

无人飞行器集群自主控制: 基于联盟形成博弈的任务分配

姜斌¹ 马亚杰¹ 薛舒心¹

摘要 针对复杂多约束条件下异构无人飞行器集群问题, 提出一种基于联盟形成博弈的分布式任务预分配和重分配方法. 考虑时效性、同时性等耦合约束条件, 引入准确的能耗模型建立任务分配模型, 利用联盟形成博弈将任务分配问题转化为联盟划分问题, 并设计一种无故障条件下的分布式任务预分配方法, 降低任务分配求解的复杂度, 同时提高最终解的平均质量; 进一步, 针对无人机故障问题, 准确分析健康无人机的运动模型, 合理划分重分配范围, 基于任务预分配结果设计重分配算法. 仿真结果证明了所提分布式任务预分配与重分配方法在不同场景下的实时性和有效性.

关键词 联盟形成博弈, 异构无人机集群, 任务分配, 重分配, 无人机能耗

引用格式 姜斌, 马亚杰, 薛舒心. 无人飞行器集群自主控制: 基于联盟形成博弈的任务分配. 自动化学报, 2025, 51(5): 942–959

DOI 10.16383/j.aas.c240593 **CSTR** 32138.14.j.aas.c240593

Autonomous Control of Unmanned Aerial Vehicle Swarms: Task Allocation Based on Coalition Formation Game

JIANG Bin¹ MA Ya-Jie¹ XUE Shu-Xin¹

Abstract This paper presents a distributed task preallocation and reallocation method for heterogeneous unmanned aerial vehicle (UAV) swarms system based on coalition formation game to solve the task allocation problem under complex and multi-constraint conditions. Considering multiple coupling constraints such as timeliness and synchronization, a task allocation model is established and an accurate energy consumption model is introduced to it. The task allocation problem is transformed into a coalition partitioning problem based on coalition formation game. Then a distributed task preallocation method under fault-free conditions with low complexity is designed, which can improve the average quality of the final solution. Furthermore, in response to the occurrence of UAV malfunctions, the motion model of healthy drones is accurately analyzed and the reallocation range is reasonably divided. Then a task reallocation algorithm is proposed based on the preallocation results. The simulation results prove the real-time performance and effectiveness of the proposed distributed task preallocation and reallocation method in different scenarios.

Key words Coalition formation game, heterogeneous UAV swarm, task allocation, reallocation, energy consumption of UAV

Citation Jiang Bin, Ma Ya-Jie, Xue Shu-Xin. Autonomous control of unmanned aerial vehicle swarms: Task allocation based on coalition formation game. *Acta Automatica Sinica*, 2025, 51(5): 942–959

收稿日期 2024-08-26 录用日期 2024-11-11

Manuscript received August 26, 2024; accepted November 11, 2024

国家自然科学基金 (62273177, 62020106003, 62233009), 江苏省自然科学基金 (BK20211566, BK20222012), 教育部高校学科创新引智基地 (B20007), 空间智能控制技术全国重点实验室开放基金 (HTKJ2023KL502006), 中央高校基础科研业务费 (NI2024001) 资助

Supported by National Natural Science Foundation of China (62273177, 62020106003, 62233009), Natural Science Foundation of Jiangsu Province of China (BK20211566, BK20222012), Programme of Introducing Talents of Discipline to Universities of China (B20007), Open Foundation of Science and Technology on Space Intelligent Control Laboratory (HTKJ2023KL502006), and Fundamental Research Funds for the Central Universities (NI2024001)

本文责任编辑 李鸿一

Recommended by Associate Editor LI Hong-Yi

1. 南京航空航天大学自动化学院 南京 210016

1. College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016

无人机 (Unmanned aerial vehicle, UAV), 也称为无人飞行器, 作为新一代人工智能技术下的代表性飞行器, 具有低成本、小体积、高机动性等特点, 已广泛应用于军事^[1]、科研^[2]、商业^[3]等领域. 随着自动化科学和人工智能技术的快速发展, 无人机向着集群化、异构化、自主化方向发展^[4]. 从“马赛克战”等先进作战概念可以看出, 异构无人机集群协同执行任务是一个极具发展潜力的应用方向. 相比于单无人机, 无人机集群系统通过个体之间的信息融合和能力互补, 能够形成多样化的任务能力以及差异性的性能优势和载荷携带, 大大提升执行任务的效率和鲁棒性^[5]. 而无人机集群协同完成任务的前提是有效的任务分配^[6], 将具有不同需求的任务分配给合适的无人机, 可以在能耗、任务效用等方

面充分发挥良好的整体效能^[7]. 无人机集群系统任务分配是决定无人机集群发展的关键技术之一, 具有非常重要的研究意义和应用价值.

日益复杂的工作环境、多样化的任务类型以及不断提高的无人机智能化水平^[8]给无人机集群系统任务分配带来新的挑战. 在异构特性方面, 无人机性能各异, 集群从单一同构发展到复杂异构, 需要考虑无人机类型和任务之间的对齐; 在集群规模方面, 从传统编队的几架无人机发展到几十架甚至成千上万架, 给集群无人机系统任务分配^[9]带来新的挑战. 此外, 恶劣的工作环境和突发事件的发生会导致无人机故障, 改变无人机任务能力和集群数量, 因此任务分配需要对环境态势迅速作出反应, 在有限时间内完成任务的重构和优化, 达成新的无冲突的任务重分配方案. 面对以上需求, 相对于集中式算法, 基于并行、协同、通信和多智能体策略的分布式算法能够赋予无人机一定的自主性和决策权, 可拓展性强且更加灵活, 对提升优化速度和鲁棒性起到重要作用^[10-11].

常见的分布式任务分配算法主要包括基于市场机制的方法、基于分布式马尔科夫决策过程的方法和基于博弈论的方法. 基于市场机制的方法主要有合同网算法和拍卖算法, 合同网包含发布者和竞标者, 并将分配过程分为招标、投标、中标、确认四个阶段. 文献 [12] 提出一种改进的基于混合合同网协议的动态任务调度方法, 包括买卖、交换、取代合同三个状态. 改进后的算法更加灵活, 但未考虑无人机的异构特性. 文献 [13] 针对通用无线系统的速率分配问题, 在网络端信息不完整的情况下, 提出一种新颖的拍卖机制, 设计的核心是一个能够用于确定正确的干扰价格的迭代过程, 所提方法能够有效地进行速率分配任务. 文献 [14] 建立一种面向异构且具有时序约束任务的多轮次分布式拍卖算法, 能够较为灵活地解决具备不同能力无人机的分布式协同任务分配问题, 在静态分配中能得到较好的分配结果.

基于分布式马尔科夫决策过程的分配方法通常使用强化学习进行训练得到最优分配解, 但该方法训练难度较大、收敛较慢. 文献 [15] 研究存在环境不确定性的异构无人机动态任务分配问题, 将任务分配问题建模为马尔科夫决策过程, 构造基于强化学习算法的发送请求方法和基于 Q-learning 算法的接受请求方法, 具有较好的可拓展性. 文献 [16] 建立基于马尔科夫决策过程的多智能体系统任务分配模型, 提出一种基于多智能体深度强化学习的任务分配算法, 能够解决状态空间大、复杂度高的任务分配问题, 但需要较高的计算成本.

传统任务分配方法难以保证复杂约束下异构无人机集群系统的任务预分配和故障后任务重分配的实时性需求. 基于博弈论的方法能够为无人机之间的合作协商提供良好性能, 近年来已成为热门研究方向, 应用于各种优化问题的求解^[17]. 博弈论是解决分布式系统资源优化问题的有效工具, 可将玩家之间的交互视为博弈, 通过计算博弈的平衡状态来寻找最优任务分配策略^[18]. 文献 [19] 提出一种新的博弈论决策框架来解决多智能体任务分配问题, 设计能够在多项式时间内收敛到纳什稳定划分的去中心化算法, 具有可扩展性能, 对动态环境具有快速适应性. 文献 [20] 设计一种分布式偏好联盟博弈算法来解决多智能体任务分配问题, 智能体通过一种包含优先级和一致阶段的分布式偏好联盟博弈算法进行决策, 以获得纳什均衡解.

联盟结构是一种高效的多无人机协同执行任务的网络架构^[21], 其中联盟形成博弈 (Coalition formation game, CFG) 通过利用无人机之间的相互合作最大化系统性能, 为参与者之间的决策提供一个高效的数学模型和工具. 文献 [22] 提出一种基于联盟形成博弈的无人机干扰群系统动态任务分配算法, 可以获得与集中式算法相似的性能且适用于动态环境. 文献 [23] 提出一种顺序重叠联盟形成博弈, 解决承载不同资源的无人机协同执行任务的任务分配问题, 设计一种偏好重力引导禁忌搜索模型, 该模型改进了联盟形成博弈模型只优化无人机在单个联盟中组成的问题. 文献 [24] 利用联盟形成博弈解决多智能体之间的相互作用问题, 提出的优化算法能够在有限次迭代内收敛到纳什均衡解, 且能够满足每个任务对智能体数量的最低要求约束. 因此, 将无人机集群系统任务分配问题建模为联盟形成博弈问题, 能够灵活地解决不同场景的任务分配问题, 具有较广的适用范围和较好的分配效果.

在任务执行过程中, 无人机故障的发生带来任务重分配的需求, 而重分配算法的设计可以以预分配算法作为基础. 如文献 [7] 针对智能体在执行任务过程中突发故障问题, 将无故障时的任务分配算法推广到故障情况下的任务重分配, 实现较低的任务重分配复杂度. 为满足不同场景下的动态重分配需求, 文献 [25] 针对环境属性的变化问题, 将预分配算法扩展到动态紧急调整场景, 设计三种任务重分配策略: 完全重分配、局部调整和分组重分配; 文献 [26] 建立多无人机系统遭遇故障时的局部重规划问题模型, 将故障等级进行分类后, 利用基于收益动态调整规则和一致性协调规则的拍卖算法解决不同情况的任务重分配问题; 文献 [27] 建立多无人机协同侦查任务预分配和重分配模型, 设计一

种基于合同网协议的局部任务重分配算法,能够处理由多种动态事件引起的任务重分配问题.以上任务重分配方法虽然在一定程度上解决了多无人机系统故障问题,但并未考虑无人机的资源异构特性和复杂的多重约束条件.

为解决异构无人机集群系统的任务分配和故障下的任务重分配问题,本文提出一种基于联盟形成博弈的异构无人机集群任务预分配和重分配算法,能够在多重约束下将异构需求的任务分配给携带异构资源的无人机.主要贡献总结如下:

1) 建立多约束多优化目标的任务分配数学模型,引入准确的能耗模型,更具有现实意义和应用价值;基于联盟形成博弈算法将任务分配问题转化为联盟划分问题,利用势博弈证明了纳什均衡解的存在性,为后续算法的设计提供模型基础.

2) 提出一种基于联盟形成博弈的分布式任务预分配算法,设计一种新的联盟偏好关系决策规则,旨在提高整体联盟收益;同时加入扰动优化增加算法的搜索范围,避免陷入局部最优,提高最终解的质量.

3) 建立任意无人机在任意时刻发生故障的任务重分配框架,设计基于预分配结果进行局部调整的任务重分配算法,提高任务重分配的求解速度.

1 问题描述与数学建模

考虑将 m 项任务分配给 n 个异构无人机的任务分配问题.假设每个任务包含多种子任务类型,且对子任务的资源需求数目不同.设定异构无人机携带不同数量的任务资源,对应其可以执行不同子任务类型.在本文的任务分配问题中,异构无人机集群协作执行任务,分配完成同一个任务的所有无人机称为一个联盟.任务分配问题包含无人机无故障情况下的任务预分配与发生故障后的任务重分配.

1.1 任务预分配建模

首先是对无人机无故障情况下的任务预分配问题的数学建模.

1.1.1 无人机与任务建模

无人机集群中有 n 个无人机,集合表示为 $U = \{u_1, u_2, \dots, u_n\}$.利用四元组 UIN 来描绘无人机状态信息:

$$UIN(Res, position, v_{max}, a) \quad (1)$$

$$UIN = \{UIN_{u_i} | i = 1, 2, \dots, n\} = \{Res_{u_i}, position_{u_i}, v_{max_{u_i}}, a_{u_i} | i = 1, 2, \dots, n\} \quad (2)$$

其中, $Res_{u_i} = \{Re_{u_i}^{1c}, Re_{u_i}^{2c}, \dots, Re_{u_i}^{lc}, Re_{u_i}^{1r}, \dots, Re_{u_i}^{kr}\}$ 表示无人机可具备执行 l 种消耗资源类任务和 k 种非消耗资源类任务的能力, c 和 r 分别表示消耗资源类和非消耗资源类任务.消耗资源类是指无人机执行类如攻击、救援等任务,可将能力的差异量化为无人机 u_i 携带的第 p 种消耗型资源的数目.非消耗资源类是类如信息采集等任务,无人机资源不会随着任务的执行而减少.本文将无人机 u_i 是否具有执行第 q 种非消耗资源类任务的能力建模为 0 或 1,即若无人机 u_i 具有执行第 q 种子任务的能力,则 $Re_{u_i}^{qr} = 1$, 否则 $Re_{u_i}^{qr} = 0$. $position_{u_i} = (x_{u_i}, y_{u_i}, z_{u_i})$ 表示无人机 u_i 的位置, $v_{max_{u_i}}$ 和 a_{u_i} 分别表示无人机 u_i 的最大速度和恒定加速度大小.

问题场景中共有 m 个任务,任务集合表示为 $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$.每个任务包含不同类型的子任务,任务状态信息四元组 TIN 描述为

$$TIN(Res, position, TW, t^{con}) \quad (3)$$

$$TIN = \{TIN_{T_j} | j = 1, 2, \dots, m\} = \{Res_{T_j}, position_{T_j}, TW_{T_j}, t_{T_j}^{con} | j = 1, 2, \dots, m\} \quad (4)$$

其中, $Res_{T_j} = \{Re_{T_j}^{1c}, \dots, Re_{T_j}^{lc}, Re_{T_j}^{1r}, \dots, Re_{T_j}^{kr}\}$, $Re_{T_j}^{pc}$ 表示任务 T_j 所需的第 p 种消耗型资源的数目, $Re_{T_j}^{qr}$ 表示其第 q 种非消耗资源类任务的工作量,如任务的音频数据大小等. $position_{T_j} = (x_{T_j}, y_{T_j}, z_{T_j})$ 表示任务 T_j 的位置信息. $t_{T_j}^{con}$ 是 T_j 总消耗资源类任务的所需固定执行时间,假设消耗资源类任务和非消耗资源类任务为递进关系依次进行,任务执行总时长为两种类型任务执行时间之和.

时间窗主要分为三种,分别为硬时间窗、软时间窗与混合时间窗.硬时间窗是指对无人机到达任务时间范围提出硬性要求,若晚于某个时间节点,任务失败;软时间窗则指对无人机到达任务时间范围提出弹性要求,但不同时间段具有不同惩罚因子.本文将任务建模为具有混合时间窗,即设置时间窗信息 $TW_{T_j} = \{t_{T_j}^{minstart}, t_{T_j}^{maxstart}, \lambda\}$, $t_{T_j}^{minstart}$ 是任务 T_j 出现的时间,该节点后任务 T_j 处于可执行状态; $t_{T_j}^{maxstart}$ 是任务 T_j 最晚开始的时间,超过该节点任务 T_j 无效,因此任务分配问题具有时效性约束. λ 是在该时间窗口内任务等待时间造成收益衰减的比例系数.

本文引入同步性约束,其是任务分配问题中的常见约束,即联盟内最晚抵达任务的无人机 u_i 的到达时间 t_{u_i, T_j} 和 $t_{T_j}^{minstart}$ 之间的最大值视为该任务开始执行的时间 $t_{T_j}^{start}$.

图 1 是本文任务分配问题中无人机执行任务的

示意图, 无人机信息的第 1 行表示机载消耗型资源的数量, 第 2 行表示无人机是否具有执行各类非消耗资源类任务的能力. 如图中 u_2 携带 3 个单位的第 1 种消耗型资源, 具有执行第 2 种和第 3 种非消耗资源类任务的能力. 同样, 对应任务的第 1 行和第 2 行分别表示其消耗型资源需求数目和非消耗资源类任务的工作量. 图中任务 T_1 对第 1 种消耗型资源的需求量为 5 个单位, 其第 1 类非消耗资源类任务的工作量为 3 个单位.

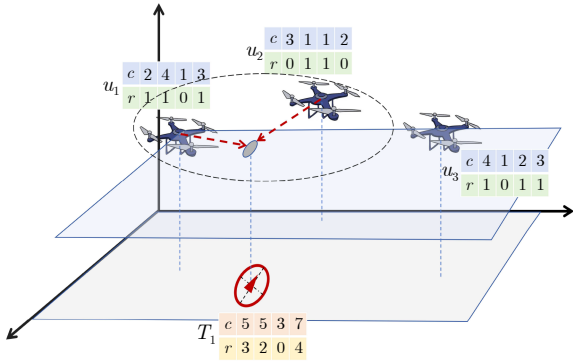


图 1 无人机执行任务示意图

Fig.1 Sketch map of UAVs perform task

考虑到分配任务为空集的无人机, 将任务集合拓展为 $\mathcal{T} = \{T_0, T_1, T_2, \dots, T_m\}$, T_0 表示无人机不执行任务. 将无人机的任务分配向量表示为 $E = \{e_{u_1}, e_{u_2}, \dots, e_{u_n}\}$, 其中, e_{u_i} 表示无人机 u_i 选择的任务. 完成同一个任务的无人机称为一个联盟, 定义 $c_{T_j} = \{e_{u_i} = T_j | u_i \in U\}$ 表示执行任务 T_j 的无人机联盟, $c_{T_j} \subseteq U$ 且 $\cup_{T_j \in \mathcal{T}} c_{T_j} = U$. $CS = \{c_{T_0}, \dots, c_{T_m}\}$ 是任务分配方案对应的联盟结构.

图 2 是本文任务分配结果示意图, 无人机集群通过任务分配算法分配到各任务对应的联盟中. 图

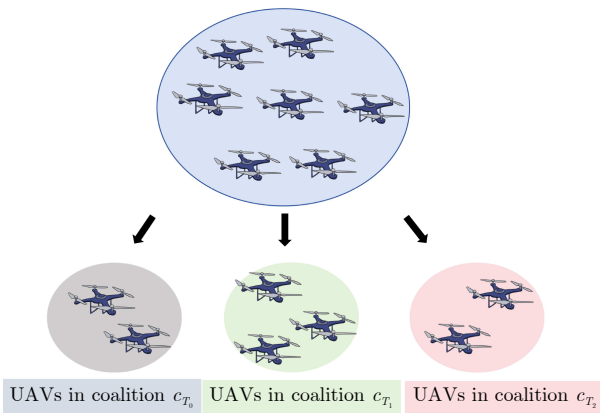


图 2 任务分配结果示意图

Fig.2 Sketch map of task allocation result

中联盟 c_{T_0} 中有两个无人机, 即在该任务分配结果中这两个无人机不执行任务; 联盟 c_{T_1} 中有三个无人机, 即在该结果中这三个无人机执行任务 T_1 , 同样可以此类推分析任务分配结果.

至此, 完成对任务分配问题中的无人机和任务以及它们之间对应关系的建模, 接下来是对无人机的能耗建模.

1.1.2 能耗模型

无人机的能耗主要包含通信能耗和推进能耗^[28], 通信能耗比推进能耗小几个数量级, 因而通常忽略通信能耗. 无人机的推进能耗主要与飞行速度和加速度有关. 本文采用 Fly-hover-communicate 协议^[29], 指无人机从初始位置出发以固定高度飞至任务上方, 在其上方进行悬停以执行任务, 如图 1 所示. 在本模型中, 忽略无人机到达任务高度前后变化消耗的能量, 无人机执行任务的推进能耗包括直飞、悬停两部分消耗的能量.

1) 直飞能耗

对于速度大小为 V 的无人机, 推进功率消耗可建模为^[28]

$$P(V) = P_0 \left(1 + \frac{3V^2}{U_{tip}^2} \right) + P_i \left(\sqrt{1 + \frac{V^4}{4v_0^4}} - \frac{V^2}{2v_0^2} \right)^{\frac{1}{2}} + \frac{1}{2} d_0 \rho s A V^3 \quad (5)$$

其中, P_0 和 P_i 是两个常数, 分别表示悬停状态下的叶片轮廓功率和诱导功率; U_{tip} 表示转子叶片的叶尖速度; v_0 为悬停时的平均转子诱导速度; d_0 和 s 分别为机身阻力比和转子实度; ρ 和 A 分别表示空气密度和转子盘面积.

无人机初始速度为 0, 具有恒定加速度大小 a 和最大速度 v_{max} . 无人机速度非常量, 因此需要引用关于转子推力 T 的推进功率消耗模型:

$$P'(V, \varepsilon) = P_0 \left(1 + \frac{3V^2}{U_{tip}^2} \right) + \frac{1}{2} d_0 \rho s A V^3 + P_i \varepsilon \left(\sqrt{\varepsilon^2 + \frac{\rho^2 A^2 V^4}{W^2}} - \frac{\rho A V^2}{W} \right)^{\frac{1}{2}} \quad (6)$$

式中, W 表示无人机的重力且

$$\varepsilon \Leftrightarrow \frac{T}{W} = \left(1 + \frac{(\rho S_{FP} V^2 + 2ma)^2}{4W^2} \right)^{\frac{1}{2}} \quad (7)$$

其中, S_{FP} 和 m 分别是机身等效平面面积和无人机的重量, 因此可以得到:

$$\begin{aligned}
P(v(t), a) &= P_0(1 + k_1 v^2(t)) + k_5 v^3(t) + \\
&P_i \sqrt{1 + (k_2 v^2(t) + k_3 a)^2} \times \\
&\left(\sqrt{1 + (k_2 v^2(t) + k_3 a)^2 + k_4^2 v^4(t) - k_4 v^2(t)} \right)^{\frac{1}{2}} \\
k_1 &= \frac{3}{U_{tip}^2}, k_2 = \frac{\rho S_{FP}}{2W}, k_3 = \frac{m}{W} \\
k_4 &= \frac{\rho A}{W}, k_5 = \frac{1}{2} d_0 \rho s A
\end{aligned} \quad (8)$$

根据无人机初始位置和任务的距离大小, 有两种运动模式, 过程中对应的速度大小如图 3 所示.

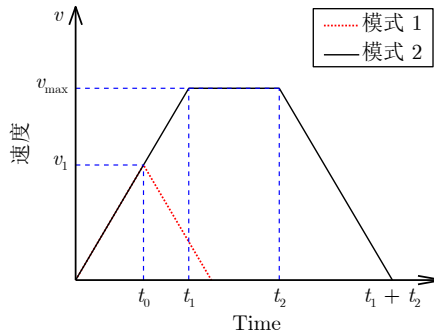


图 3 预分配时无人机运动模式示意图

Fig.3 The motion mode of UAVs during preallocation

根据式 (5) 和式 (8) 分析两种飞行模式的能耗:

a) 当无人机和任务的距离 d 小于等于 v_{\max}^2/a , 无人机飞行速度遵循模式 1: 首先从 0 加速到速度 v_1 , 之后匀速减速至 0, 飞行时间为 $2t_0$:

$$v_1^2 = ad \Rightarrow v_1 = \sqrt{ad} \quad (9)$$

$$t_0 = \frac{\sqrt{ad}}{a} \quad (10)$$

在 $[0, t_0]$ 段, 直飞能耗计算式为

$$\begin{aligned}
E_f &= \int_0^{t_0} \left[P_0(1 + k_1 a^2 t^2) + k_5 a^3 t^3 + \right. \\
&P_i \sqrt{1 + (k_2 a^2 t^2 + k_3 a)^2} \times \\
&\left. \left(\sqrt{1 + (k_2 a^2 t^2 + k_3 a)^2 + k_4^2 a^4 t^4 - k_4 a^2 t^2} \right)^{\frac{1}{2}} \right] dt
\end{aligned} \quad (11)$$

b) 当无人机和任务的距离 d 大于 v_{\max}^2/a , 无人机飞行速度遵循模式 2: 首先从 0 加速到速度 v_{\max} , 保持匀速飞行一段时间后减速至 0, 飞行时间为 $t_1 + t_2$:

$$t_1 = \frac{v_{\max}}{a} \quad (12)$$

$$(t_2 - t_1)v_{\max} + \frac{v_{\max}^2}{a} = d \Rightarrow t_2 = \frac{d}{v_{\max}} \quad (13)$$

在 $[t_1, t_2]$ 段, 直飞能耗计算式为

$$\begin{aligned}
E_f &= P(v_{\max})(t_2 - t_1) = \left[P_0(1 + k_1 v_{\max}^2) + \right. \\
&P_i \left(\sqrt{1 + \frac{v_{\max}^4}{4v_0^4} - \frac{v_{\max}^2}{2v_0^2}} \right)^{\frac{1}{2}} + \\
&\left. k_5 v_{\max}^3 \right] (t_2 - t_1)
\end{aligned} \quad (14)$$

因此可在不同的运动模式下利用上述分段计算式计算无人机的直飞能耗.

2) 悬停能耗

当无人机保持悬停状态时, 飞行速度为 0, 无人机的悬停功率 P_{hov} 为

$$P_{\text{hov}} = P(v = 0) = P_0 + P_i \quad (15)$$

这是一个仅取决于飞机的重量、空气密度以及旋翼面积等因素的常量. 无人机的悬停时间取决于到达任务后的等待时间 $t_{u_i, T_j}^{\text{wait}}$ 与任务的执行时间 $t_{T_j}^{\text{con}} + t_{T_j}^{\text{mcon}}$. 由于任务具有时间窗约束和同时性约束, 无人机需要等到联盟内最晚无人机到达且任务出现时, 任务才开始执行, 因此无人机 u_i 等待时间的计算为

$$t_{u_i, T_j}^{\text{wait}} = t_{T_j}^{\text{start}} - t_{u_i, T_j} \quad (16)$$

其中, t_{u_i, T_j} 是无人机 u_i 到达任务 T_j 的时间. 当无人机执行消耗资源类任务时, 所需固定任务执行时间为 $t_{T_j}^{\text{con}}$; 当无人机执行类似采集数据的非消耗资源类任务时, 任务执行时间计算如下.

根据文献 [30] 中的地对空路径损耗模型, 无人机 u_i 与任务 T_j 之间的平均路径损耗表示为

$$L_{u_i, T_j} = P_{u_i, T_j}^{\text{LoS}} L_{u_i, T_j}^{\text{LoS}} + (1 - P_{u_i, T_j}^{\text{LoS}}) L_{u_i, T_j}^{\text{NLoS}} \quad (17)$$

$L_{u_i, T_j}^{\text{LoS}}$ 和 $L_{u_i, T_j}^{\text{NLoS}}$ 分别表示无人机 u_i 和任务 T_j 在视距链路和非视距链路下的路径损耗, 计算为

$$\begin{cases} L_{u_i, T_j}^{\text{LoS}} = \eta_1 \left(\frac{4\pi f_c h_{u_i, T_j}}{c_0} \right)^2, & \text{LoS} \\ L_{u_i, T_j}^{\text{NLoS}} = \eta_2 \left(\frac{4\pi f_c h_{u_i, T_j}}{c_0} \right)^2, & \text{NLoS} \end{cases} \quad (18)$$

其中, f_c 是载波频率, c_0 是光速, η_1 和 η_2 是视距链路和非视距链路下的额外路径损耗系数, h_{u_i, T_j} 是无人机 u_i 和任务 T_j 之间的高度差. 视距链路概率计算式为

$$P_{u_i, T_j}^{\text{LoS}} = \frac{1}{1 + \alpha(\exp(-\beta(90 - \alpha)))} \quad (19)$$

其中, α 和 β 是取决于载波频率和环境类型的常数

值. 在本文中, 假设无人机与任务的传输信道由视距链路主导^[31], 可得

$$P_{u_i, T_j}^{LoS} = 1 \quad (20)$$

$$L_{u_i, T_j} = L_{u_i, T_j}^{LoS} = \eta_1 \left(\frac{4\pi f_c h_{u_i, T_j}}{c} \right)^2 \quad (21)$$

无人机 u_i 和任务 T_j 之间的传输速率为

$$R_{u_i, T_j} = B \log_2 \left(1 + \frac{p_{T_j}}{L_{u_i, T_j} \sigma^2} \right) \quad (22)$$

其中, B 为带宽, p_{T_j} 是任务 T_j 的发射功率, σ^2 是噪声功率. 任务分配结果中执行任务 T_j 第 q 种非消耗资源类子任务的无人机与任务的总传输速率为

$$R_{c_{T_j}, T_j}^{qr} = \sum_{u_i \in c_{T_j}, Re_{u_i}^{qr}=1} R_{u_i, T_j} \quad (23)$$

可以得到该子任务的持续时长为

$$t_{T_j}^{qr} = \frac{Re_{T_j}^{qr}}{R_{c_{T_j}, T_j}^{qr}} \quad (24)$$

则任务 T_j 的非消耗资源类任务总执行时间定义为

$$t_{T_j}^{ucon} = \max \left(t_{T_j}^{1r}, t_{T_j}^{2r}, \dots, t_{T_j}^{kr} \right) \quad (25)$$

悬停能耗记为

$$E_{hov}^{u_i, T_j} = P_{hov} \times \left(t_{u_i, T_j}^{wait} + t_{T_j}^{con} + t_{T_j}^{ucon} \right) \quad (26)$$

根据设定, 初始无人机集群 U 高度相同, 无人机飞行到任务上方执行任务且始终保持高度一致. 能够执行同种非消耗资源类任务的无人机与同一任务之间具有相同的传输速率, 因此非消耗资源类任务持续时间取决于执行任务的无人机数量和其工作量.

至此, 完成本文的能耗建模. 接下来考虑无人机能耗的任务预分配问题的目标函数与约束条件的建模.

1.1.3 目标函数与约束条件

本文任务效用设计考虑以下元素: 联盟内机载资源和任务所需资源的重叠关系、时间窗口内等待时间造成的衰减以及联盟内无人机的总推进能耗. 效用的设计旨在提高任务的完成度, 同时冗余资源和能耗的考虑确保最优联盟内无人机数量的有限性. 综上, 将预分配时任务 T_j 效用定义为

$$R(T_j) = -w \sum_{u_i \in c_{T_j}} (E_f^{u_i, T_j} + E_{hov}^{u_i, T_j}) + (K(j, :)I(:, j) - P \times O) \exp(\lambda(t_{T_j}^{start} - t_{T_j}^{minstart})) \quad (27)$$

其中, K 和 P 分别是权重矩阵和常数权值, 权衡资

源在全局效用中的比重. $K(j, :) = [k_{j1}, k_{j2}, \dots, k_{jl}]$ 表示对任务 T_j 而言各消耗型资源的重要程度.

对于权重矩阵 K , 可灵活设置其元素大小. 当对任务 t_j 而言各异构资源类型无具体优先级区分时, 此时向量内所有元素值为 k_m/l , k_m 表示总的可用资源占比权重. 当资源优先级取决于任务所需资源数量时, 此时权重值正比于任务所需该资源数目, 如某个任务所需攻击资源远远大于其他资源数量, 即对攻击类无人机需求更为强烈, 可增大其权重使携带攻击资源较多的无人机具有更高的优先级. 总之, 权重矩阵的设计使异构资源优先级有所区分, 从而贴合任务实际需求.

$I(:, j) = [i_{T_j}^{1c}, i_{T_j}^{2c}, \dots, i_{T_j}^{lc}]^T$ 表示联盟中可利用的每类资源的数目, 有

$$i_{T_j}^{pc} = \begin{cases} \sum_{u_i \in c_{T_j}} Re_{u_i}^{pc}, & 0 \leq \sum_{u_i \in c_{T_j}} Re_{u_i}^{pc} < Re_{T_j}^{pc} \\ Re_{T_j}^{pc}, & \sum_{u_i \in c_{T_j}} Re_{u_i}^{pc} \geq Re_{T_j}^{pc} \end{cases} \quad (28)$$

O 指联盟内总冗余资源数目为

$$O(T_j) = \sum_{p \in \{1, 2, \dots, l\}} \left(\max \left(\sum_{u_i \in c_{T_j}} Re_{u_i}^{pc} - Re_{T_j}^{pc}, 0 \right) \right) \quad (29)$$

w 是能耗的权重, 决定能耗在效用函数中的占比.

因此, 本文异构无人机集群任务预分配问题目标函数可建模为

$$\max \sum_{T_j \in \mathcal{T}} R(T_j) \quad (30)$$

约束条件有

$$t_{T_j}^{minstart} \leq t_{T_j}^{start} < t_{T_j}^{maxstart}, T_j \in \mathcal{T} \quad (31)$$

$$t_{T_j}^{start} = \max \{ t_{u_i, T_j}, t_{T_j}^{minstart} \}, u_i \in c_{T_j} \quad (32)$$

$$c_{T_j} \cap c_{T_i} = \emptyset, T_j, T_i \in \mathcal{T}, j \neq i \quad (33)$$

$$\sum_{u_i \in c_{T_j}} Re_{u_i}^{qr} > 0, \text{ 若 } Re_{T_j}^{qr} > 0 \\ T_j \in \mathcal{T}, q \in \{1, 2, \dots, k\} \quad (34)$$

式 (31) 是时效性约束, 指任务开始执行时间应满足时间窗约束; 式 (32) 是同步性约束, 联盟内所有无人机的到达视为任务开始的必要条件; 式 (33) 是指无人机的任务模式约束, 即无人机最多加入一个联盟且不同联盟之间交集为空集; 式 (34) 是指当任务的非消耗资源类子任务工作量不为 0 时, 联盟

内至少有一个具备执行该子任务能力的无人机。

上述是任务预分配问题描述与完整建模,可以看出与其他最优化问题相同,由于问题本身的组合特性和搜索空间复杂而庞大,获得最优联盟解显然是个 NP-hard 问题。

在得到预分配结果后,无人机在执行任务过程中无法避免发生故障的可能性.假设故障无人机不再具备完成任务的能力,因而存在任务的重分配需求.接下来是针对任务重分配的数学建模.

1.2 任务重分配建模

考虑无人机在执行任务过程中发生不可逆故障导致无法继续执行任务,此时为保证任务的完成质量,需要实时调整分配策略.为降低运算量和提高重分配的实时性,利用无人机与任务之间的欧氏距离大小粗略代替准确的直飞能耗的计算.因此,异构无人机集群中存在无人机发生故障后的任务重分配问题可建模为

$$\begin{aligned} & \max \sum_{T_j \in \mathcal{T}} R'(T_j) \\ & R'(T_j) = \left[- \sum_{u_i \in c_{T_j}} (w_1 \text{dis}(u_i, T_j) + w_2 E_{\text{hov}}^{u_i, T_j}) + \right. \\ & \left. (K(j, :)I(:, j) - P \times O) \exp(\lambda(t_{T_j}^{\text{start}} - t_{T_j}^{\text{minstart}})) \right] \\ & \text{s.t.} \\ & t_{T_j}^{\text{minstart}} \leq t_{T_j}^{\text{start}} < t_{T_j}^{\text{maxstart}}, T_j \in \mathcal{T} \\ & t_{T_j}^{\text{start}} = \max\{t_{u_i, T_j}, t_{T_j}^{\text{minstart}}\}, u_i \in c_{T_j} \\ & c_{T_j} \cap c_{T_i} = \emptyset, T_j, T_i \in \mathcal{T}, j \neq i \\ & \sum_{u_i \in c_{T_j}} R e_{u_i}^{q_r} > 0, \text{ 若 } R e_{T_j}^{q_r} > 0 \\ & T_j \in \mathcal{T}, q \in \{1, 2, \dots, k\} \end{aligned} \quad (35)$$

可以看出任务重分配问题的约束条件与任务预分配约束条件相同,主要区别在于对联盟效用的计算.

图 4 是当无人机出现故障后的任务重分配示意图,此时,随着联盟内故障无人机的退出,新的无人机加入联盟,继续执行任务.

相比于图 1,图 4 是当无人机 u_2 发生故障后基于图 1 中的预分配结果进行调整后的任务重分配示意图.此时 u_2 退出任务的执行,为继续完成任务,无人机 u_3 加入任务 T_1 对应的联盟中,与 u_1 继续协同完成任务 T_1 .

针对上述任务重分配需求,本文目标在于设计基于预分配结果的任务重分配算法,追求在较短时

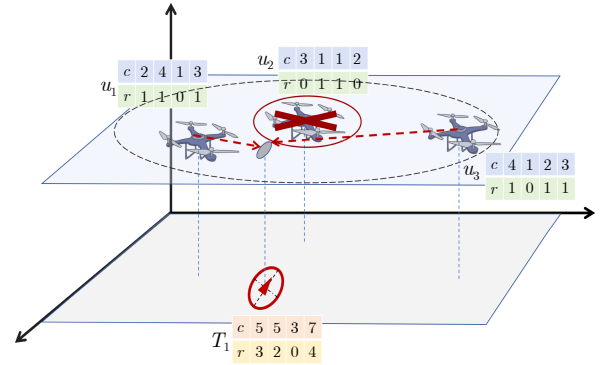


图 4 故障发生后任务重分配示意图

Fig.4 Sketch map of task reallocation after faults

间内得到新的无冲突的解.

2 联盟形成博弈建模

根据上述任务分配数学模型,本节旨在建立联盟形成博弈模型,将任务分配问题转化为联盟划分问题,为后续算法的设计提供理论基础.

2.1 博弈模型构建

定义联盟形成博弈模型 $\mathcal{G} = \{U, R, E, \varepsilon\}$. U 是任务分配问题中的无人机集合,表示博弈玩家; R 是效用函数,能够评估每个联盟的收益; $E = \{E_{u_1}, E_{u_2}, \dots, E_{u_n}\}$ 是博弈中每个无人机的可选择的策略集合, $E = \mathcal{T}$; ε 是博弈玩家的效用函数.为与联盟总效用保持一致,玩家效用函数设计如下:

$$\begin{aligned} \varepsilon_{u_i}(e_{u_i}, E_{-u_i}) &= R(c_{u_i}) - R(c_{u_i} \setminus \{u_i\}), \\ \forall u_i \in U, e_{u_i} \in E_{u_i} \end{aligned} \quad (36)$$

其中, e_{u_i} 表示无人机 u_i 的策略, E_{-u_i} 是其他无人机的策略, c_{u_i} 是任务 e_{u_i} 对应的联盟, $c_{u_i} \setminus \{u_i\}$ 是指将 u_i 从原属联盟中删除后的新的无人机联盟.

至此,基于联盟形成博弈的模型已建立,图 5 是基本元素博弈流程图.每个无人机选择任务组成联盟结构,根据联盟效用函数和玩家效用函数得到每个联盟和玩家的效用,之后按照一定的规则调整策略选择.

联盟偏好关系决策规则是影响博弈性能的重要因素,通过比较博弈玩家对不同联盟的偏好来指导无人机转换其所在联盟以提高整体效用.不同的决策规则会导致最终联盟结构不同.定义 \succ_{u_i} 表示无人机 u_i 的偏好, $c_{T_p} \succ_{u_i} c_{T_q}$ 表示相比较于加入联盟 c_{T_q} , u_i 有更高的意愿加入联盟 c_{T_p} .帕累托顺序是常见的用以决定联盟偏好关系的规则.

定义 1 (帕累托顺序 (Pareto order)^[23]).对于无人机 $u_i \in U$ 且 $e_{u_i} = T_q$,帕累托顺序指当且仅当

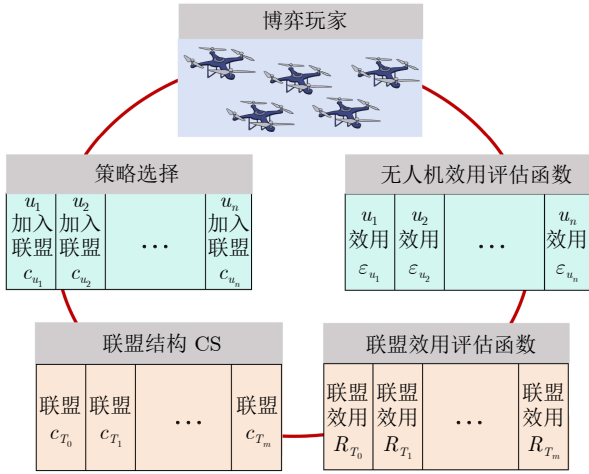


图 5 任务分配问题基本元素博弈流程图

Fig. 5 Flowchart of game of basic element in task allocation problem

满足

$$\varepsilon_{u_i}(T_p, E_{-u_i}) \geq \varepsilon_{u_i}(T_q, E_{-u_i}) \quad (37)$$

且 $\forall u_j \in (c_{T_p}, c_{T_q})$, 满足

$$\varepsilon_{u_j}(c_{T_p} \cup \{u_i\}, c_{T_q} \setminus \{u_i\}) \geq \varepsilon_{u_j}(c_{T_p}, c_{T_q}) \quad (38)$$

的条件时, 有 $c_{T_p} \succ_{u_i} c_{T_q}$.

按照帕累托顺序, 无人机当且仅当能够提高原属联盟和新联盟中的所有无人机效用时, 才会考虑转换联盟. 这种规则保证每次转换联盟都会提高整体效用, 然而由于转换联盟条件过于严苛, 容易得到质量较差的局部最优解. 本文基于式 (27) 联盟总效用和式 (36) 无人机效用设计一种新的适用于本任务分配问题的联盟偏好关系决策规则:

$$c_{T_p} \succ_{u_i} c_{T_q} \Leftrightarrow$$

$$R(c_{T_p} \cup \{u_i\}) + R(c_{T_q} \setminus \{u_i\}) + \sum_{T_j \in \mathcal{T} \setminus \{T_p, T_q\}} R(c_{T_j}) >$$

$$R(c_{T_p}) + R(c_{T_q}) + \sum_{T_j \in \mathcal{T} \setminus \{T_p, T_q\}} R(c_{T_j}) \Rightarrow$$

$$R(c_{T_p} \cup \{u_i\}) - R(c_{T_p}) > R(c_{T_q}) - R(c_{T_q} \setminus \{u_i\}) \Rightarrow \varepsilon_{u_i}(e_{T_p}, E_{-u_i}) > \varepsilon_{u_i}(e_{T_q}, E_{-u_i}) \quad (39)$$

即无人机转换联盟能够提高自身效用也意味着能够提高整体收益时, 转换联盟是可行的. 因此可以围绕这一原则设计基于联盟形成博弈的分布式任务分配算法. 本文博弈的目标是获得纳什均衡解 CS^* , 其定义如下.

定义 2 (纳什均衡 (Nash equilibrium))^[32]. 当没有玩家可以通过转移联盟来提高自身收益时, 此时的联盟结构 CS^* 即为纳什均衡的. 即

$$\varepsilon_{u_i}(e_{u_i}^*, E_{-u_i}^*) > \varepsilon_{u_i}(e_{-u_i}, E_{-u_i}^*), \quad \forall u_i \in U, e_{-u_i} \in E_{u_i} \quad (40)$$

$e_{u_i}^*$ 和 $E_{-u_i}^*$ 分别是 u_i 和其他无人机在纳什均衡的联盟结构 CS^* 下的策略选择, e_{-u_i} 是 E_{u_i} 内除 $e_{u_i}^*$ 外的其他任意策略选择.

接下来是博弈模型纳什均衡解存在性的证明.

2.2 纳什均衡解的分析

势博弈是用以证明纳什均衡解存在的有效工具, 其定义和性质如下.

定义 3 (势博弈 (Potential game))^[22]. 若博弈存在一个势函数 P , 当无人机改变策略引起的效用函数的变化能够等量地反映在该势函数上时, 即

$$\varepsilon_{u_i}(e_{u_i}, E_{-u_i}) - \varepsilon_{u_i}(\bar{e}_{u_i}, E_{-u_i}) = P(e_{u_i}, E_{-u_i}) - P(\bar{e}_{u_i}, E_{-u_i}), \quad \forall u_i \in U, \forall e_{u_i}, \bar{e}_{u_i} \in E_U \quad (41)$$

则该博弈称为势博弈.

为证明所设计的基于联盟形成博弈的无人机任务分配算法的效果, 本文使用文献 [7] 给出的以下两个势博弈的性质.

性质 1. 势博弈必然存在至少一个纯纳什均衡解且能够使势函数最大的解为纯纳什均衡解之一.

性质 2. 势博弈具有有效递增属性, 任何单方面改进的序列可在有效时间内收敛到纳什均衡状态.

定理 1. 本文博弈模型 $\mathcal{G} = \{U, R, E, \varepsilon\}$ 为势博弈, 且博弈模型的纳什均衡解存在.

证明. 将联盟总效用之和设为势函数, 即

$$P(e_{u_i}, E_{-u_i}) = \sum_{c_{T_j} \in CS} R(c_{T_j}) \quad (42)$$

当无人机 u_i 策略由 e_{u_i} 改为 \bar{e}_{u_i} 且其他无人机策略保持不变时,

$$P(e_{u_i}, E_{-u_i}) - P(\bar{e}_{u_i}, E_{-u_i}) = \left[R(c_{e_{u_i}}) + R(c_{\bar{e}_{u_i}}) + \sum_{c_m \in CS_\Delta} R(c_m) \right] - \left[R(c_{\bar{e}_{u_i}}) + R(\bar{c}_{\bar{e}_{u_i}}) + \sum_{c_m \in CS_\Delta} R(c_m) \right] \quad (43)$$

$CS_\Delta = CS \setminus \{c_{e_{u_i}}, c_{\bar{e}_{u_i}}\}$, 是指不包含任务 e_{u_i} 和 \bar{e}_{u_i} 的所有任务联盟, 在策略转换中始终保持不变. 无人机 u_i 策略的变化在联盟结构中的表示为

$$\bar{c}_{e_{u_i}} = c_{e_{u_i}} \setminus \{u_i\} \quad (44)$$

$$c_{\bar{e}_{u_i}} = \bar{c}_{\bar{e}_{u_i}} \cup \{u_i\} \quad (45)$$

可得

$$\begin{aligned} P(e_{u_i}, E_{-u_i}) - P(\bar{e}_{u_i}, E_{-u_i}) = \\ (R(c_{e_{u_i}}) - R(\bar{c}_{e_{u_i}})) - (R(\bar{c}_{\bar{e}_{u_i}}) - R(c_{\bar{e}_{u_i}})) = \\ \varepsilon_{u_i}(e_{u_i}, E_{-u_i}) - \varepsilon_{u_i}(\bar{e}_{u_i}, E_{-u_i}) \end{aligned} \quad (46)$$

因此, 存在势函数能够等量反映单个无人机策略的变化. 根据定义 2, 可将本博弈模型构造为势博弈, $\sum_{c_{T_j} \in CS} R(c_{T_j})$ 为势函数.

由于势博弈性质 1 的特性, 本文博弈模型的纳什均衡解必然存在. 因此可基于已建立的理论基础设计合理的联盟形成算法实现最终的任务分配. 性质 2 可用于证明求解纳什均衡解算法的收敛性. \square

3 基于联盟形成博弈的任务分配算法设计

针对已建立的博弈模型, 本节提出一种基于联盟形成博弈的异构无人机集群系统分布式任务分配算法, 算法的目标是获得纳什均衡的联盟结构. 本文任务分配需求产生于两个阶段: 初始无故障情况下的任务预分配和任务执行过程中故障情况下的任务重分配. 因此本节分别提出一种初始环境下的预分配和故障情况下的重分配算法, 并给出算法收敛性证明.

3.1 基于联盟形成博弈的任务预分配算法设计

首先是针对初始无故障情况下的任务预分配算法设计. 任务预分配问题的目标函数与约束条件如式 (30) ~ (34) 所示, 本部分旨在设计能够满足约束条件且最大化目标函数的分布式算法.

3.1.1 预分配算法与收敛证明

首先, 本文设计的基于联盟形成博弈的任务预分配算法的第 1 阶段是初始化任务分配方案. 由于时间窗口的约束限制, 为保证任务的成功执行和降低任务分配组合数量, 初始化时无人机只能加入能够在任务截止时间前到达的任务或选择不加入任务, 其表示为

$$E_{u_i} = \{T_j | T_j \in \mathcal{T}, t_{u_i, T_j} < t_{T_j}^{\max\text{start}}\} \cup \{T_0\} \quad (47)$$

当完成初始化后, 算法的第 2 阶段是无人机根据上述联盟偏好关系决策规则进行任务选择的博弈. 当无人机在新联盟中的自身效用高于在原联盟中的效用时, 意味着转换联盟能够提高整体收益, 此时转换联盟是可行的. 在第 l 次迭代中随机选择一个无人机 u_i , $u_i \in U$, 计算其此时的策略选择下的收益:

$$\varepsilon_{u_i}(e_{u_i}^{l-1}, E_{-u_i}^{l-1}) = R(c_{u_i}^{l-1}) - R(c_{u_i}^{l-1} \setminus \{u_i\}) \quad (48)$$

其中, $e_{u_i}^{l-1}$ 是无人机在上一次迭代后的策略选择, 对应的联盟记为 $c_{u_i}^{l-1}$, $E_{-u_i}^{l-1}$ 是除 u_i 外的其他无人机的策略集合. 随机选择 E_{u_i} 中的除 $e_{u_i}^{l-1}$ 外的任务 $\bar{e}_{u_i}^l$, 计算更新策略后的无人机收益:

$$\varepsilon_{u_i}(\bar{e}_{u_i}^l, E_{-u_i}^{l-1}) = R(c_{\bar{e}_{u_i}^l}^{l-1} \cup \{u_i\}) - R(c_{\bar{e}_{u_i}^l}^{l-1}) \quad (49)$$

$c_{\bar{e}_{u_i}^l}^{l-1}$ 是在第 $l-1$ 次迭代后选择任务 $\bar{e}_{u_i}^l$ 的联盟. 根据已建立的理论基础, $\varepsilon_{u_i}(e_{u_i}^{l-1}, E_{-u_i}^{l-1})$ 和 $\varepsilon_{u_i}(\bar{e}_{u_i}^l, E_{-u_i}^{l-1})$ 的数值关系与转换联盟前后势函数大小关系是一致的, 因此可以通过单个无人机转换联盟收益博弈的过程最大化势函数.

当 $\varepsilon_{u_i}(\bar{e}_{u_i}^l, E_{-u_i}^{l-1}) > \varepsilon_{u_i}(e_{u_i}^{l-1}, E_{-u_i}^{l-1})$ 时, 基于上述联盟偏好关系决策规则可得无人机能够通过改变联盟以提高总联盟收益, 此时 u_i 的策略选择更新为 $\bar{e}_{u_i}^l$, 即

$$c_{\bar{e}_{u_i}^l}^l \leftarrow c_{\bar{e}_{u_i}^l}^{l-1} \cup \{u_i\}, c_{u_i}^l \leftarrow c_{u_i}^{l-1} \setminus \{u_i\} \quad (50)$$

而当 $\varepsilon_{u_i}(e_{u_i}^{l-1}, E_{-u_i}^{l-1}) \geq \varepsilon_{u_i}(\bar{e}_{u_i}^l, E_{-u_i}^{l-1})$ 时, 无人机 u_i 策略选择不改变, 即

$$c_{\bar{e}_{u_i}^l}^l \leftarrow c_{\bar{e}_{u_i}^l}^{l-1}, c_{u_i}^l \leftarrow c_{u_i}^{l-1} \quad (51)$$

经过第 2 阶段的博弈过程, 由性质 2 可知在经历多次迭代后可以得到纳什均衡解, 从博弈论的角度可知该解不等同于全局最优解. 为提高解的质量, 算法增加扰动优化的思想: 当连续无法提高联盟总收益的迭代次数超过设定的 N 次且总迭代次数小于设定的迭代次数 β_{\max} 时, 认为联盟结构已达到纳什均衡状态. 此时在保存稳定联盟结构的基础上强制交换两个无人机的策略选择, 以此为基础进入新的迭代过程, 具体迭代步骤与第 2 阶段一致.

本文采用基于概率的方式选择两个需要交换的无人机. 当得到纳什均衡解后计算每个无人机的收益 $\varepsilon_{u_i}(e_{u_i}^*, E_{-u_i}^*)$, c_{T_0} 中的无人机收益记为 $\varepsilon_{u_i}(T_0^*, E_{-u_i}^*) = 0$. 因此

$$\varepsilon_{u_i}(e_{u_i}^*, E_{-u_i}^*) \geq 0, u_i \in U \quad (52)$$

利用 sigmoid 函数对其处理后进行归一化, 得到每个无人机被选择的概率 pro_{u_j} :

$$\begin{cases} \gamma_{u_j} = \frac{1}{1 + e^{-\varepsilon_{u_j}}} \\ pro_{u_j} = \frac{\gamma_{u_j}}{\sum_{u_i \in U} \gamma_{u_i}} \end{cases} \quad (53)$$

无人机收益越小, 被选中的概率越大.

基于概率选择两个在不同联盟中的无人机互换

任务后, 继续随机选择单个无人机进行迭代直至达到新的稳定联盟结构, 此时比较新的稳定解和交换前保存的纳什均衡解的质量, 保留总收益最大的联盟结构. 当仍未达到最大迭代次数 β_{\max} 时, 在保存的解的基础上继续循环交换和迭代过程. 直至迭代结束, 得到质量较高的纳什均衡解.

交换两个无人机的任务选择的目的在于通过给予局部最优解小的扰动以增加探索其他可行解空间的可能性, 相比于随机初始化再进行博弈迭代, 起始于一个较优的解使寻优更具有导向性.

任务分配问题中有 n 个无人机和 m 个任务, 联盟组合的可能性为 $(m+1)^n$, 这是一个有限的数字. 根据性质 1, 势博弈至少存在一个纳什均衡点, 联盟总收益最大的解必为纳什均衡解. 根据性质 2 势博弈的有限递增属性, 任何单方面改进的序列能够在有限时间内收敛于纳什均衡.

针对上述设计的健康情况下的基于联盟形成博弈的分布式任务预分配算法, 接下来是其收敛性的具体证明.

定理 2. 本文提出的基于联盟形成博弈的分布式任务预分配算法能够使无人机在有限次迭代内收敛到纳什均衡状态, 即最终的联盟结构是纳什稳定的.

证明. 利用反证法, 假设算法得到的解是不稳定的, 则 U 中至少存在一个无人机是不稳定的. 即在联盟结构中不存在不稳定无人机 u_i , $u_i \in c_{u_i}$ 且 $\exists c_{\bar{u}_i} \neq c_{u_i}$, 有

$$\varepsilon_{u_i}(c_{\bar{u}_i}, E_{-u_i}) > \varepsilon_{u_i}(c_{u_i}, E_{-u_i}) \quad (54)$$

根据算法的设计, 在第 2 阶段博弈过程中, 无人机的收益随着策略的调整呈现单调变化, 即只有新的策略能够提高收益时, 无人机才会更换所在联盟. 基于此收益博弈过程原理, u_i 可以通过改变策略来提高自身收益.

因此在其他无人机策略已达到稳定状态且不变的情况下, 单个无人机可以通过有限次数的改变策略以提高自身收益并收敛至稳定, 与所提假设矛盾. 从而本文提出的基于联盟形成博弈的分布式任务预分配算法能够在有限次迭代内收敛到纳什均衡解. \square

3.1.2 算法复杂度分析

将 m 个任务分配给 n 个无人机, 上述提出的基于联盟形成博弈的分布式任务预分配算法复杂度分析如下.

1) 计算可行解空间. 考虑时效性约束给出可行的任务分配解, 即判断每个无人机是否能在任务截

止前到达任务. 此时时间复杂度记为 $O(c_1 mn)$, 其中 c_1 为常数, 是计算无人机抵达任务所需时间的复杂度.

2) 计算出可行解空间后, 进行随机初始化. 按照算法流程, 在每次迭代时随机选择一个无人机进行交换联盟, 需要计算交换前后的无人机效用大小. 若将 c_2 记为计算单个联盟效用的时间复杂度, 每次迭代的时间复杂度则为 $O(4c_2)$. 因此 β_{\max} 次迭代下的时间复杂度可记为 $O(4\beta_{\max} c_2)$.

3) 算法增加扰动优化阶段后, 当连续迭代超过设定次数而联盟总收益未提高且总迭代次数小于 β_{\max} 时, 基于概率选择收益较小的无人机强制交换联盟继续迭代, 因此每次强制交换前对概率的计算的复杂度为 $O(2nc_2)$, 即计算了 n 个无人机退出联盟前后的联盟效用.

因此, 算法的总时间复杂度可记为 $O(c_1 mn) + O(4\beta_{\max} c_2) + k \times O(2nc_2)$, 是多项式型复杂度, k 是满足强制交换的次数. 枚举法的组合个数为 $(m+1)^n$, 而时间复杂度为 $O((m+1)^n \times m \times c_2)$, 复杂度呈指数型增加.

本文基于联盟形成博弈的分布式任务预分配算法能够有效降低问题的复杂度, 更适合应用于较大规模的任务分配问题. 同时在本算法中无人机只根据自己的效用函数进行任务选择的优化, 具有良好的鲁棒性.

在无人机状态均健康情况下可利用上述基于联盟形成博弈的分布式任务预分配算法完成任务的预分配, 得到分配结果后无人机开始执行任务. 当无人机发生故障后, 产生任务重分配需求, 接下来是任务重分配的算法设计.

3.2 任务重分配算法设计

考虑在任务执行过程中无人机由于突发事件或外部攻击等情况发生不可逆故障, 即无人机不再具备继续执行任务的能力. 根据任务分配问题数学建模可知, 式中约束条件均为耦合约束, 单个无人机的策略的变化可能会影响全局任务分配结果. 为保证后续任务的成功完成, 此时产生任务重分配需求.

为尽可能降低故障影响并提高重分配的实时性, 设计以预分配结果为基础且能在尽可能短的时间内得到新的无冲突解的重分配算法. 其中故障无人机记为 u^{fault} , 故障发生时刻记为 t^{fault} . 本文针对任意无人机在任意时刻可能发生故障的情况, 建立以下的重分配框架, 框架主要流程包含:

步骤 1. 根据预分配结果判断各无人机故障时刻的状态, 即位置与速度的更新.

步骤 2. 根据每个任务对应的联盟内所有无人机的状态判断任务的执行情况:

1) 当 $t^{\text{fault}} \geq t_{u_i, T_j} (\forall u_i \in c_{T_j})$, 此时联盟内无人机已到达任务位置, 无人机的状态有三种, 分别为

$$\begin{cases} t^{\text{fault}} < t_{T_j}^{\text{start}}, & \text{等待任务的执行} \\ t_{T_j}^{\text{start}} \leq t^{\text{fault}} < t_{T_j}^{\text{start}} + t_{T_j}^{\text{con}} + t_{T_j}^{\text{ucon}}, & \text{任务执行中} \\ t^{\text{fault}} \geq t_{T_j}^{\text{start}} + t_{T_j}^{\text{con}} + t_{T_j}^{\text{ucon}}, & \text{已完成任务} \end{cases} \quad (55)$$

2) 当 $t^{\text{fault}} < t_{u_i, T_j} (\exists u_i \in c_{T_j})$, 则故障发生时任务尚未开始执行.

步骤 3. 根据任务执行情况判断重分配的无人机和任务范围, 合理的局部重分配范围的选取能够降低问题规模, 在更短的时间内得到一个较优解. 对于联盟内全为健康无人机且都已到达任务的情况, 该任务和联盟内的无人机不参与重新分配, 只对剩余无人机和任务进行局部的重分配.

步骤 4. 接下来是任务重分配的建模准备, 需要根据步骤 1 中得到的无人机的位置和速度判断其到达任务的运动模式, 不同的运动模式取决于无人机与任务之间的距离. 准确判断运动模式对无人机到达任务时间的准确计算有重要意义. 图 6 是可能的四种运动模式示意图, 具体分析如下:

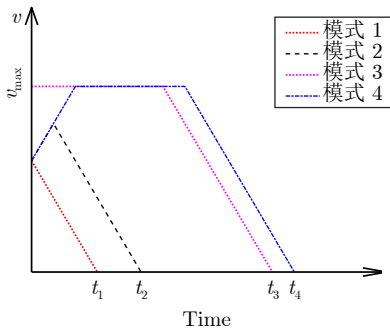


图 6 出现故障后剩余无人机运动模式示意图

Fig. 6 The motion mode of remaining UAVs after faults

1) 当故障发生时无人机的速度为 v 而与任务的距离 $d \leq \frac{v^2}{2a}$ 时, 距离仍按照 $\frac{v^2}{2a}$ 计算. 由于无人机不能快速减速, 因此延长飞行距离以确保到达任务时的速度为 0. 此时到达任务的时间为

$$t_1 = \frac{v}{a} \quad (56)$$

2) 当故障发生时无人机的速度为 v 而与任务的距离 $\frac{v^2}{2a} < d \leq \frac{2v_{\text{max}}^2 - v^2}{2a}$, 此时无人机先加速到 v' 后开始匀速减速至 0, 此时到达任务的时间计算为

$$\begin{aligned} \frac{v'^2 - v^2}{2a} + \frac{v'^2}{2a} &= d \Rightarrow \\ v' &= \sqrt{\frac{v^2 + 2ad}{2}} \Rightarrow \\ t_2 &= \frac{v' - v}{a} + \frac{v'}{a} = \frac{2v' - v}{a} \end{aligned} \quad (57)$$

3) 当故障发生时无人机的速度为 v_{max} 而与任务的距离 $d > \frac{v_{\text{max}}^2}{2a}$, 此时无人机先以最大速度 v_{max} 保持匀速一段时间后减速至 0, 此时到达任务的时间计算为

$$\begin{aligned} \frac{v_{\text{max}}}{2a} + v_{\text{max}}t' &= d \Rightarrow \\ t' &= \frac{d}{v_{\text{max}}} - \frac{v_{\text{max}}}{2a} \Rightarrow \\ t_3 &= \frac{v_{\text{max}}}{a} + t' = \frac{v_{\text{max}}}{2a} + \frac{d}{v_{\text{max}}} \end{aligned} \quad (58)$$

4) 当故障发生时无人机的速度为 $v < v_{\text{max}}$, 而与任务的距离 $d > \frac{2v_{\text{max}}^2 - v^2}{2a}$, 此时无人机先匀速加速到最大速度 v_{max} 后保持匀速一段时间后减速至 0, 此时到达任务的时间计算为

$$\begin{aligned} \frac{v_{\text{max}}^2 - v^2}{2a} + v_{\text{max}} \left(t' - \frac{v_{\text{max}} - v}{a} \right) + \frac{v_{\text{max}}^2}{2a} &= d \Rightarrow \\ t_4 &= \frac{2v_{\text{max}} - v}{a} + \frac{d}{v_{\text{max}}} - \frac{2v_{\text{max}}^2 - v^2}{2av_{\text{max}}} \end{aligned} \quad (59)$$

步骤 5. 在上述局部划分和运动模型分析后进行故障后的重分配, 具体算法如下: 首先为提高重分配的实时性, 以预分配结果为初始任务重分配方案, 将故障无人机从所在任务联盟中删除. 相比于迭代起始于一个随机分配解, 预分配结果质量更高且只需要进行局部调整即可得到较优的任务重分配解.

当 $u^{\text{fault}} \notin c_{T_j}$ 时, 有

$$\begin{cases} c'_{T_j} = c_{T_j} \\ R'(T_j) = R(T_j) \end{cases} \quad (60)$$

否则当 $u^{\text{fault}} \in c_{T_j}$ 时, 有

$$\begin{cases} c'_{T_j} = c_{T_j} \setminus \{u^{\text{fault}}\} \\ R'(T_j) = R(T_j) - \varepsilon_{u^{\text{fault}}}(c_{T_j}, E_{-u^{\text{fault}}}) \end{cases} \quad (61)$$

其次, 与任务预分配算法中的第 2 阶段博弈过程相同, 在第 l 次迭代中随机选择一个健康无人机 $u_j, u_j \in U$, 计算其此时的策略选择下的收益:

$$\varepsilon_{u_j}(e_{u_j}^{l-1}, E_{-u_j}^{l-1}) = R'(c_{u_j}^{l-1}) - R'(c_{u_j}^{l-1} \setminus \{u_j\}) \quad (62)$$

选择 E_{u_j} 中的除 $e_{u_j}^{l-1}$ 外的任务 $\bar{e}_{u_j}^l$, 计算更新

策略后的无人机收益:

$$\varepsilon_{u_j}(\bar{e}_{u_j}^l, E_{-u_j}^{l-1}) = R'(c_{\bar{e}_{u_j}^l}^{l-1} \cup \{u_j\}) - R'(c_{\bar{e}_{u_j}^l}^{l-1}) \quad (63)$$

随机选择无人机进行任务的调整, 当无人机在新联盟中的自身效用高于在原联盟中的效用时, 可以通过转换联盟提高整体收益. 算法不断随机选择健康无人机进行迭代直至达到稳定状态. 下面是任务重分配算法的收敛性证明.

定理 3. 基于本文设计的重分配框架下的基于联盟形成博弈的任务重分配算法最终可以收敛到纳什均衡解.

证明. 首先在该任务重分配框架中, 通过局部划分将无人机集群分为保持预分配结果不变和需参与重分配的两类无人机. 任务重分配算法得到的解的纳什均衡状态是基于参与局部重分配的无人机讨论的.

因此, 其收敛性证明与任务预分配算法的证明相同, 采用反证法假设存在无人机在重分配结果中是不稳定的, 在算法的收益博弈过程中, 该无人机可以在其他无人机策略保持不变的情况下通过改变策略来提高自身收益. 因此本文提出的基于联盟形成博弈的任务重分配算法最终可以收敛到纳什均衡解. \square

4 仿真验证

为验证所提任务预分配和任务重分配算法的可行性和有效性, 本文构建多无人机执行多任务的仿真场景, 分别围绕该场景在无故障情况下和发生故障情况下进行仿真验证. 首先给出该场景中的具体仿真条件.

4.1 仿真条件

仿真场景是将 3 个任务分配给 12 个无人机, 区域大小为 $1\,000\text{ m} \times 1\,000\text{ m}$. 本文任务分配问题中的无人机仿真飞行参数如表 1 所示. 无人机飞行高度 $H = 100\text{ m}$, 携带四种消耗型异构资源, 资源数量随机生成; 共有三种非消耗资源传输数据类子任务, 每个无人机能力各异, 如 $(1, 0, 1)$ 表示无人机能够传输第 1 种和第 3 种数据类型. 加速度大小 $a = 5\text{ m/s}^2$, 最大速度 $v_{\max} = 20\text{ m/s}$. 无人机和任务的位置随机生成, 任务仿真参数如表 2 所示.

接下来是基于上述仿真参数设置的任务预分配和任务重分配算法的仿真结果与分析.

4.2 任务预分配算法仿真结果

首先基于上述参数设置验证本文基于联盟形成博弈的任务预分配算法. 为比较算法得到的最终解

表 1 仿真参数符号和数值

Table 1 The notations and values of simulation parameters

符号	数值和单位	符号	数值和单位
P_0	158.76 W	W	20 N
P_i	88.63 W	m	2.04 kg
U_{tip}	120 m/s	S_{FP}	0.015 l m^2
v_0	4.03 m/s	f_c	2.4 GHz
d_0	0.6	c	$3 \times 10^8\text{ m/s}$
ρ	1.225 kg/m^3	σ^2	-174 dBm/Hz
s	0.05	η_1	3 dB
A	0.503 m^2	η_2	23 dB
B	1 MHz		

表 2 任务仿真参数

Table 2 The simulation parameters of tasks

参数	编号		
	T_1	T_2	T_3
位置	(798.4, 848.3)	(442.5, 829.7)	(585.9, 501.6)
消耗型资源需求数量	(22, 11, 14, 22)	(19, 13, 17, 27)	(26, 19, 11, 18)
数据量 (Mbit)	(30, 90, 100)	(150, 100, 40)	(60, 100, 50)
时间窗口	(24, 55)	(24, 55)	(24, 55)
折扣系数	0.1	0.1	0.1
任务持续时间 $t_{T_j}^{\max}$	1.38	1.52	1.48
任务发射功率 p_{T_j} (W)	1	1	1

的质量, 在相同参数设置下分别将灰狼算法、粒子群算法、文献 [22] 中的分布式联盟形成博弈算法与本文设计的预分配算法重复运行 50 次, 统计运行结果得到联盟总收益箱型图 (图 7). 同时分别选取算法具有代表性的收敛过程加以解释说明, 得到图 8.

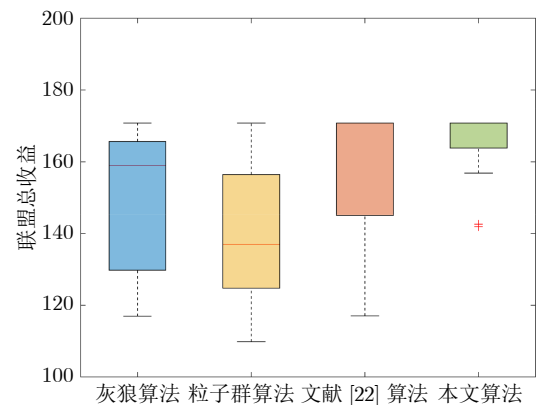


图 7 联盟总收益统计值箱型图

Fig. 7 Box plots of statistical results of total revenue of coalitions

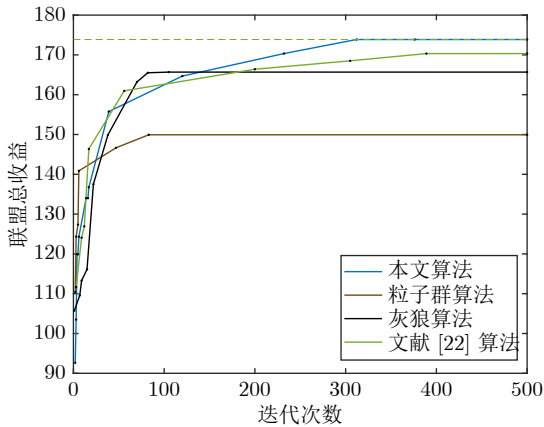


图 8 $n = 12, m = 3$ 时的算法典型收敛过程图
 Fig.8 Typical convergence process diagram of algorithm under $n = 12, m = 3$

从图 8 可以看出, 灰狼算法和粒子群算法均可较快收敛到稳定解, 但结合图 7 可得早熟收敛导致最终解的平均质量较差. 灰狼算法的性能在一定程度上依赖于初始种群的质量, 初始种群的选择可能会影响最终结果, 且过早收敛导致无法找到全局最优解; 粒子群算法收敛速度快但全局搜索能力较差, 寻优过程中粒子群体的多样性可能逐渐丧失, 导致搜索空间收缩陷入局部最优. 文献 [22] 和本文中的算法收敛速度相似, 然而算法只收敛到纳什均衡状态的性质使其同样容易陷入较差的局部最优解. 在增加扰动优化后, 本文所提算法解的平均质量显然高于上述三种算法, 大概率能够收敛到全局最优, 能够提高最终解的质量稳定性.

同样基于上述参数设置在运行时间超过 1 300 s 后得到利用枚举法求解的最优任务分配方案, 与上述四种算法运行得到的最优结果相同, 从而验证了四种算法的有效性. 图 9 为无人机仿真场景下的最优分配结果图, 图中画布为 1 000 m × 1 000 m 的二维仿真场景, 椭圆内是完成同一任务的无人机联盟集合, u_1, u_2, u_8 是未被分配任务的无人机.

为进一步验证算法的有效性, 给出本文算法与其他算法分别在不同的无人机数量 n 和不同的任务数量 m 下的联盟总效用值的对比, 每个实验重复 50 次取平均值. 首先取任务数量 $m = 3$, 无人机数量 $n \in \{4, 6, 8, 10, 12, 14, 16\}$, 不同算法仿真参数条件均相同. 图 10 是当无人机数量变化时, 四种算法重复运行得到的联盟总收益的平均值与全局最优任务分配方案收益即最大总收益之比.

当无人机数量 $n \leq 12$ 时, 最大联盟总收益取枚举法运行结果; 当无人机数量 $n > 12$ 时, 由于组合可能性数量大导致枚举法计算量过大, 取四种算法

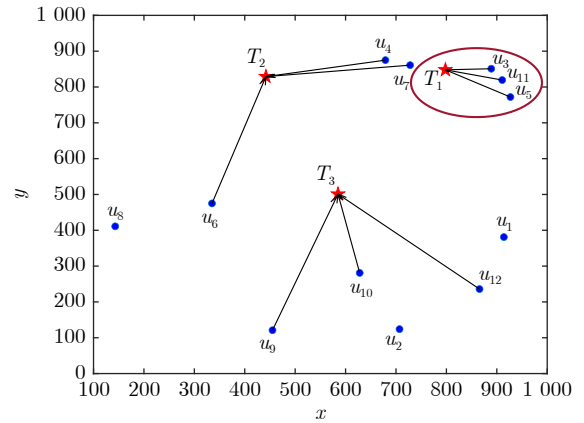


图 9 健康情况下的预分配最优结果图
 Fig.9 Graph of optimal preallocation result under healthy conditions

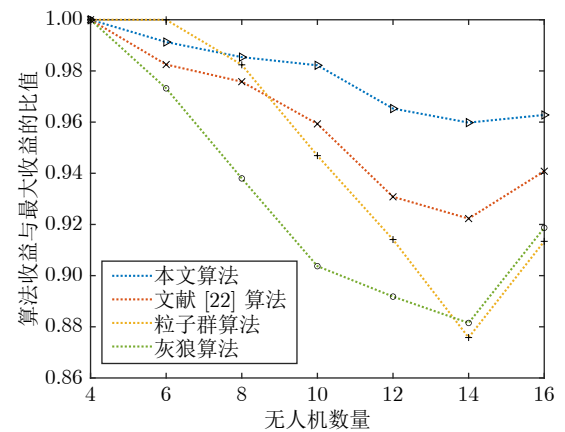


图 10 不同无人机数量下算法收益与最大总收益之比
 Fig.10 The ratio of revenues to maximum total revenues under different numbers of UAVs

重复实验得到的最大值作为全局最大值. 由图 10 可以看出, 当无人机数量 $n = 16$ 时, 本文提出的算法平均联盟总收益仍能达到全局最优的 95.98%, 相比于其他三种算法有明显优势. 特别说明的是, 当无人机数量 $n = 16$ 时算法运行效果比 $n = 14$ 时效果更好, 这是不同参数设置导致的正常现象, 如联盟最大总收益基数变大时同样的差值占比将会减小.

图 11 是算法各运行 50 次后得到的标准差示意图, 通过衡量收益的离散程度进一步验证解的稳定性. 可得本文所提算法在平均收益最高的同时解的稳定性最高, 即算法每次运行普遍可以收敛到质量较高的纳什均衡状态. 而粒子群算法虽然在图 10 中解的质量可观, 但从标准差可以看出解的稳定性较差. 同时结果可以进一步解释上文中的 $n = 16$ 算法运行效果比 $n = 14$ 时更优越的原因, 标准差较小说明局部最优解较为密集且逼近全局最优, 从而平

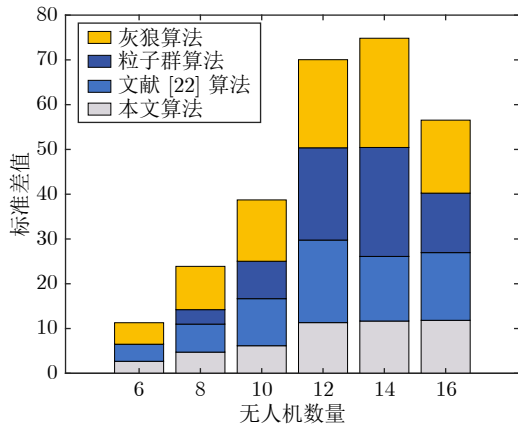


图 11 不同无人机数量下算法运行结果标准差
Fig. 11 Standard deviation of running results under different numbers of UAVs

均解质量更高。

将算法运行时间与枚举法运行时间进行对比以验证算法的实时性, 得到图 12. 图中柱状图是算法随着无人机数量变化时的运行时间, 红色折线图是与枚举法的运行时间之比. 可以看出当无人机数量 $n = 16$ 时, 算法平均运行时间仍未超过 3.5 s, 远小于枚举法的运行时间的 1/1000, 算法的实时性得证.

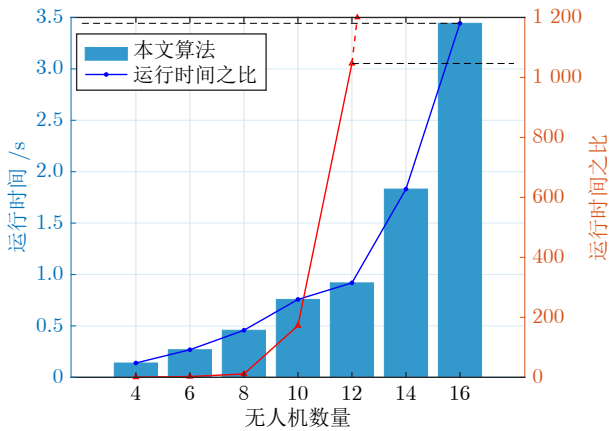


图 12 不同无人机数量下算法平均运行时间及与枚举法运行时间之比

Fig. 12 The average running time of the algorithm and the ratio to the running time of the enumeration method under different numbers of drones

取无人机数量 $n = 12$, 任务数量 $m \in \{2, 3, 4, 5, 6, 7, 8\}$, 图 13 给出在不同的任务数量下算法的平均联盟总收益的对比. 枚举法随着组合的指数型增加而计算时间迅速增长导致不可行, 每组的最优衡量标准仍取四种算法多次运行的最大值.

从图 13 中可得本文所提算法在任务增加的情况下平均联盟总收益仍可达到全局最优的 93.71%

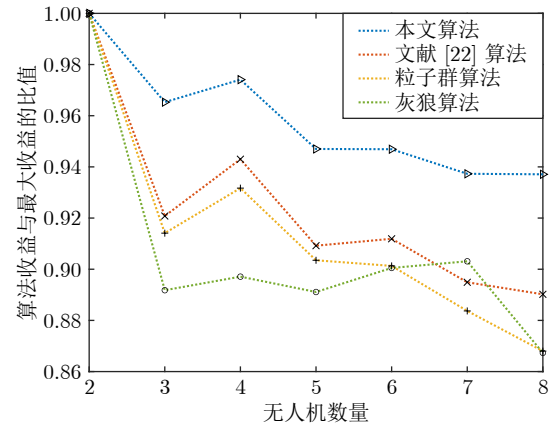


图 13 不同任务数量下算法收益与最大总收益之比
Fig. 13 The ratio of algorithm revenues to maximum total revenues under different numbers of tasks

以上, 优于上述三种算法; 粒子群算法、灰狼算法的解效果较差, 与算法依赖初始种群质量、易陷入局部最优解的性质相关, 同时算法参数的设置也会影响优化过程.

综上所述, 所提出的基于联盟形成博弈的分布式任务预分配算法具有较好的实时性和稳定性, 能较快地收敛到一个质量较高的纳什均衡状态.

4.3 任务重分配算法仿真结果

基于上述仿真参数, 利用分布式任务预分配算法得到的结果如表 3 所示. 本文忽略任务预分配算法运行所需时间, 假设各无人机在得到预分配结果后从 $t = 0$ s 开始执行任务.

表 3 $m = 3, n = 12$ 时的任务预分配方案
Table 3 The task preallocation solution under $m = 3, n = 12$

无人机	任务	无人机	任务	无人机	任务
u_1	T_0	u_5	T_1	u_9	T_3
u_2	T_0	u_6	T_2	u_{10}	T_3
u_3	T_1	u_7	T_2	u_{11}	T_1
u_4	T_2	u_8	T_0	u_{12}	T_3

图 14 ~ 16 为无人机仿真场景下的最优分配结果图, 图中画布为 $1\ 000\ \text{m} \times 1\ 000\ \text{m}$ 的二维仿真场景.

场景 1. 假设无人机 u_{12} 在 $t = 0$ s 时出现故障, 基于联盟形成博弈的分布式任务重分配算法得到的重分配结果和示意图分别如表 4 和图 14 所示.

图 14 中, 红色圆圈内 u_{12} 为故障无人机, 在预分配结果中执行任务 T_3 , 发生故障后退出任务的执行; 黑色方框内 u_2 在预分配结果中不执行任务, 经

表 4 u_{12} 在 $t = 0$ s 时发生故障后任务重分配方案

Table 4 The task reallocation solution after the fault of u_{12} at $t = 0$ s

无人机	任务	无人机	任务	无人机	任务
u_1	T_0	u_5	T_1	u_9	T_3
u_2	T_3	u_6	T_2	u_{10}	T_3
u_3	T_1	u_7	T_2	u_{11}	T_1
u_4	T_2	u_8	T_0	u_{12}	故障, 退出

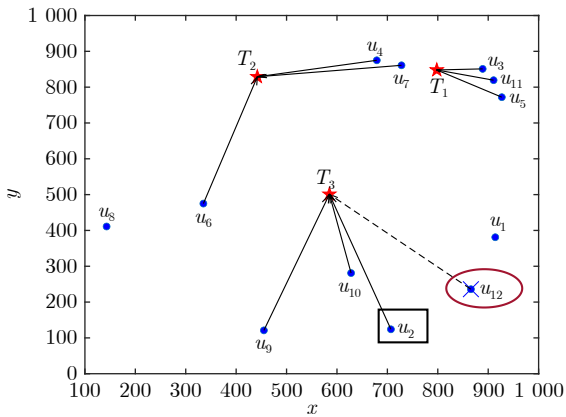


图 14 u_{12} 在 $t = 0$ s 时发生故障后任务重分配示意图

Fig. 14 Schematic of task reallocation after the fault of u_{12} at $t = 0$ s

历重分配后加入故障无人机 u_{12} 初始所在联盟。

场景 2. 假设无人机 u_9 在 $t = 8$ s 时出现故障, 基于联盟形成博弈的分布式任务重分配算法得到的重分配结果和示意图分别如表 5 和图 15 所示。

图 15 中, 红色圆圈内 u_9 为故障无人机, 在预分配结果中执行任务 T_3 , 发生故障后退出任务的执行; 黑色方框内 u_6 在预分配结果中执行任务 T_2 , 经历重分配后改为加入故障无人机 u_9 初始所在联盟。

场景 3. 假设无人机 u_7 在 $t = 15$ s 时出现故障, 基于联盟形成博弈的分布式任务重分配算法得到的重分配结果和示意图分别如表 6 和图 16 所示。

图 16 中, u_7 为故障无人机, 在预分配结果中执行任务 T_2 , 发生故障后退出任务的执行; 经历重分

表 5 u_9 在 $t = 8$ s 时发生故障后任务重分配方案

Table 5 The task reallocation solution after the fault of u_9 at $t = 8$ s

无人机	任务	无人机	任务	无人机	任务
u_1	T_0	u_5	T_1	u_9	故障, 退出
u_2	T_0	u_6	T_3	u_{10}	T_3
u_3	T_1	u_7	T_2	u_{11}	T_1
u_4	T_2	u_8	T_0	u_{12}	T_3

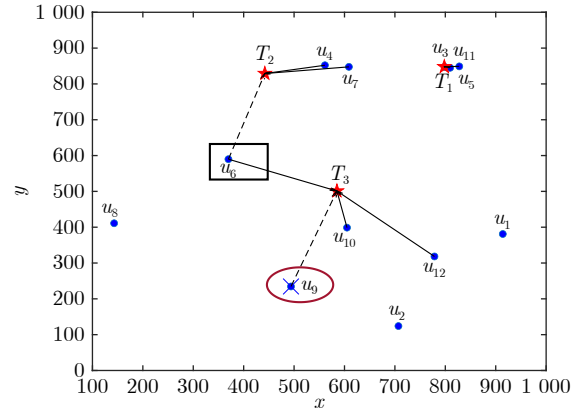


图 15 u_9 在 $t = 8$ s 时发生故障后任务重分配示意图

Fig. 15 Schematic of task reallocation after the fault of u_9 at $t = 8$ s

表 6 u_7 在 $t = 15$ s 时发生故障后任务重分配方案

Table 6 The task reallocation solution after the fault of u_7 at $t = 15$ s

无人机	任务	无人机	任务	无人机	任务
u_1	T_0	u_5	T_1	u_9	T_3
u_2	T_0	u_6	T_2	u_{10}	T_3
u_3	T_1	u_7	故障, 退出	u_{11}	T_1
u_4	T_2	u_8	T_0	u_{12}	T_3

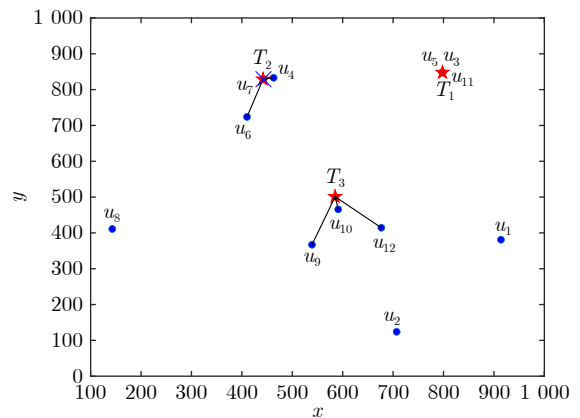


图 16 u_7 在 $t = 15$ s 时发生故障后任务重分配示意图

Fig. 16 Schematic of task reallocation after the fault of u_7 at $t = 15$ s

配后无无人机的任务选择发生变化。

从上述三种场景的任务重分配结果可得, 当无人机发生故障后, 为尽可能降低故障影响, 可以通过调用未分配任务的无人机、调整已分配任务的无人机策略、保持原任务分配方案不变等方式获得新的重分配方案, 不同故障场景下的最佳局部调整方式不同。

为进一步验证本文所提任务重分配算法的有效性和可行性, 在三个场景下分别利用枚举法、基于随机初始化迭代的算法和模拟退火算法进行联盟总收益和运行时间的对比. 按照所设计的任务重分配框架, 场景 1 和场景 2 故障发生时所有无人机和任务需要参与重分配, 场景 3 故障发生时任务 1 和对应的联盟内无人机不参与重分配, 因此参与计算部分的重分配总收益降低.

图 17 是不同算法在三个场景下的任务重分配联盟总收益对比. 由图 17 可得本文设计的分布式任务重分配算法能够通过局部调整获得较优的重分配收益, 可以实现与枚举法相同的分配效果.

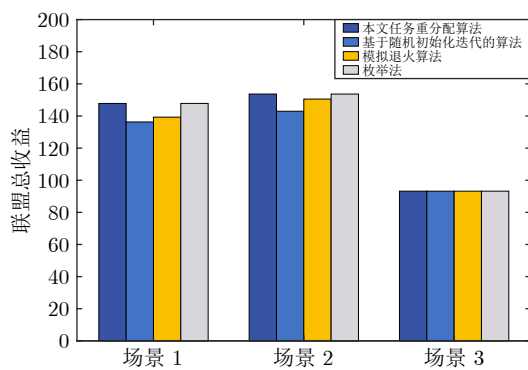


图 17 不同场景下任务重分配总收益图

Fig. 17 Total revenues of task reallocation in different scenarios

图 18 是不同算法在三个场景下的算法运行时间对比. 相比于另外三种算法, 本文任务重分配算法通过充分利用预分配结果, 显著提升了重分配的实时性, 只需要局部调整即可达到新的纳什均衡状态. 在不同场景下本文提出的分布式任务重分配

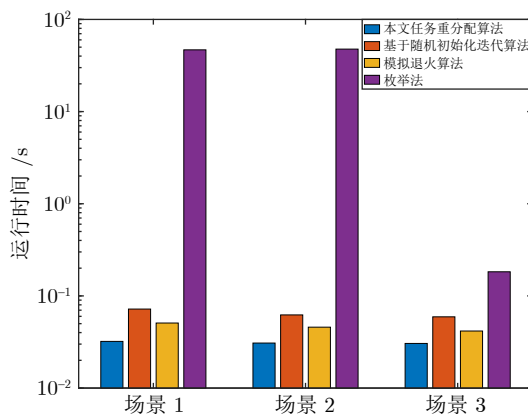


图 18 不同场景下任务重分配算法运行时间示意图

Fig. 18 Diagram of task reallocation algorithm running time in different scenarios

算法平均运行时间是 $t = 0.0306$ s, 具有明显时间优势.

在场景 1 中, u_{12} 在 $t = 0$ s 发生故障后利用重分配可以在原方案的基础上提高联盟总收益的 3.41%, 其中, T_3 收益提高 14.17%; 在场景 2 中, u_9 在 $t = 8$ s 发生故障后重分配能够提高联盟总收益的 2.79%, 其中, T_2 收益降低 23.85%, T_3 收益提高 51.51%; 在场景 3 中, 由于任务具有时间窗口约束且重新调动无人机会导致飞行能耗的增加, 因此此时最优重分配策略是保持原方案不变.

综上所述, 本文提出的基于联盟形成博弈的分布式任务重分配算法能够在所设计的重分配框架下以较短时间得到一个较优的任务重分配方案, 基于预分配结果进行局部调整的思想显著提升了重分配的实时性.

5 结束语

本文基于联盟形成博弈解决异构无人机集群系统任务预分配和重分配问题, 建立了具有现实意义和应用价值的任务分配数学模型. 利用势博弈的性质为算法的设计提供理论基础, 设计了一种基于联盟形成博弈的分布式任务预分配和重分配算法.

在任务预分配中, 基于一种新的联盟偏好关系决策规则不断寻优以提高整体联盟收益, 在迭代中加入扰动优化的思想, 当寻优停止时基于概率强制交换无人机的任务, 增加探索其他最优解的可能性. 仿真实验证明了算法的有效性和实时性, 降低问题复杂度的同时具有较高的平均运行质量. 针对任务重分配问题, 为更好地应对任意无人机在任意时刻发生故障的情况设计了任务重分配框架; 设计了基于预分配结果进行局部调整的任务重分配算法, 从而获得新的任务重分配方案, 显著提升了重分配的实时性, 仿真实验证明了算法的有效性.

本文算法的应用场景为静态下的任务预分配和故障后的任务重分配, 缺少对动态环境下的任务分配研究. 同时算法是在无人机之间能够持续直接通信的条件下设计的, 忽略了无人机通信范围可能受限的情况, 且在分布式网络系统中通信延迟是不可避免的^[33]. 因此, 可在以后的工作中考虑动态环境和通信条件约束下的任务分配方法.

References

- Hu Z J, Gao X G, Wan K F, Wang Q L, Zhai Y W. Asynchronous curriculum experience replay: A deep reinforcement learning approach for UAV autonomous motion control in unknown dynamic environments. *IEEE Transactions on Vehicular Technology*, 2023, **72**(11): 13985-14001
- Liu K, Zheng J. UAV trajectory planning with interference

- awareness in UAV-enabled time-constrained data collection systems. *IEEE Transactions on Vehicular Technology*, 2024, **73**(2): 2799–2815
- 3 Zeng Y, Xu X L, Jin S, Zhang R. Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning. *IEEE Transactions on Wireless Communications*, 2021, **20**(7): 4205–4220
- 4 He W J, Yao H P, Mai T L, Wang F, Guizani M. Three-stage stackelberg game enabled clustered federated learning in heterogeneous UAV swarms. *IEEE Transactions on Vehicular Technology*, 2023, **72**(7): 9366–9380
- 5 Wu Wen-Liang, Zhou Xing-She, Shen Bo, Zhao Yue. A review of swarm robotic systems property evaluation research. *Acta Automatica Sinica*, 2022, **48**(5): 1153–1172
(武文亮, 周兴社, 沈博, 赵月. 集群机器人系统特性评价研究综述. 自动化学报, 2022, **48**(5): 1153–1172)
- 6 Zhang J, Cui Y N, Ren J. Dynamic mission planning algorithm for UAV formation in battlefield environment. *IEEE Transactions on Aerospace and Electronic Systems*, 2023, **59**(4): 3750–3765
- 7 Ju Kai, Mao Ze-Hui, Jiang Bin, Ma Ya-Jie. Task allocation and reallocation for heterogeneous multiagent systems based on potential game. *Acta Automatica Sinica*, 2022, **48**(10): 2416–2428
(鞠楷, 冒泽慧, 姜斌, 马亚杰. 基于势博弈的异构多智能体系统任务分配和重分配. 自动化学报, 2022, **48**(10): 2416–2428)
- 8 Wang Feng, Huang Zi-Lu, Han Meng-Chen, Xing Li-Ning, Wang Ling. A knee point based coevolution multi-objective particle swarm optimization algorithm for heterogeneous UAV cooperative multi-task allocation. *Acta Automatica Sinica*, 2023, **49**(2): 399–414
(王峰, 黄子路, 韩孟臣, 邢立宁, 王凌. 基于 KnCMPPO 算法的异构无人机协同多任务分配. 自动化学报, 2023, **49**(2): 399–414)
- 9 Xiong F, Zheng H, Ruan L, Wang H, Tang L J, Dong X, et al. Energy-saving data aggregation for multi-UAV system. *IEEE Transactions on Vehicular Technology*, 2020, **69**(8): 9002–9016
- 10 Du Yong-Hao, Xing Li-Ning, Cai Zhao-Quan. Survey on intelligent scheduling technologies for unmanned flying craft clusters. *Acta Automatica Sinica*, 2020, **46**(2): 222–241
(杜永浩, 邢立宁, 蔡昭权. 无人飞行器集群智能调度技术综述. 自动化学报, 2020, **46**(2): 222–241)
- 11 Chen Z, Nian X H, Meng Q. Distributed optimization of multi-integrator agent systems with mixed neighbor interactions. *Automatica*, 2023, **157**(3): Article No. 111245
- 12 Zhang Z S, Liu H, Wu G H. A dynamic task scheduling method for multiple UAVs based on contract net protocol. *Sensors*, 2022, **22**(12): Article No. 4486
- 13 Garcia A, Hong M Y. Efficient rate allocation in wireless networks under incomplete information. *IEEE Transactions on Automatic Control*, 2016, **61**(5): 1397–1402
- 14 Lv Ye, Zhou Rui, Li Xing, Liu Zhi-Heng, Di Bin. Multi-task assignment algorithm based on multi-round distributed auction. *Journal of Beijing University of Aeronautics and Astronautics*, 2025, **51**(3): 1018–1027
(吕晔, 周锐, 李兴, 刘志恒, 邸斌. 基于多轮次分布式拍卖的异构多任务分配算法. 北京航空航天大学学报, 2025, **51**(3): 1018–1027)
- 15 Liu D, Dou L Q, Zhang R L, Zhang X Y, Zong Q. Multi-agent reinforcement learning-based coordinated dynamic task allocation for heterogeneous UAVs. *IEEE Transactions on Vehicular Technology*, 2023, **72**(4): 4372–4383
- 16 Yuan R P, Dou J T, Li J T, Wang W, Jiang Y F. Multi-robot task allocation in e-commerce RMFS based on deep reinforcement learning. *Mathematical Biosciences and Engineering*, 2023, **20**(2): 1903–1918
- 17 Xu Y H, Jiang B, Yang H. Two-level game-based distributed optimal fault-tolerant control for nonlinear interconnected systems. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, **31**(11): 4892–4906
- 18 Jie Y M, Guo C, Choo K-K R, Liu C Z, Li M C. Game-theoretic resource allocation for fog-based industrial internet of things environment. *IEEE Internet of Things Journal*, 2020, **7**(4): 3041–3052
- 19 Jang I, Shin H S, Tsourdos A. Anonymous hedonic game for task allocation in a large-scale multiple agent system. *IEEE Transactions on Robotics*, 2018, **34**(6): 1534–1548
- 20 Wang L X, Qiu T H, Pu Z Q, Yi J Q, Zhu J Y, Yuan W M. Hedonic coalition formation for distributed task allocation in heterogeneous multi-agent system. *International Journal of Control Automation and Systems*, 2024, **22**(4): 1212–1224
- 21 Chen J X, Wu Q H, Xu Y H, Qi N, Guan X, Zhang Y L, et al. Joint task assignment and spectrum allocation in heterogeneous UAV communication networks: A coalition formation game-theoretic approach. *IEEE Transactions on Wireless Communications*, 2021, **20**(1): 440–452
- 22 Zhang T X, Wang Y H, Ma Z J, Kong L J. Task assignment in UAV-enabled front jammer swarm: A coalition formation game approach. *IEEE Transactions on Aerospace and Electronic Systems*, 2023, **59**(6): 9562–9575
- 23 Qi N, Huang Z Q, Zhou F H, Shi Q J, Wu Q H, Xiao M. A task-driven sequential overlapping coalition formation game for resource allocation in heterogeneous UAV networks. *IEEE Transactions on Mobile Computing*, 2023, **22**(8): 4439–4455
- 24 Huo X, Zhang H, Huang C, Wang Z P, Yan H C. Task allocation with minimum requirements for multiple mobile robot systems: A game-theoretical approach. *IEEE Transactions on Network Science and Engineering*, 2024, **11**(1): 1202–1213
- 25 Tang J, Chen X, Zhu X M, Zhu F. Dynamic reallocation model of multiple unmanned aerial vehicle tasks in emergent adjustment scenarios. *IEEE Transactions on Aerospace and Electronic Systems*, 2023, **59**(2): 1139–1155
- 26 Zhou Wen-Hui, Qi Rui-Yun, Jiang Bin. Mission replanning method of distributed multiple unmanned aerial vehicles for pop-up faults. *Control and Decision*, 2023, **38**(5): 1373–1385
(周文惠, 齐瑞云, 姜斌. 面向突发故障的分布式多无人机任务重规划方法. 控制与决策, 2023, **38**(5): 1373–1385)
- 27 Wang G, Lv X, Cui L, Yan X. The methods of task pre-allocation and reallocation for multi-UAV cooperative reconnaissance mission. *IET Collaborative Intelligent Manufacturing*, 2023, **5**(4): Article No. e12090
- 28 Zeng Y, Xu J, Zhang R. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Transactions on Wireless Communications*, 2019, **18**(4): 2329–2345
- 29 Yang Z H, Xu W, Shikh-Bahaei M. Energy efficient UAV communication with energy harvesting. *IEEE Transactions on Vehicular Technology*, 2020, **69**(2): 1913–1927
- 30 Mozaffari M, Saad W, Bennis M, Debbah M. Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications. *IEEE Transactions on Wireless Communications*, 2017, **16**(11): 7574–7589
- 31 Zeng Y, Zhang R, Lim T J. Throughput maximization for UAV-enabled mobile relaying systems. *IEEE Transactions on Communications*, 2016, **64**(12): 4983–4996

- 32 Huang S J, Lei J L, Hong Y G. A linearly convergent distributed Nash equilibrium seeking algorithm for aggregative games. *IEEE Transactions on Automatic Control*, 2023, **68**(3): 1753–1759
- 33 Liu J, Ho D W C, Li L L. A generic algorithm framework for distributed optimization over the time-varying network with communication delays. *IEEE Transactions on Automatic Control*, 2024, **69**(1): 371–378



姜斌 南京航空航天大学自动化学学院教授. 主要研究方向为智能故障诊断与容错控制及应用. 本文通信作者.

E-mail: binjiang@nuaa.edu.cn

(JIANG Bin Professor at the College of Automation Engineering, Nanjing University of Aeronautics

and Astronautics. His research interest covers intelligent fault diagnosis and fault-tolerant control and their applications. Corresponding author of this paper.)



马亚杰 南京航空航天大学自动化学学院教授. 主要研究方向为自适应故障诊断与容错控制及应用.

E-mail: yajiema@nuaa.edu.cn

(MA Ya-Jie Professor at the College of Automation Engineering, Nanjing University of Aeronautics

and Astronautics. His research interest covers adaptive fault diagnosis and fault-tolerant control and their applications.)



薛舒心 南京航空航天大学自动化学学院硕士研究生. 主要研究方向为无人机集群任务分配.

E-mail: lyuxuan@nuaa.edu.cn

(XUE Shu-Xin Master student at the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics. Her main research interest is

task allocation for UAV swarms.)