静目标就对其进行一次探查是一种良好的策略，可以大幅降低探查阶段的工作量。智能体间的调度指的是：智能体完成自己的覆盖任务后，进入协同探查阶段，若此时区域内还存在未探查完成的静目标，中心服务器需要将智能体调度至相应的静目标点以完成探查任务。基于 2.3.2 节建立的协同探查多目标优化模型，提出了一种改进的贪婪算法来实现智能体间的调度，该方法是一种动态调度方法，能够为智能体实时动态地作出决策。

在探查阶段的改进贪婪算法中，中心服务器通过指派智能体前往探查代价最小的静目标点执行探查任务来实现智能体间的调度，探查代价的计算基于智能体和静目标的距离以及该目标剩余的间隔时间。具体而言，中心服务器会根据智能体和静目标之间的距离估算出前去探查该目标耗费的时间，并与该目标剩余间隔时间进行对比，取二者之间的较大值，遍历系统中所有已发现的静目标点，选择代价最小的静目标分配给该智能体前去探查。

该贪婪算法的“改进”之处在于，在算法中加入了“静目标抢夺”的部分。具体而言，中心服务器将某静目标 O 分配给智能体 A 后，会实时更新智能体 A 探查该静目标 O 的代价，在为其他智能体分配静目标时，若某智能体 A' 探查静目标 O 的代价远远小于智能体 A ，中心服务器会视情况调度智能体 A' 探查静目标 O ，并为智能体 A 分配新的静目标。改进贪

婪算法的具体流程如算法 3 所示。

算法 4: 改进贪婪探查算法

- 1 对于智能体 A_i ，若 A_i 已经完成覆盖任务：
 - 2 遍历系统中的所有静目标，计算 A_i 探查每个目标的代价，选择探查代价最小的静目标 O_x
 - 3 若 O_x 还需要的探查次数 m 小于选择探查 O_x 的智能体数量 n ：
 - 4 选择这 n 个智能体中当前探查代价最大的智能体 A_j ，若 A_j 还未开始探查动作，为其重新分配静目标
 - 5 令智能体 A_i 前去探查 O_x
 - 6 否则：
 - 7 令智能体 A_i 前去探查 O_x
-

3.4 智能体避障策略

对于智能体避障问题，采用了分布式的 ORCA 方法（Optimal Reciprocal Collision Avoidance，最优交互避碰）。将动态避碰问题转化为二次线性规划问题，在速度平面凸区域内求解，该算法是第一个能够保证大量智能体在一个杂乱的工作空间中的局部无碰撞运动的方法，但未考虑到环境中出现静态障碍物的问题。该模型将速度方向限定在一个平面内，只要智能体速度落到该半平面内，则可以避免碰撞。

根据其他智能体（例如 A_2 ）的位置、速度，智能体 A_1 规划得出速度障碍和避碰速度偏移向量。速度障碍、避碰速度偏移向量的数学表示如下：

$$\text{VO}_{A_1|A_2}^r = \{v \mid \exists t \in (0, \tau), tv \in D(p_2 - p_1, r_{\text{threat}})\},$$

$$u = \left(\arg \min_{v \in \partial \text{VO}_{A_1|A_2}^r} \left\| v - (v_1^- - v_2^-) \right\| \right) - (v_1^- - v_2^-)。$$

对于与任意智能体的会碰撞速度区域和偏移向量表示方式均同上。上式中， v^- 表示智能体原有速度， u 表示速度障碍 $\text{VO}_{A_1|A_2}^r$ 边界上的最小偏移向量，是智能体为了避障做出的最小速度变化。

记 \vec{n} 是速度障碍区域在避碰速度偏移向量点的外法向量。针对周围所有的障碍物（自身以外的智能体）的避障偏移向量，智能体 A_i 生成允许速度半平面，并对所有速度半平面做交集得到最后的可选速度集：

$$\text{ORCA}_{A_i}^r = \{v | (v - (v_i^- + \alpha \times u_i)) \cdot \vec{n}_i \geq 0, i = 2, 3, \dots, N_o\},$$

式中， α 表示责任分配系数，在智能体间避障时设置为 0.5，即避障双方两两分担相同的速度变化责任。上述 ORCA 避障算法的可视化表示如图 11。

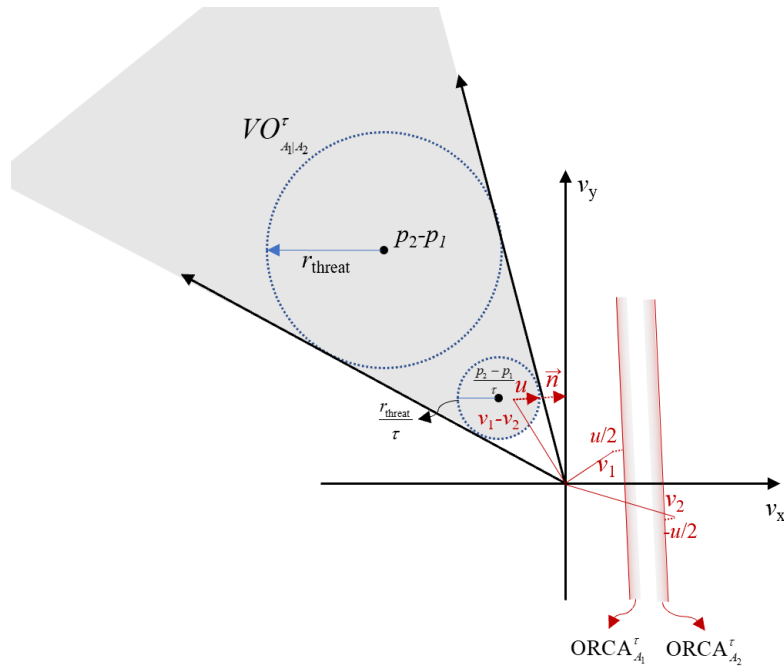


图 11 ORCA 避障算法对智能体 A_1, A_2 避障速度选取示意图

针对动态避障和静态避障两种情况，实现了分层的责任分配机制。对于静态障碍物采用同样的 ORCA 避障策略，但是避障责任分配由智能体承担，提高了智能体的避障灵活性。在最后的可选速度集内，智能体将选择与当前速度最接近的速度解，即为最终的碰撞避免速度。

当个体处在一个高密度的场景中时，其某个时刻的 ORCA 半平面交集很有可能是空集，此时通常会选择一个惩罚最小的速度作为新的个体速度。惩罚大小常选用速度到半平面分割线的欧几里得距离的最大值，智能体将选择该惩罚项值最小的速度。

4 模型的工程应用方法

4.1 模型的推广

本文所提出和采用的区域划分算法和覆盖算法可以扩展到针对不同外围边界、不同面积的二维作业海域以及异构的多智能体。本文算法利用蜂窝状六边形将整个二维任务区域覆盖，得出所有六边形的中心点集，并对任务边界区域的离散点分布进行调整，保证完全覆盖的同时尽可能降低耗时，适用于多种任务区域形状；将任务区域转换成若干离散点后，基于智能体的初始位置、覆盖半径和航行速度等条件，利用聚类的方法将离散点转换成若干子区域，再利用启发式模拟退火算法将子区域分配给各智能体；同时，考虑智能体实际运动学模型，对子区域内智能体的运动路径进行平滑优化，减少拐弯次数，减少实际船只的能量消耗，便于将模型引入实际应用。

本文所提出的改进贪婪协同探查算法可以尽可能保证智能体的负载平衡，减少任务完成时间。一方面，智能体在覆盖的过程中，每次发现静目标时立即对其探查，减少了探查阶段的任务数量；当智能体执行完被分配到的覆盖任务时，会进入协同探查阶段，中心服务器采用改进的贪婪算法，不断选择探查代价最小的静目标分配给该智能体。此外，中心服务器分配静目标后，实时更新该智能体探查目标所需要的代价，一旦发现前去探查该静目标的智能体数量多于静目标需要的探查次数，若其他智能体探查该目标的代价更小，中心服务器会对当前的分配结果进行调整，始终以最小代价实现对静目标的探查，减少了任务的完成时间，适用于各种未知场景，保证智能体的负载平衡。

本文使用的底层避碰算法 ORCA 计算量小，可以分布式地部署到无人船上。ORCA 算法是一种分布式的避碰算法，应用场景广泛。通过该算法，可以求解智能体在不同速度目标下（例如保持当前速度、速度尽可能大等）的最优避碰速度。该算法只需要知道智能体自身及存在潜在碰撞可能物体的位置、速度等信息，就可以利用线性规划方式在多项式时间进行求解。根据智能体执行任务时的不同状态，可以对智能体避障设定相应的责任，执行重要任务的智能体避障时承担较低的责任，在各种任务中进行责任的灵活分配，实现到一般模型的推广。在一般工程场景中，仅需智能体估计或测量本算法需要的位置和速度等参数，就可以直接对该算法进行分布式部署。

4.2 工程部署方法

实际工程中区域划分及覆盖算法的部署。实际工程中，有限的表面区域都可以近似抽象

为多边形包围的二维封闭图形，根据图形的特征，本文算法可以直接给出多智能体协同情形下的区域覆盖路径近似最优解；根据智能体的位置和能力，本文算法可以为各智能体分配相应的子区域并进行路径平滑。智能体只需要根据分配到的路径进行循迹，就可以完成海面区域的全覆盖，并尽量保证总覆盖时长最短。

实际工程改进贪婪协同探查算法的部署。实际工程中，智能体面临的决策情形与本文所考虑的简化场景十分相似。智能体在子区域覆盖阶段发现静目标时会短暂切换到探查阶段，仅需计算出能够探查到静目标的有效位置就可以做到两个任务之间的高效切换，且对全局任务完成情况的影响较小。在实际工程中，尤其是在智能体的覆盖阶段，智能体的轨迹并非与既定路线完全一致，这可能导致智能体完成任务的时间比预期的要滞后，通过第二阶段中心服务器智能体间的调度，可以使先完成任务的智能体帮助其余滞后智能体完成任务，使航速快、探测范围大的智能体帮助航速慢、探测范围小的智能体，实现高效、快速、准确的探查。即使出现智能体故障、智能体失联等情况，在有中心服务器的情况下本算法也可以保证尽可能完成更多的任务，这符合工程上的假设，部署也较为容易。

实际工程中分布式避碰算法的部署。实际工程中，需要考虑无人船运动的动力学约束。而本文所使用的 ORCA 避碰算法可以通过设置最优避碰速度为当前速度，来使得智能体规划出的新避碰速度尽可能离原速度最接近，这可以尽量满足无人船速度的动力学约束。本文算法对静目标和不同状态下的智能体的避障优先级进行多级设置，例如智能体总要避开静目标，执行覆盖任务的智能体总要避开执行探查任务的智能体，且该算法不仅适用于理想条件下物体的质点模型，也适用于物体具有一定体积的工程情形。另一方面，从通信的角度来看，本文使用的避碰算法为底层分布式算法，无需中心服务器调度。部署到实际工程时，可以根据任务具体情形，对执行避障任务的各角色优先级进行调整，障碍物或执行高精度任务的角色具有较高的避障优先级，执行低精度任务的角色具有较低的避障优先级。在海面，无人船可以通过雷达、超声波等传感器来估计和测量待避障目标（其他无人船、静目标）相应的速度和位置等参数，并根据相应的参数在本地求解出当前时刻的最佳避碰速度，实现算法到实际工程中的推广。

自测报告

1 计分方式

1.1 区域有效覆盖率

区域有效覆盖率=(智能体实际覆盖面积-智能体重叠覆盖面积)/区域面积。按照区域覆盖率 100%计 10 分, 80%计 3 分, 0%计 0 分, 构建线性分段函数计算得分。该指标不同值的得分如图 12。

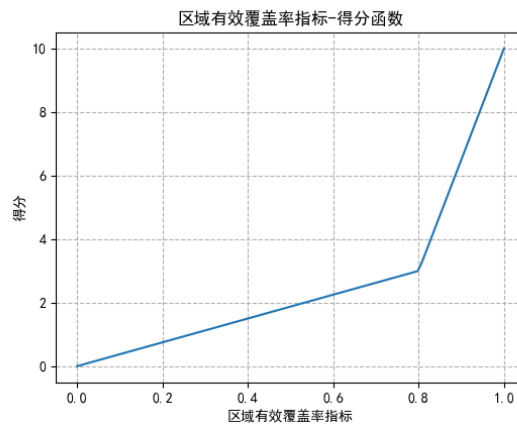


图 12 区域有效覆盖率指标不同值的得分

1.2 静目标有效探查指标

智能体进入静目标的探查范围计一次有效探查, 目标属性的待探查次数-1。静目标有效探查率=1-静目标剩余待探查次数/静目标总待探查次数=正确探查次数/静目标总待探查次数。静目标有效探查率 100%计 10 分, 80%计 6 分, 0%计 0 分, 构建线性分段函数计算得分。该指标不同值的得分如图 13 所示。

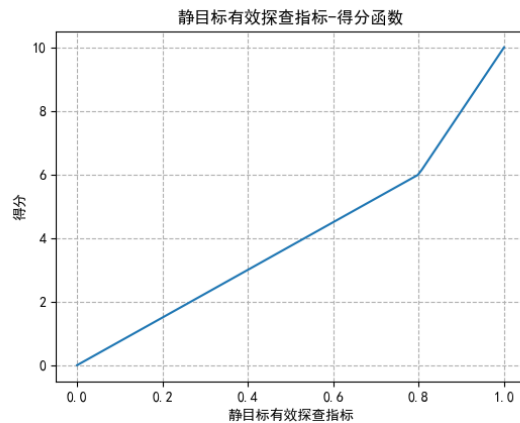


图 13 静目标有效探查指标不同值的得分

1.3 静目标正确探查指标

智能体按照指定角度完成目标抵近探查动作，计一次正确探查次数，静目标正确探查率 = 正确探查次数 / 探查执行总次数。静目标正确探查率 100% 计 10 分，0% 计 0 分，构建线性函数计算得分。该指标不同值的得分如图 14 所示。复赛中，根据实际工程中传感器安装位置与感知范围，设置正确探查的标准如图 15 所示，即当船舶处于静目标的探查扇形区域内、威胁区域外，且静目标处于船舶的双扇形区域内时，并且船舶满足探查速度要求，计一次正确探查。

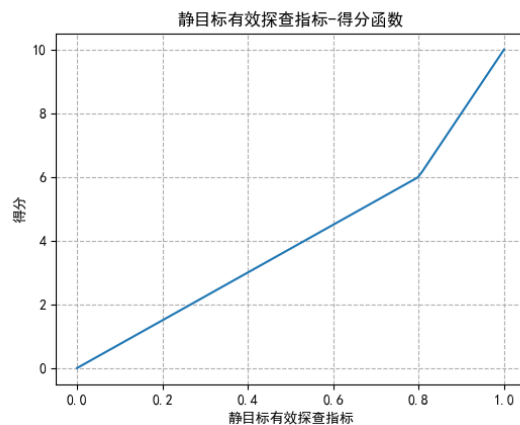


图 14 静目标有效探查指标不同值的得分

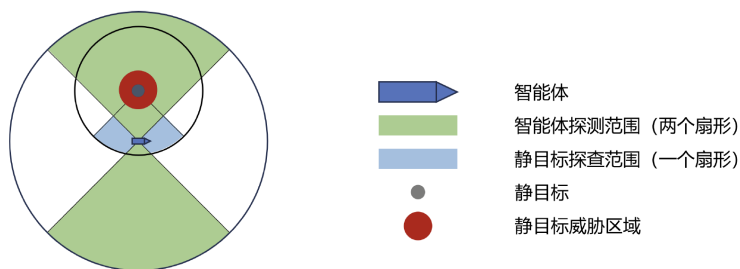


图 15 正确探查标准图

1.4 协同有效性指标

各智能体进入目标威胁区域或者其他智能体安全区域，计一次协同失败。按照协同失败总次数，构建线性分段函数计算最终得分，0 次计 10 分，5 次计 6 分，10 次及以上计 0 分。该指标不同值的得分如图 16。

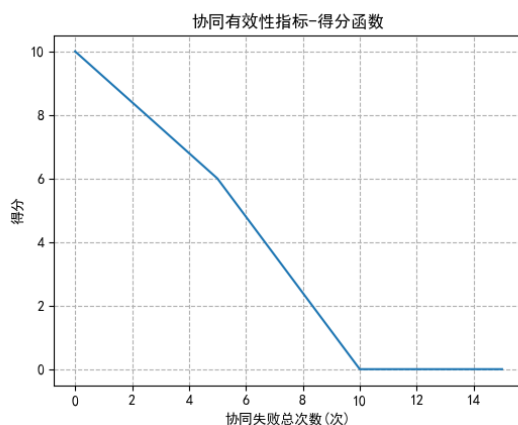


图 16 协同有效性指标不同值的得分

2 测试结果

将算法在银河麒麟 Kylin V10 SP1 系统上按照赛题规则执行仿真测试，并计算各项指标和得分情况。图 17 展示了算法的可视化运行界面，当前状态船舶已经协同完成区域的全覆盖与静目标的探查。图中，灰色多边形区域为目标海域；红色多边形区域表示禁航区；红、蓝圆圈组成的图形表示无人船舶，其中深蓝色双扇形区域表示探查范围，中心的红色圆形表示其威胁范围；红圆圈及蓝色扇形组成的图形表示静目标，其中深蓝色扇形表示其探查范围，红色圆形表示其威胁范围。

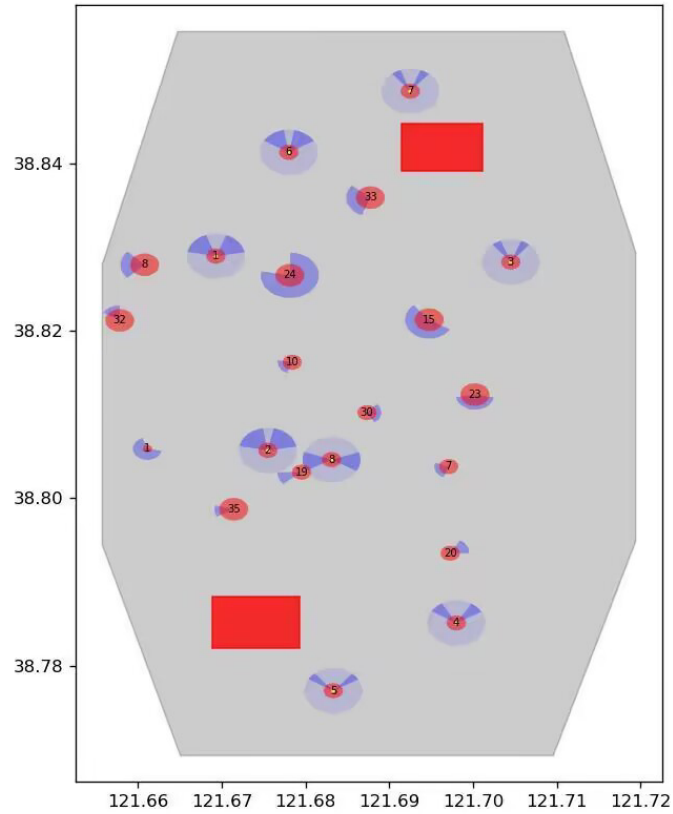


图 17 算法的可视化运行界面

2.1 区域有效覆盖率

经仿真测试，区域覆盖率为 99.67%，区域有效覆盖率指标得分为 9.94 分，由于本算法在蜂窝点基础上还对可能遗漏的边角区域增补了新的路径点，不应产生区域遗漏。分析认为原因如下：船舶受动力学约束，不能原地转向，在转弯时可能会产生一定的路径偏移，与规划路径不一致，导致区域覆遗漏一小部分区域。

2.2 静目标有效探查指标

有效探查是智能体正对目标完成探查动作，从探查范围外到探查范围的过程。经仿真测试，静目标总待探查次数为 136 次，执行正确的探查次数为 136 次。静目标有效探查率为 100%，静目标有效探查指标得分 10 分。

2.3 静目标正确探查指标

若有效探查满足时间、角度、速度的限制条件，才会记为正确探查。经仿真测试，探查执行总次数为 136 次，执行正确的探查次数为 136 次。静目标正确探查率为 100%，静目标正确探查指标得分 10 分。

2.4 协同有效性指标

在仿真测试过程中，智能体不会进入其他智能体或静目标的威胁范围，协同失败次数为 0。协同有效性指标得分 10 分。

2.5 全区域有效覆盖任务时长指标

经仿真测试，完成自测场景内全覆盖及所有探查仿真总时长为 85 分 27 秒。

2.6 小结

经过在银河麒麟系统上的仿真及上述分项计分，图 18 为各项指标及计分情况。

```
=====
                        结算阶段
=====
船舶数量：8
禁航区数量：2
静目标数量：35
-----
仿真总时长：85m 27s
共发现 35 / 35 个静目标
区域覆盖率：99.675%，覆盖率指标分：9.935 分
静目标有效探查 136/136 次，有效探查指标分：10.0 分
静目标正确探查 136/136 次，正确探查指标分：10.0 分
协同失败 0 次，协同有效性指标分：10.0 分
```

图 18 仿真结束时各指标及计分情况

附录：参考文献

- [1] Zhu M, Hahn A, Wen Y Q, et al. Maritime unmanned vehicle cruise path planning for maritime information collection[J]. IFAC-PapersOnLine, 2016, 49(23): 103-108.
- [2] Hazon N, Kaminka G A. Redundancy, efficiency and robustness in multi-robot coverage[C]//2005 IEEE International Conference on Robotics and Automation (ICRA). Barcelona, Spain: IEEE, 2005: 735-741.
- [3] Balch T. The case for randomized search[C]//2000 IEEE International Conference on Robotics and Automation (ICRA). San Francisco, CA. IEEE, 2000: 213-215.
- [4] Mitra R, Saha I. Scalable online coverage path planning for multi-robot systems[C]//2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Kyoto, Japan: IEEE, 2022: 10102-10109.
- [5] Tang Y, Zhou R, Sun G, et al. A novel cooperative path planning for multirobot persistent coverage in complex environments[J]. IEEE Sensors Journal, 2020, 20(8): 4485-4495.
- [6] Collins L, Ghassemi P, Esfahani E T, et al. Scalable coverage path planning of multi-robot teams for monitoring non-convex areas[C]//2021 IEEE International Conference on Robotics and Automation (ICRA). Xi'an, China: IEEE, 2021: 7393-7399.
- [7] Li M, Richards A, Sooriyabandara M. Asynchronous reliability-aware multi-uav coverage path planning[C]//2021 IEEE International Conference on Robotics and Automation (ICRA). Xi'an China: IEEE, 2021: 10023-10029.
- [8] Gao L, Lv W, Yan X, et al. Complete coverage path planning algorithm based on energy compensation and obstacle vectorization[J]. Expert Systems with Applications, 2022, 203: 117495.
- [9] Vandermeulen I, Groß R, Kolling A. Turn-minimizing multirobot coverage[C]//2019 IEEE International Conference on Robotics and Automation (ICRA). Montreal, QC, Canada: IEEE, 2019: 1014-1020.

- [10] Gosrich W, Mayya S, Li R, et al. Coverage control in multi-robot systems via graph neural networks[C]//2022 IEEE International Conference on Robotics and Automation (ICRA). Philadelphia, PA, USA: IEEE, 2022: 8787-8793.
- [11] Chen J, Du C, Zhang Y, et al. A clustering-based coverage path planning method for autonomous heterogeneous uavs[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(12): 25546-25556.
- [12] Chen J, Ling F, Zhang Y, et al. Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system[J]. Swarm and Evolutionary Computation, 2022, 69: 101005.
- [13] Xie J, Chen J. Multiregional coverage path planning for multiple energy constrained uavs[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(10): 17366-17381.
- [14] Jiao L, Peng Z, Xi L, et al. Multi-agent coverage path planning via proximity interaction and cooperation[J]. IEEE Sensors Journal, 2022, 22(6): 6196-6207.
- [15] Jing W, Deng D, Wu Y, et al. Multi-uav coverage path planning for the inspection of large and complex structures[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas, NV, USA: IEEE, 2020: 1480-1486.
- [16] Nakamura K, Santos M, Leonard N E. Decentralized learning with limited communications for multi-robot coverage of unknown spatial fields[C]//2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Kyoto, Japan: IEEE, 2022: 9980-9986.
- [17] 田微. 基于动态粒子蜜蜂算法的群机器人任务分配方法研究[D]. 吉林大学, 2017.
- [18] Zhu W, Li L, Teng L, et al. Multi-uav reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding[J]. Chinese Journal of Aeronautics, 2018, 31(2): 339-350.
- [19] Zlot R, Stentz A, Dias M B, et al. Multi-robot exploration controlled by a market economy[C]//2002 IEEE International Conference on Robotics and Automation (ICRA). Washington, DC, USA: IEEE, 2002, 3: 3016-3023.

- [20] Ponda S S, Johnson L B, Kopeikin A N, et al. Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams[J]. *IEEE Journal on Selected Areas in Communications*, 2012, 30(5):861-869.
- [21] Lee D H, Zaheer S A, Kim J H. A resource-oriented, decentralized auction algorithm for multirobot task allocation[J]. *IEEE Transactions on Automation Science and Engineering*, 2015, 12(4):1469-1481.
- [22] Luo L, Chakraborty N, Sycara K. Distributed algorithms for multirobot task, assignment with task deadline constraints[J]. *IEEE Transactions on Automation Science and Engineering*, 2015, 12(3):876-888.
- [23] Neville G, Chernova S, Ravichandar H. D-itags: a dynamic interleaved approach to resilient task allocation, scheduling, and motion planning[J]. *IEEE Robotics and Automation Letters*, 2023, 8(2): 1037-1044.
- [24] Luo X, Zavlanos M M. Temporal logic task allocation in heterogeneous multirobot systems[J]. *IEEE Transactions on Robotics*, 2022, 38(6): 3602-3621.
- [25] Al-Hussaini S, Gregory J M, Gupta S K. Generating task reallocation suggestions to handle contingencies in human-supervised multi-robot missions[J]. *IEEE Transactions on Automation Science and Engineering*, 2023, early access: 1-15.
- [26] Bai X, Fielbaum A, Kronmüller M, et al. Group-based distributed auction algorithms for multi-robot task assignment[J]. *IEEE Transactions on Automation Science and Engineering*, 2022, 20(2): 1292-1303.
- [27] Nie Z, Chen K C. Hypergraphical real-time multirobot task allocation in a smart factory[J]. *IEEE Transactions on Industrial Informatics*, 2021, 18(9): 6047-6056.
- [28] Park S, Zhong Y D, Leonard N E. Multi-robot task allocation games in dynamically changing environments[C]//2021 IEEE International Conference on Robotics and Automation (ICRA). Xi'an, China: IEEE, 2021: 8678-8684.
- [29] Gilbert E, Johnson D. Distance functions and their application to robot path planning in the presence of obstacles[J]. *IEEE Journal on Robotics and Automation*, 1985, 1(1): 21-30.
- [30] Lazarowska A. Discrete artificial potential field approach to mobile robot path planning[J]. *IFAC-PapersOnLine*, 2019, 52(8): 277-282.

- [31] Van den Berg J, Guy S J, Lin M, et al. Reciprocal n-body collision avoidance[C]//Robotics Research: The 14th International Symposium ISRR. Berlin, Heidelberg: Springer, 2011: 3-19.
- [32] Fulgenzi C, Spalanzani A, Laugier C. Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid[C]//2007 IEEE International Conference on Robotics and Automation (ICRA). Rome, Italy: IEEE, 2007: 1610-1616.
- [33] Pfeiffer M, Schaeuble M, Nieto J, et al. From perception to decision: a data-driven approach to end-to-end motion planning for autonomous ground robots[C]//2017 IEEE International Conference on Robotics and Automation (ICRA). Singapore: IEEE, 2017: 1527-1533.
- [34] Sui Z, Pu Z, Yi J, et al. Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(6): 2358-2372.
- [35] Shi H, Shi L, Xu M, et al. End-to-end navigation strategy with deep reinforcement learning for mobile robots[J]. IEEE Transactions on Industrial Informatics, 2020, 16(4): 2393-2402.
- [36] Salt L, Howard D, Indiveri G, et al. Parameter optimization and learning in a spiking neural network for uav obstacle avoidance targeting neuromorphic processors[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(9): 3305-3318.
- [37] Guo Y, Qu Z. Coverage control for a mobile robot patrolling a dynamic and uncertain environment[C]//Fifth World Congress on Intelligent Control and Automation. Hangzhou, China: IEEE, 2004, 6: 4899-4903.