



基于联盟形成博弈的异构无人机集群分布式任务分配算法

薛舒心^{1,3}, 马亚杰^{1,3*}, 姜斌^{1,3}, 李文博^{1,2}, 刘成瑞^{1,2}

1. 南京航空航天大学自动化学院, 南京 210016

2. 北京控制工程研究所空间智能控制技术全国重点实验室, 北京 100094

3. 飞行器自主控制技术教育部工程研究中心, 南京 210016

* 通信作者. E-mail: yajiem@nuaa.edu.cn

收稿日期: 2024-06-04; 修回日期: 2024-08-02; 接受日期: 2024-08-19; 网络出版日期: 2024-11-11

国家自然科学基金 (批准号: 62273177, 62020106003, 62233009)、江苏省自然科学基金 (批准号: BK20211566, BK20222012)、高校学科创新引智基地 (批准号: B20007)、空间智能控制技术全国重点实验室开放基金 (批准号: HTKJ2023KL502006) 和中央高校基本科研业务费 (批准号: NI2024001) 资助项目

摘要 针对无人机集群携带异构资源以及任务的异构需求下的复杂任务分配优化求解问题, 提出一种基于联盟形成博弈的分布式任务分配算法. 首先针对任务分配问题规模较大的特点以及资源的异构性, 提出一种基于异构资源的改进 K-medoids 聚类算法, 通过对无人机集群和任务进行聚类的预处理, 降低了任务分配的规模和难度. 考虑任务需求、机载资源以及路径成本等条件建立任务分配模型, 将原有任务分配问题转化为联盟划分问题, 设计了一种基于联盟形成博弈的分布式任务分配算法进行求解. 最后, 将 30 个具有异构需求的任务分配给 100 架携带 3 种异构资源的无人机的仿真结果表明, 所提算法能够实现较好的任务分配效果, 同时极大提高任务分配的实时性, 充分发挥集群效能.

关键词 任务分配, 异构无人机集群, 异构资源, 聚类算法, 联盟形成博弈

1 引言

随着人工智能等无人机相关技术的突破和发展, 无人机自适应性和鲁棒性不断增强, 在各领域都得到广泛应用^[1]. 例如, 可利用无人机执行航拍摄影、电力巡检、高层消防救援^[2]等任务; 或利用无人机执行情报侦察、电子干扰^[3]等任务. 同时, 无人机类型趋向多样化、集群化, 由于任务领域范围的拓展和执行环境的复杂性, 单架无人机受到探测能力、武器载荷等因素的限制^[4], 无人机集群执行大规模任务时可以提供更好的性能. 随着资源需求和任务种类增多, 通过采用任务分配算法, 将具有不同需求的任务分配给合适的异构无人机, 集群可以在能耗、任务效用等方面充分发挥良好的整体效

引用格式: 薛舒心, 马亚杰, 姜斌, 等. 基于联盟形成博弈的异构无人机集群分布式任务分配算法. 中国科学: 信息科学, 2024, 54: 2657-2673, doi: 10.1360/SSI-2024-0167

Xue S X, Ma Y J, Jiang B, et al. Distributed task allocation algorithm for heterogeneous unmanned aerial vehicle swarm based on coalition formation game (in Chinese). Sci Sin Inform, 2024, 54: 2657-2673, doi: 10.1360/SSI-2024-0167

能^[5]. 无人机集群任务分配技术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使用价值.

日益增加的问题规模和复杂性给无人机集群任务分配提出新的挑战. 在异构性上, 无人机性能各异, 集群从单一的同构发展到复杂异构; 在规模上, 从传统编队的数架无人机发展到几十架甚至更多. 同时, 任务通常需要多个无人机协同完成, 因此算法还需要考虑无人机之间的合作. 面对以上需求, 分布式算法能够赋予无人机一定的自主性和决策权, 具有灵活性和鲁棒性, 比集中式算法消耗更少的通信和计算资源^[6], 成为学术研究的重要方向.

常见的分布式任务分配算法主要有博弈论方法、基于市场机制的方法^[7]和分布式马尔可夫决策过程方法. 合同网算法和拍卖算法是应用较为广泛的基于市场机制的方法. 文献 [8] 针对异构无人机和未知目标背景提出了一种基于合作网协议的任务分配算法, 能够获得较大的系统效能, 但算法缺少对某无人机可能同属多个任务最优联盟解的冲突消除考虑. 文献 [9] 针对不同作战任务, 建立了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, 本文模型中无人机可执行不同类型的任务, 但缺少对执行相同类型任务的无人机能力的区分. 基于分布式马尔可夫法通常训练难度大, 收敛较慢. 文献 [10] 针对边缘服务器的分布式任务和资源分配问题, 将其建模为部分观测的马尔可夫决策过程, 提出一种依赖于多智能体的深度强化两步策略, 所提出的解决方案在训练和收敛方面优于基准.

目前, 得益于博弈论能够为无人机之间的合作协商提供良好性能, 博弈方法已成为热门研究方向, 应用于各种优化问题^[11]. 文献 [5] 设计了智能体之间的组合策略, 将全局效用映射为势函数获得最优分配方案. 文献 [12] 采用了基于势博弈的分布式多智能体动态任务分配方法, 该算法在全局搜索能力方面表现良好. 文献 [13] 设计了基于偏好联盟形成博弈的任务分配算法, 利用偏好程度定义机器人加入联盟后的效用, 算法保证了多项式时间复杂性和解的纳什稳定性.

其中, 联盟形成博弈 (coalition formation game, CFG) 模型是常用的博弈模型之一, 目的在于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了一个高效的模型与工具. 在整个博弈过程中, 参与者会自行组成若干个联盟, 从全局来看形成若干分组. 文献 [14] 设计了一种基于联盟博弈的面向组的分布式聚类算法, 算法经验证在通信链路稳定性、聚类数量和负载平衡等方面优于现有算法. 文献 [15] 针对无人机前置群干扰提出一种基于分布式联盟形成博弈的动态任务分配算法, 将任务分配问题转化为多智能体协同问题, 算法可以实现与集中式优化相似的性能. 文献 [16] 考虑资源属性和任务执行顺序的重叠和互补关系, 在基于顺序重叠联盟形成博弈模型基础上开发了一种偏好重力引导禁忌搜索算法, 可获得稳定的联盟结构.

综上所述, 分布式任务分配算法已取得较多研究成果, 尤其在博弈方面. 然而, 目前工作仍存在以下问题: 一是对于机载资源和任务需求资源的异构性研究较为薄弱; 二是大规模无人机集群在进行任务分配时求解难度剧增, 导致算法实时性较差; 三是单个无人机运载能力很难满足任务需求, 需要考虑无人机之间的协作.

基于联盟形成博弈解决任务分配问题能够根据无人机在性能、能力和资源方面的差异灵活进行任务分配, 提高系统整体效率; 其次具有较高的可拓展性, 能够适应不同规模的无人机集群. 同时联盟形成博弈支持分布式决策, 无人机能够自主进行任务分配, 提升系统的鲁棒性和可靠性; 通过合理的收益分配机制, 激励无人机积极参与任务执行, 增强协作效果. 因此, 本文提出一种基于联盟形成博弈的分布式任务分配算法, 主要贡献总结如下:

(1) 针对原任务分配问题规模大以及资源的异构性, 在 K-medoids 聚类算法的基础上加入关于异构资源数目均衡度的收敛条件以进行改进, 对无人机集群和任务进行合理的预处理, 降低原问题的规模, 提高任务分配的实时性;

(2) 建立了联盟形成博弈数学模型, 将任务分配问题转化为联盟划分问题; 通过证明本文联盟形成博弈模型为势博弈, 证明了纳什均衡解的存在性并为算法的设计提供了理论基础;

(3) 设计了基于联盟形成博弈的分布式任务分配算法, 主要包含了最大加权匹配阶段和无人机转移联盟判断阶段, 使得算法相比随机寻优更具有导向性, 能够获得较好的分配效果并且提高任务分配的实时性.

2 问题描述

考虑将 m 项任务分配给 n 个异构无人机的任务分配问题. 与固定翼无人机相比, 旋翼无人机具有更好的悬停能力, 适合执行需要稳定悬停状态的侦察、监测等类型任务. 因此本文任务分配问题中考虑的无人机类型为能够垂直起降的旋翼无人机. 设定异构无人机可以同时执行不同任务类型, 且每个无人机在执行不同任务时发挥的效能各异, 其特性可量化为携带资源的类型与数目的差别. 同时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. 在本文任务分配问题中, 任务需要由多个无人机协同执行, 分配结果中完成同一个任务的无人机称为一个联盟. 通过结合实际简化问题, 做出以下假设:

- (1) 无人机、任务状态信息已知, 且任务状态信息在分配过程中保持不变;
- (2) 每架无人机都具备良好的通信能力, 若有需无人机可以进行两两通信.

2.1 数学建模

2.1.1 无人机建模

无人机集群中有 n 个无人机, 集合表示为 $U = \{u_1, u_2, \dots, u_n\}$, 无人机在类型、资源等方面是异构的, 表现在执行同一个任务时可以提供不同的效能, 量化为携带资源的类型与数目不同. 利用四元组来描绘无人机状态信息 UIN :

$$UIN \langle Res, position, value, v \rangle, \quad (1)$$

$$UIN = \{UIN_{u_i} | i = 1, 2, \dots, n\}. \quad (2)$$

假设任务分配问题中共有 l 种资源类型, $Re_{u_i}^k$ 是无人机携带第 k 种资源的数目. 无人机具体的性能表示为 $Res_{u_i} = \{Re_{u_i}^1, Re_{u_i}^2, \dots, Re_{u_i}^l\}$, 即携带的资源类型及数目, $position_{u_i}$ 表示无人机 u_i 位置信息. 在类如作战等特定背景中, 无人机自身价值相异, 为了提高模型的适用程度, 加入价值信息 $value_{u_i}$.

得益于旋翼无人机的高机动性, 无人机转变飞行方向和改变速度所需时间可忽略不计. 旋翼无人机的弊端在于依靠升力导致能量效率较低, 因此飞行速度有限. 为了使模型更具有实际意义, 引入了无人机飞行速度约束, 即无人机速度大小需满足

$$v_{u_i} \leq v_{u_i}^{\max}, \forall u_i \in U, \quad (3)$$

其中, $v_{u_i}^{\max}$ 为常数, 表示各无人机最大速度约束.

$$\begin{aligned} UIN_{u_i} &= \langle Res_{u_i}, position_{u_i}, value_{u_i}, v_{u_i} \rangle \\ &= \langle \{Re_{u_i}^1, \dots, Re_{u_i}^l\}, (x_{u_i}, y_{u_i}, z_{u_i}), value_{u_i}, v_{u_i} \rangle. \end{aligned} \quad (4)$$

2.1.2 任务建模

共有 m 个任务, 集合表示为 $\mathbf{T} = \{t_1, t_2, \dots, t_m\}$. 每个任务包含不同类型的子任务, 任务状态信息 TIN 描述如下:

$$TIN \langle Res, position, TW, a \rangle, \quad (5)$$

$$TIN = \{TIN_{t_j} | j = 1, 2, \dots, m\}. \quad (6)$$

$Res_{t_j} = \{Re_{t_j}^1, Re_{t_j}^2, \dots, Re_{t_j}^l\}$ 是任务所需的资源类型及数目, $Re_{t_j}^k$ 是任务所需要的第 k 种资源的数目. 任务 t_j 的位置信息记为 $position_{t_j}$. TW 是任务的硬时间窗口信息, 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.

$$TW = [t^{\minstart}, t^{\maxstart}]. \quad (7)$$

当无人机执行不同任务时, 任务本身可能对无人机有一定的破坏能力, 因此在建模中引入威胁指数 a_{t_j} , 表示任务 t_j 对无人机的威胁系数.

$$\begin{aligned} TIN_{t_j} &= \langle Res_{t_j}, position_{t_j}, TW_{t_j}, a_{t_j} \rangle \\ &= \langle \{Re_{t_j}^1, \dots, Re_{t_j}^l\}, (x_{t_j}, y_{t_j}, z_{t_j}), [t_{t_j}^{\minstart}, t_{t_j}^{\maxstart}], a_{t_j} \rangle. \end{aligned} \quad (8)$$

2.1.3 联盟形成博弈数学建模

在本文中无人机集群任务分配问题建模为联盟形成博弈模型. 定义博弈模型 $\mathbf{G} = (\mathbf{U}, \mathbf{E}, \varepsilon, \mathbf{R})$, \mathbf{U} 是无人机集合. 策略集合 $\mathbf{E} = [e_{u_1}, e_{u_2}, \dots, e_{u_n}]$ 是无人机集合选择的任务对应的集合, 在本文中等价于任务集合 \mathbf{T} , 即 $\mathbf{E}_U = \mathbf{T}$. ε 是设计用来评估无人机收益的效用函数, \mathbf{R} 旨在评估各个联盟的效用, 图 1 是联盟形成博弈模型的总体流程图.

此时原问题转化为联盟划分问题, 当无人机做出策略后会形成 $(m+1)$ 个任务联盟. 博弈目标是无人机集合选择合适策略从而得到稳定的联盟结构 \mathbf{CS} , $\mathbf{CS} = \{c_{t_0}, c_{t_1}, c_{t_2}, \dots, c_{t_m}\}$. c_{t_j} 为执行任务 t_j 的无人机联盟集合, 其中, c_{t_0} 是未分配任务的无人机联盟集合.

当无人机 u_i 加入联盟 c_{t_j} 时, 收益 $r_{u_i}(t_j)$ 包含三部分, 分别是无人机资源贡献 $\mathbf{val}(u_i, t_j)$ 、执行任务的路径成本 $\mathbf{cost}(u_i, t_j)$ 以及执行任务威胁代价 $\mathbf{risk}(u_i, t_j)$.

无人机对任务的资源贡献 $\mathbf{val}(u_i, t_j)$ 定义为

$$\mathbf{val}(u_i, t_j) = \begin{cases} \mathbf{K}(j, :) \mathbf{I} - PO, & \text{if } \mathbf{K}(j, :) \mathbf{I} - PO > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

其中, \mathbf{K} 和 P 分别是权重矩阵和常数权值, 权衡资源在价值收益中的比重. $\mathbf{K}(j, :) = \{k_{j1}, k_{j2}, \dots, k_{jl}\}$, 表示对于任务 t_j 各异构资源的重要程度. $\mathbf{I} = \{i_1, i_2, \dots, i_l\}$, 表示无人机加入任务对应的联盟集合中可利用的每类资源的数目, 取决于任务本身与联盟内其他成员. O 指该无人机未利用的总资源数目, 即加入该联盟后冗余资源总数目.

对于权重矩阵 \mathbf{K} , 可灵活设置其元素大小. 当对任务 t_j 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 $\frac{m}{l}$, 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时, 此时向量内权重值正比于联盟所需该类型资源数目, 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, 可根据各类异构资源数量取具体权重, 使携带攻击资源较多的无人机

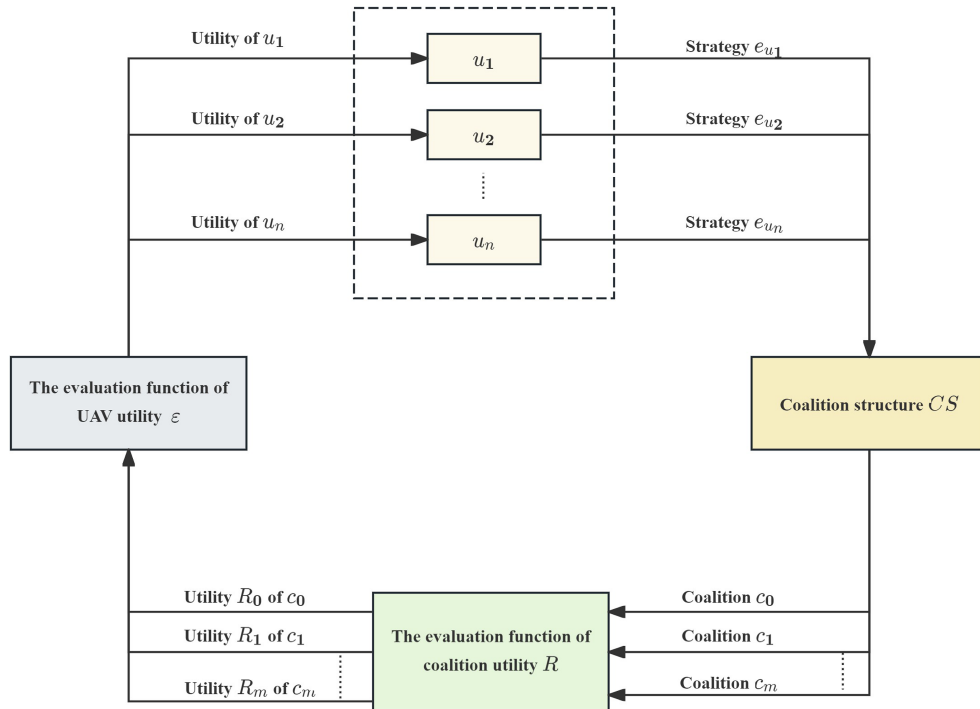


图 1 (网络版彩图) 联盟形成博弈模型总体流程图

Figure 1 (Color online) Overall flowchart of the coalition formation game model

具有更高的优先级. 权重的具体值可以根据实际情况灵活设置, 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.

执行任务路径成本 $\text{cost}(u_i, t_j)$ 记为

$$\text{cost}(u_i, t_j) = \begin{cases} 1 - \mathbf{d}(u_i, t_j) / \sqrt{x^2 + y^2}, & \mathbf{val}(u_i, t_j) > 0, \\ \mu, & \text{otherwise,} \end{cases} \quad (10)$$

其中, 路径成本 \mathbf{d} 指的是无人机与任务之间的欧氏距离, x 和 y 是任务环境区域大小. 其中, μ 为一个小于 0 的常数. 当 $\mathbf{val}(u_i, t_j)$ 为 0 时, 设计 $r_{u_i}(t_j)$ 小于 0. 含义是当无人机 u_i 加入任务 t_j 联盟无法贡献资源时, 加入该联盟的收益小于在 c_{t_0} 中的收益.

执行任务的威胁代价 $\mathbf{risk}(u_i, t_j)$ 的定义是

$$\mathbf{risk}(u_i, t_j) = \text{value}_{u_i} a_{t_j}, \quad (11)$$

表示任务的探测雷达和攻击能力等因素对无人机造成的威胁代价评估, 是常见的衡量任务执行的经济成本之一. 任务会对无人机造成损伤, 通过加入对异构无人机价值的考虑从而降低无人机整体损耗^[17].

综上, 任务收益记为

$$r_{u_i}(t_j) = \alpha \mathbf{val}(u_i, t_j) + \beta \text{cost}(u_i, t_j) - \gamma \mathbf{risk}(u_i, t_j), \quad (12)$$

其中, α 、 β 、 γ 都是常数权重值, 分别决定资源重叠度和路径成本以及执行任务威胁代价在收益中的

比重. 因此, 任务 t_j 的收益记为

$$\mathbf{R}(c_{t_j}) = \sum_{u_i \in c_{t_j}} r_{u_i}(t_j). \quad (13)$$

式 (13) 设计用来评估联盟效用. 任务分配问题的总收益记为

$$SR = \sum_{c_{t_j} \in CS} \mathbf{R}(c_{t_j}). \quad (14)$$

此外, 设计用以评估无人机效用的函数表达式如下:

$$\varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i}) = \mathbf{R}(c_{u_i}) - \mathbf{R}(c_{u_i}|u_i), \quad (15)$$

其中, c_{u_i} 是无人机 u_i 选择策略 e_{u_i} 时加入的任务联盟. $A|b$ 的含义是从 A 中排除元素 b , $\mathbf{R}(c_{u_i}|u_i)$ 是将 u_i 从原所属联盟中删除后的剩余联盟效用.

本文博弈的目标是获得最终稳定的纳什均衡解 CS^* , 纳什均衡是博弈论中一种解的概念, 如下给出其定义.

定义1 (纳什均衡 (Nash equilibrium)) 是指满足下面性质的策略组合: 在其他玩家策略不变的前提下, 任何一位玩家在此策略组下单方面改变自己的策略都不会提高自身的收益. 在此组合中, 每个博弈者的策略都是为了达到自己期望收益的最大值. 在本文中, 具体定义为

$$\begin{aligned} CS^* &= \arg \max SR(CS) \Rightarrow \\ \varepsilon_{u_i}(e_{u_i}^*, \mathbf{E}_{-u_i}^*) &\geq \varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i}^*), \forall u_i \in U, \end{aligned} \quad (16)$$

其中, $e_{u_i}^*$ 是联盟结构 CS^* 中无人机 u_i 的策略, $\mathbf{E}_{-u_i}^*$ 是 u_i 之外的无人机在稳定的联盟结果中的策略集合, \bar{e}_{u_i} 是除 $e_{u_i}^*$ 外的任意策略, $\bar{e}_{u_i} \in \mathbf{T}$. 无人机在联盟之间不断切换以提高自身效用, 当任何无人机都无法通过独自调整所在联盟来提高收益时, 达到稳定的联盟结构. 联盟形成博弈的目标是得到稳定的联盟结构, 与任务分配的目标一致, 因此通过博弈得到的解是任务分配问题的解.

定义2 (势博弈 (potential game)) 在博弈模型中, 每个博弈参与者可以调整自身策略来使自身收益或效用最大. 若存在一个势函数 P , 满足

$$\begin{aligned} \varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i}) - \varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i}) &= P(e_{u_i}, \mathbf{E}_{-u_i}) - P(\bar{e}_{u_i}, \mathbf{E}_{-u_i}), \\ \forall u_i \in U, \forall e_{u_i}, \bar{e}_{u_i} \in \mathbf{E}_U, \end{aligned} \quad (17)$$

其能够反映参与者改变策略后的效用变化, 则该博弈称作势博弈.

性质1 势博弈必然存在纳什均衡点, 可以通过最小化或最大化势函数达到博弈的纳什均衡. 在有限递增特性下, 势博弈能够在有限时间内收敛到纳什均衡状态.

根据定义 2, 当其他无人机策略保持不变时, 某架无人机单独加入或退出某联盟的效用函数变化如果能够等量反映在势函数中, 可利用势博弈模型求取最优解.

定理1 可将已建立的博弈模型 $G = (U, \mathbf{E}, \varepsilon, \mathbf{R})$ 构造为势博弈, 且博弈模型的纳什均衡解存在.

证明 根据上述定义, 设计

$$P(e_{u_i}, \mathbf{E}_{-u_i}) = SR(e_{u_i}, \mathbf{E}_{-u_i}) = \sum_{c_{t_j} \in CS} \mathbf{R}(c_{t_j}), \quad (18)$$

当无人机 u_i 策略由 e_{u_i} 改为 \bar{e}_{u_i} , 此时

$$\begin{aligned} P(e_{u_i}, \mathbf{E}_{-u_i}) - P(\bar{e}_{u_i}, \mathbf{E}_{-u_i}) &= SR(e_{u_i}, \mathbf{E}_{-u_i}) - SR(\bar{e}_{u_i}, \mathbf{E}_{-u_i}) \\ &= [\mathbf{R}(c_{e_{u_i}}) + \mathbf{R}(c_{\bar{e}_{u_i}}) + \sum_{c_m \in CS_\Delta} \mathbf{R}(c_m)] - [\mathbf{R}(c_{\bar{e}_{u_i}}) + \mathbf{R}(c_{e_{u_i}}) + \sum_{c_m \in CS_\Delta} \mathbf{R}(c_m)]. \end{aligned} \quad (19)$$

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

$$\bar{c}_{e_{u_i}} = c_{e_{u_i}} | u_i, \quad (20)$$

$$c_{\bar{e}_{u_i}} = \bar{c}_{\bar{e}_{u_i}} | u_i, \quad (21)$$

可得

$$\begin{aligned} P(e_{u_i}, \mathbf{E}_{-u_i}) - P(\bar{e}_{u_i}, \mathbf{E}_{-u_i}) &= (\mathbf{R}(c_{e_{u_i}}) - \mathbf{R}(\bar{c}_{e_{u_i}})) - (\mathbf{R}(\bar{c}_{\bar{e}_{u_i}}) + \mathbf{R}(c_{\bar{e}_{u_i}})) \\ &= \varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i}) - \varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i}). \end{aligned} \quad (22)$$

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, 本博弈模型可构造为势博弈, $SR = \sum \mathbf{R}(c_{t_j})$ 为势函数. 由于势博弈的性质 1, 本文博弈模型的纳什均衡解必然存在, 因此在已建立的理论基础上可设计合理的联盟形成博弈算法实现最终的任务分配.

2.2 目标函数与约束条件

通过上述分析, 无人机在进行任务分配时的目标函数具体可表示为

$$\max SR = \max \sum_{c_{t_j} \in CS} \mathbf{R}(c_{t_j}), \quad (23)$$

约束条件有:

(1) 无人机飞行速度约束

$$v_{u_i} \leq v_{u_i}^{\max}, \forall u_i \in U, \quad (24)$$

表示执行任务的旋翼无人机具有最大速度约束, 意味着无人机到达任务具有最短所需时间约束.

(2) 任务时效性约束

$$\text{mint}(u_i, t_j) \leq t_{t_j}^{\max\text{start}}, \forall u_i \in U, t_j \in T, \quad (25)$$

$\text{mint}(u_i, t_j)$ 是无人机 u_i 到达任务 t_j 最短所需时间. 该约束表示任务的时效性约束, 即无人机需要在截止时间前到达任务.

(3) 无人机执行任务模式约束

$$c_{t_j} \cap c_{t_i} = \emptyset, \forall t_i, t_j \in T, \quad (26)$$

即不同任务对应的联盟之间交集为空集, 无人机最多执行一个任务.

因此, 以最大化势函数为优化目标, 本文数学模型建立如下:

$$\max_{CS} SR, \quad (27)$$

$$\text{s.t.} \quad v_{u_i} \leq v_{u_i}^{\max}, \forall u_i \in U, \quad (28)$$

$$\text{mint}(u_i, t_j) \leq t_{t_j}^{\max\text{start}}, \forall u_i \in U, t_j \in T, \quad (29)$$

$$c_{t_j} \cap c_{t_i} = \emptyset, \forall t_i, t_j \in \mathbf{T}. \quad (30)$$

本文任务分配算法的目标是求取任务对应的无人机联盟, 该联盟内无人机能够满足任务需求和约束, 充分发挥集群效能, 同时分配结果能够降低路径成本以及减少联盟内冗余资源的数量. 至此, 任务分配问题已建模为最大化势函数的联盟划分问题, 接下来是基于联盟形成博弈的算法设计.

3 联盟形成博弈算法设计

3.1 基于异构资源的改进 K-medoids 聚类算法

聚类是按照某个指标将数据集划分为不同的簇或类, 最常见的是基于距离进行分割, 尽可能增大类内元素的相似性, 同时降低类间元素的相似性. 对相似的数据进行分割归簇, 实现聚类. 为了降低原问题的规模大小, 本文在进行全局任务分配前, 作基于距离和资源对任务和无人机集群聚类的预处理, 从而缩短任务分配的计算时间, 提高任务分配效率^[18].

3.1.1 无人机集群聚类

为了保证后续类簇内局部任务分配, 要求聚类结果中每个类簇内的资源相对均衡. 因此本文基于距离和异构资源对无人机进行合理聚类, 具体如下: 首先利用 K-medoids 聚类算法对无人机进行初始聚类; 其次, 检查聚类结果是否满足设定条件, 若不满足, 重新初始化类簇中心进入迭代; 若满足, 聚类结束.

K-medoids 算法相对 K-means 鲁棒性更强, 异常点不会严重影响聚类结果. 假定数据集内共有 n 个点, 记为 $\mathbf{P} = \{p_1, p_2, \dots, p_n\}$. 将数据内所有点分为 k 个类簇, 类簇记为 $\mathbf{C} = \{c_1, c_2, \dots, c_k\}$, c_i 对应的类簇中心记为 o_i . 设计代价函数以评估聚类结果, 使其尽可能小, 其表示如下:

$$E = \sum_{j=1}^k \sum_{P \in c_j} \|p - o_i\|_2, \quad (31)$$

其中, E 为各个样本点距离各类簇中心的误差之和.

在本文中, 合理的聚类需要在收敛条件中加入对异构资源的考虑. 首先对每类资源作零均值化处理, 资源数目转化为一组和为零的正负特征数据, 类簇内特征数据的和可以表示资源的均衡度, 因此有

$$\left| \sum_{U \in c_i} \bar{R}_{u_i}^j \right| < \beta, i = \{1, 2, \dots, k\}, j = \{1, 2, \dots, l\}, \quad (32)$$

其中 β 是设定的常值, $\bar{R}_{u_i}^j$ 是经过零均值化处理后的值.

$$\sum_{u_i \in U} \bar{R}_{u_i}^j = 0. \quad (33)$$

当聚类结果满足公式 (32) 条件时, 迭代结束, 此时类簇内每类资源满足均衡要求. 基于异构资源的改进 K-medoids 聚类算法的伪代码如算法 1 所示.

上述算法基于机载异构资源类型和距离对无人机进行聚类, 通过更合理的预处理降低原任务分配问题的规模, 能够更好地开展后续的任务分配.

算法 1 基于异构资源的改进 K-medoids 聚类算法

输入: 无人机集合 U 的机载资源信息 Res_U 和位置信息 $position_U$, 类簇个数: k , 初始聚类结果

主迭代:

```

1: for  $j = 1 : l$  do
2:    $feature_j \leftarrow bsxfun(@minus, Res_U^j, mean(Res_U^j))$ , 计算无人机每类机载资源的特征数据值;
3: end for
4: for  $i = 1 : k$  do
5:   while 同一类簇内某类资源的特征数据之和  $\sum(feature_j(\text{find}(\text{cluster} == i))) \geq \beta$  do
6:     随机选择  $k$  个无人机作为初始类簇中心  $prev\_medoids\_index$ ;
7:     重复 K-medoids 聚类算法得到新的聚类结果;
8:   end while
9: end for

```

输出: 最终无人机集群聚类结果

3.1.2 任务聚类

任务聚类的具体实现步骤与上文一致, 将 m 个任务进行聚类划分为 k 个类簇. 经过聚类后的无人机集群和任务类簇可一一对应, 构成一个完整分区. 当无人机集群的总资源满足任务需求且单无人机和单任务资源相对均衡时, 经过基于资源聚类后的分区内无人机资源亦能够满足任务需求.

通过聚类的预处理极大地降低了原问题的规模, 从而缩短任务分配的计算时间, 显著提高任务分配效率.

3.2 基于联盟形成博弈的任务分配算法

利用上一节聚类算法对无人机集群和任务作聚类预处理后, 对 k 个分区分别进行任务分配, 得到分区内局部分配的解. 当所有分区完成分配, 全局任务分配完成. 算法的主要流程在于如下: 首先基于最大加权匹配原则进行求解, 其次设定无人机转移联盟规则, 逐层建立稳定联盟, 直至所有任务对应的联盟资源满足需求. 如下给出本文算法涉及内容的具体定义:

定义3 (匹配 (matching)) 给定一个二分图 G , M 是 G 的一个子图且其边集 $\{E\}$ 中的任意两个边都不依附于同一个顶点, 则 M 称为 G 的一个匹配.

定义4 (最大加权匹配 (maximum weighted matching)) 在带有权值边的二分图中, 使匹配边上的权值和最大的匹配.

在此基础上, 进行以下阶段求解. 算法的整体流程图如图 2 所示.

(1) 最大加权匹配阶段

在本文中, 将任务和无人机集合看作是二分图中的 X 和 Y 集合, 边的权值取决于无人机加入任务对应的联盟后的收益, 进行最大加权匹配. 该阶段算法具体流程如下:

步骤 1 初始化分配结果, 所有 m 项任务标记为未分配, 对应的联盟结构为空集; 所有无人机均未被分配任务, 此时

$$c_{t_0} = U. \quad (34)$$

步骤 2 在每个任务对应的联盟结构为空集的情况下, 计算每个无人机单独执行每项任务的收益, 得到 $M \times N$ 维收益矩阵 V . 为了满足任务的时效性约束, 在计算无人机加入任务的收益矩阵时, 需要根据无人机与任务的距离判断是否能够满足约束条件: 若能够在截止时间前到达则按照式 (12) 计算; 否则收益记为 0.

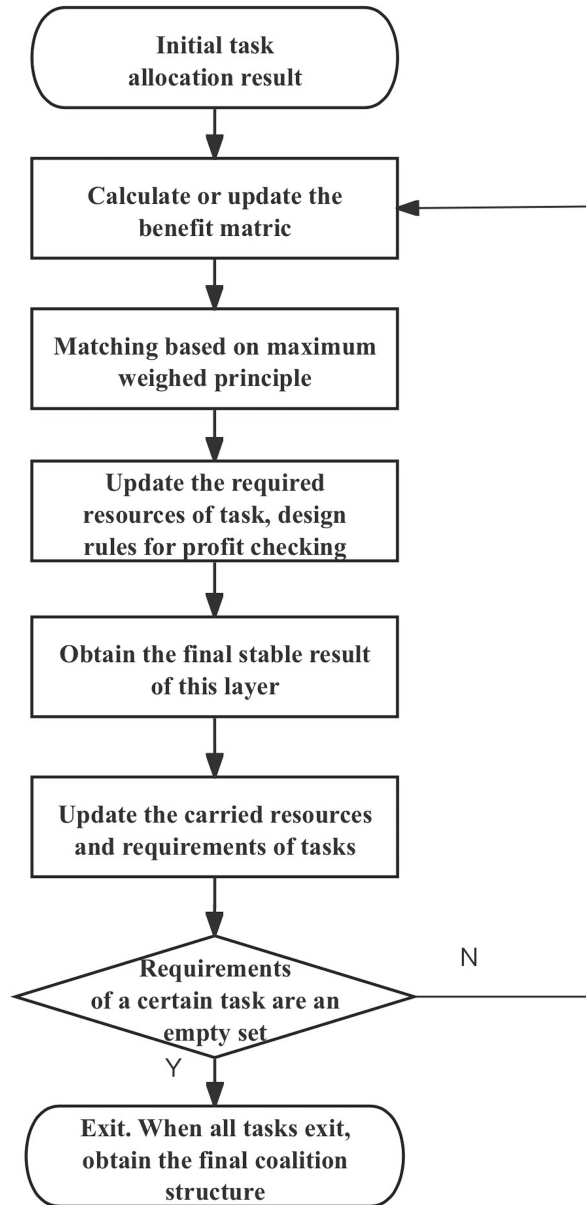


图 2 基于联盟形成博弈的任务分配算法整体流程图

Figure 2 Overall flowchart of the task allocation algorithm based on coalition formation game

步骤 3 将收益矩阵转化为加权二分图, 初始化顶标, 将任务的顶点设置为各无人机单独加入该联盟后收益的最大值 $x[i]$, 无人机顶点预设为 $y[i] = 0$. 开始进行匹配, 匹配原则是顶点值相加需等于与其相连的边的权重 $w[i][j]$, 权值 $w[i][j]$ 对应收益矩阵 V . 利用 KM 算法核心思想通过不断修改顶标和深度探索寻找可行增广路径, 匹配原则是

$$x[i] + y[j] = w[i][j]. \quad (35)$$

由于任务的数量小于无人机的数量, 需增加虚拟节点, 把虚拟节点的权重设为一个特别小的值, 使

算法 2 无人机转移联盟判断算法

输入: 最大加权匹配结果 \mathbf{E}' 和任务信息 TIN

主迭代:

```

1: for  $j = 1 : m$  do
2:   for  $u_i \in c_{t_j}$  and  $u_i = \arg \min_{u_k \in c_{t_j}} r_{u_k}(t_j)$  do
3:      $\varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i}) \leftarrow \mathbf{R}(c_{u_i}) - \mathbf{R}(c_{u_i}|u_i)$ ,
4:     for  $\bar{c}_{u_i} \neq c_{u_i}$  do
5:        $\varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i}) \leftarrow \mathbf{R}(\bar{c}_{u_i} \cup u_i) - \mathbf{R}(\bar{c}_{u_i})$ ,
6:       if  $\varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i}) \geq \varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i})$  then
7:          $u_i \in c_{u_i}$ ,
8:       else
9:          $u_i \in c_{t_0}$ ,
10:      end if
11:    end for
12:  end for
13:   $TIN_{t_j} \leftarrow TIN'_{t_j}$ ,
14: end for
15:  $TIN \leftarrow TIN'$ ,
16:  $\mathbf{V} \leftarrow \mathbf{V}'(TIN)$ .

```

输出: 最终联盟结果和更新后的 \mathbf{V}

其不会影响其他点的正常匹配. 最后删除虚拟节点匹配部分. 此时每项任务匹配到一个无人机, 满足最大加权匹配.

(2) 无人机转移联盟判断阶段

根据上一阶段匹配结果更新任务剩余所需资源, 设计无人机转移联盟规则. 在第一层得到初次分配结果后, 对已分配的无人机进行检查, 将其加入更新所需资源后的其他任务联盟内进行收益比较. 当且仅当存在加入其他更新所需资源后的任务收益仍大于该无人机在原联盟内的收益时, 无人机退出原始联盟结构, 等待新的匹配. 否则, 第一层分配结果有效, 无人机已达到稳定状态.

具体检查规则如下: 首先计算无人机此时在联盟中的效用 $\varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i})$, 计算方法为

$$\varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i}) = \mathbf{R}(c_{u_i}) - \mathbf{R}(c_{u_i}|u_i), \quad (36)$$

其中, c_{u_i} 是无人机 u_i 选择策略 e_{u_i} 时加入的任务联盟, $\mathbf{R}(c_{u_i}|u_i)$ 是将 u_i 从原所属联盟中删除后的联盟效用. 其次计算该无人机加入其他任务联盟时的自身效用 $\varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i})$, 计算方法为

$$\varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i}) = \mathbf{R}(\bar{c}_{u_i} \cup u_i) - \mathbf{R}(\bar{c}_{u_i}), \quad (37)$$

其中, \bar{c}_{u_i} 是无人机 u_i 选择策略 \bar{e}_{u_i} 时加入的任务联盟, \bar{e}_{u_i} 是除 e_{u_i} 外的任意策略, $\bar{e}_{u_i} \in \mathbf{T}$.

比较 $\varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i})$ 和 $\varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i})$ 的大小, 当存在 \bar{e}_{u_i} 使无人机效用 $\varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i})$ 较大时, 此时无人机退出原任务联盟, 等待重新分配; 如果 $\varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i})$ 最大时, 认为无人机已达到稳定状态, 保持原联盟关系不变.

根据检查结果将达到稳定状态的无人机分配到其所属联盟中, 更新剩余未分配无人机集合 c_{t_0} 和每项任务匹配后剩余所需资源类型和数目, 计算得到新的收益矩阵 \mathbf{V} . 该阶段的伪代码如算法 2 所示.

(3) 迭代, 直至稳定

依据新的收益矩阵重复最大加权匹配, 同样按照上述无人机转移联盟规则对新匹配的无人机进行效用检查, 依次进行迭代. 算法的收敛结果: 所有无人机稳定, 即经过检查后不会通过更改所属联盟来提高自身收益.

由上述基于联盟形成博弈的分布式任务分配算法得到的最终联盟结构 CS^* 是纳什稳定的, 其证明如下.

证明 假设最终联盟结构 CS^* 不稳定, 则无人机集合中至少存在一个 u_i 是不稳定的, 对于该无人机 u_i , 其可以通过切换所在联盟 c_{u_i} 来提高自身的收益. 即存在 \bar{e}_{u_i} ,

$$\varepsilon_{u_i}(\bar{e}_{u_i}, \mathbf{E}_{-u_i}) > \varepsilon_{u_i}(e_{u_i}, \mathbf{E}_{-u_i}). \quad (38)$$

根据本文的无人机转移联盟判断规则, u_i 可以转换至新的联盟, 而联盟内其他无人机效用不变. 因此 u_i 可以通过有限次数的转移联盟以实现自身的稳定, 与所提假设矛盾. 从而通过上述算法得到的最终联盟结构 CS^* 是纳什稳定的得证.

在本文所提的分布式算法中, 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, 且算法的设计追求包含全局效用最大化和实时性的提高. 根据势博弈的性质, 联盟结构经过有限次改变会收敛至纳什均衡, 算法收敛性得证.

3.3 算法复杂度分析

将 m 项任务分配给 n 个无人机的任务分配问题, 算法工作量主要在于对联盟效用的计算. 假设每次联盟效用的计算复杂度记为 CP , 则枚举法的算法复杂度为

$$O_{enum} = (m + 1)^n \times CP, \quad (39)$$

这是一个指数级的复杂度, 对于较大的无人机数量, 计算量会急剧增加. 而本文任务分配算法的复杂度为

$$\begin{aligned} O_{CFG} &= (mn + m(m - 1)) \times CP + (mn' + m(m - 1)) \times CP + \dots \\ &\leq (mn + m(m - 1)) \times CP \times l, \end{aligned} \quad (40)$$

其中, l 是迭代次数, 取决于无人机和任务参数. 这是一个多项式型复杂度, 显然本文提出的基于联盟形成博弈的分布式任务分配算法能够降低问题的复杂度和计算量.

4 仿真验证

4.1 基于异构资源的改进 K-medoids 聚类算法结果分析

首先利用简单场景验证所提聚类算法的有效性, 假设无人机集群位置在 300×300 的任务区域中随机生成, 共有 15 架, 利用聚类算法将其分为三个类簇. 每架无人机可执行侦察、监测任务, 即携带相关资源, 初始参数随机生成, 不再具体展示. 利用所设计的基于异构资源的 K-medoids 聚类算法, 所得结果如图 3 和表 1 所示.

图 3 中, 同一颜色表示属于同一类簇的无人机, 类簇中心无人机已特别标注, 虚线内是同属于类簇 2 的无人机集合. 同时, 为了进一步验证该算法的聚类效果, 采用仅考虑距离相似性的 K-medoids 聚类算法, 得到如表 2 所示的聚类结果. 由表 1 和表 2 可知, 算法改进前后的评估函数值近似. 然而,

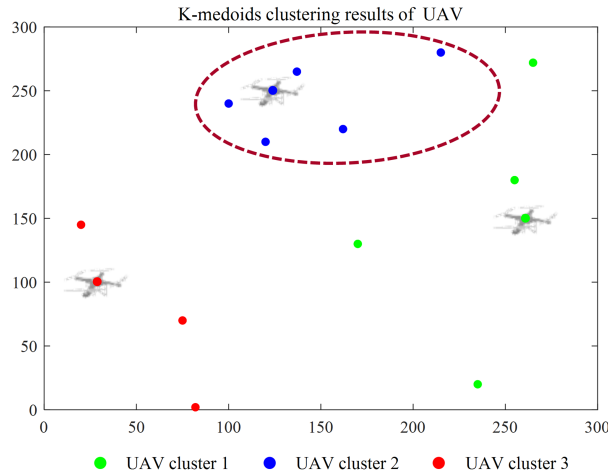


图 3 (网络版彩图) 基于异构资源的改进 K-medoids 算法聚类结果

Figure 3 (Color online) Clustering results of the improved K-medoids clustering algorithm based on heterogeneous resources

表 1 基于异构资源的改进 K-medoids 聚类算法结果分析

Table 1 Analysis of results of the improved K-medoids clustering algorithm based on heterogeneous resources

Cluster number	UAV number	Reconnaissance resources	Detection resources
1	{ $u_1, u_3, u_{10}, u_{11}, u_{15}$ }	12	15
2	{ $u_5, u_6, u_7, u_8, u_9, u_{14}$ }	12	14
3	{ u_2, u_4, u_{12}, u_{13} }	11	14

表 2 基于距离的 K-medoids 聚类算法结果分析

Table 2 Analysis of results of the K-medoids clustering algorithm based on distance

Cluster number	UAV number	Reconnaissance resources	Detection resources
1	{ u_3, u_6, u_{10} }	10	7
2	{ $u_4, u_5, u_7, u_8, u_9, u_{12}, u_{14}$ }	14	21
3	{ $u_1, u_2, u_{11}, u_{13}, u_{15}$ }	11	15

基于 K-medoids 聚类结果中, 各类簇资源均衡度远远差于基于异构资源的改进 K-medoids 聚类算法的结果, 不利于后续各类簇内的任务分配的开展.

综上, 在基于距离相似性的基础上加入对异构资源类型及数目的考虑, 设置相应的收敛条件, 能够优化聚类结果, 使无人机集群的资源能够相对均衡地分配到每个类簇中, 更利于开展后续类簇内的任务分配.

本文构建以下场景: 在 1000×1000 区域内包含 100 架无人机以及 30 个任务, 其中任务由不同子任务组成, 子任务有侦查、监测、攻击三种类型; 无人机集群由能力异构的无人机组成, 分别可以在上述三种子任务作业中发挥不同的效能. 无人机与任务的坐标随机生成. 将子任务类型使用 A、B、C 区分表示, 携带资源和任务资源需求使用常值进行量化表示. 无人机集群携带每类资源数量总和大于任务所需, 保证任务需求.

使用已验证合理性的基于资源的改进 K-medoids 聚类算法对所构建场景进行预处理, 对 100 架无

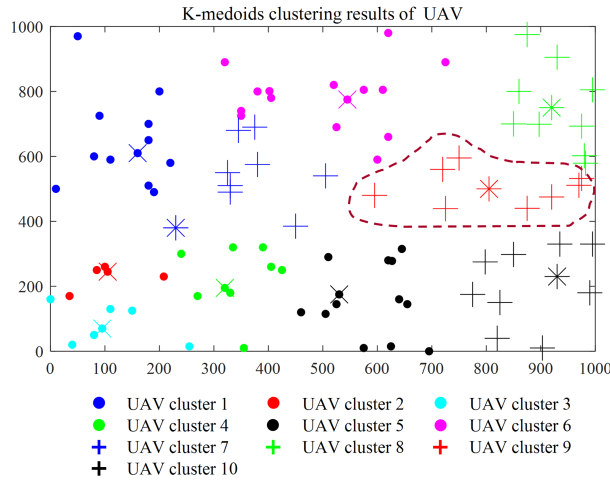


图 4 (网络版彩图) 大规模无人机集群聚类结果

Figure 4 (Color online) Clustering results of large-scale UAV clusters

表 3 基于联盟形成博弈的任务分配结果

Table 3 Results of task allocation based on coalition formation game

Task number	UAV number (according to allocation order)	Number of redundant resources	Total revenue
t_1	$\{u_5, u_9\}$	2	12.20
t_2	$\{u_3, u_4\}$	0	12.99
t_3	$\{u_7, u_1\}$	5	12.91

人和 30 个任务进行聚类, 无人机和任务信息随机生成, 将其分别分为 10 个分区, 无人机集群聚类结果如图 4 所示. 图中相同颜色的点属于同一类簇的无人机, “x” 和 “*” 表示类簇中心无人机, 虚线内是属于同一类簇的无人机集合.

4.2 基于联盟形成博弈的任务分配算法结果分析

为了验证本文基于联盟形成博弈任务分配算法, 仿真场景设置在某一次聚类结果的分区内, 共有九架异构无人机和三个任务, 初始参数不具体展开. 在相同参数设置下与枚举法进行对比, 从而验证算法的有效性.

基于联盟形成博弈算法运行结果如表 3 所示, 重复运行 10 次, 平均仿真时间为 0.094 s. 上述结果中, 每个联盟内的资源能够满足任务需求; 无人机不需要全部参与到任务联盟中, 避免了资源的浪费. 基于相同仿真参数设置, 利用枚举法来验证上述结果. 运行结果相同, 重复运行十次, 平均仿真时间为 1.392 s.

为了进一步验证算法的有效性和实时性, 设置在不同的任务和无人机数量下与枚举法、粒子群算法、遗传算法进行联盟总收益和算法运行时间的对比. 首先固定任务数量 $m = 3$, 取无人机数量 $n \in \{5, 6, 7, 8, 9, 10, 11\}$, 各算法重复运行 50 次. 图 5 是本文算法与遗传算法、粒子群算法相比于全局最优的收益衰减量, 图 6 给出了算法相对于枚举法的平均运行时间之比.

由图 5 可知, 当任务数量固定时, 遗传算法和粒子群算法在解决无人机数量较少的任务分配问题具有优势, 而随着无人机数量的增加, 本文算法的优势更加明显. 当无人机数量 $n = 11$ 时, 本文算

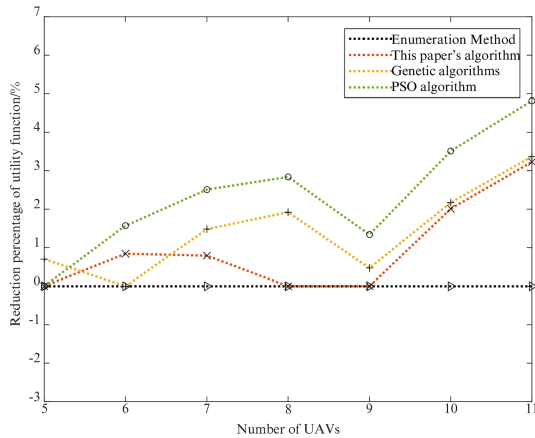


图 5 (网络版彩图) 不同无人机数量下收益函数衰减量
Figure 5 (Color online) The decreased amount of revenue function under different numbers of UAVs

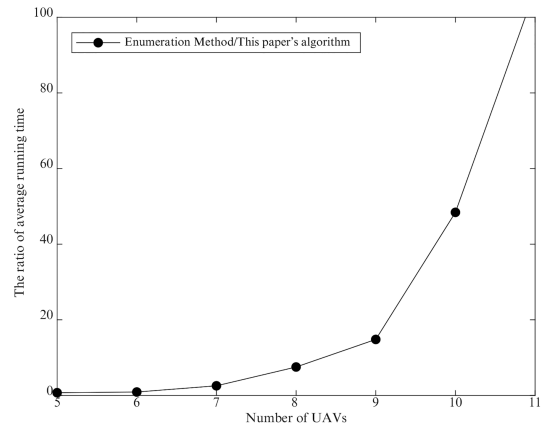


图 6 不同无人机数量下平均运行时间对比图
Figure 6 The comparison of average operating time under different numbers of UAVs

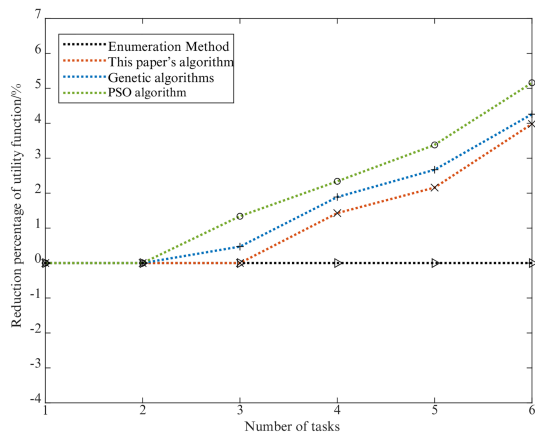


图 7 (网络版彩图) 不同任务数量下收益函数衰减量
Figure 7 (Color online) The decreased amount of revenue function under different numbers of tasks

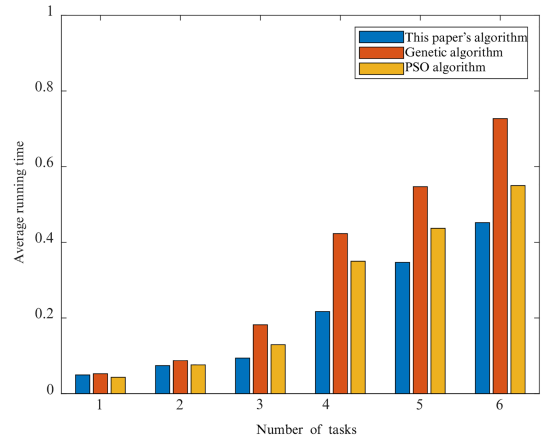


图 8 (网络版彩图) 不同任务数量下平均运行时间图
Figure 8 (Color online) The average running time under different numbers of tasks

法得到的联盟总收益衰减量仍未超过 3.30%。由图 6 可以看出当任务数量 $m = 3$ 固定时, 无人机数量 $n > 8$ 时, 算法运行时间已不足枚举法运行时间的 0.1, 具有较好的实时性。无人机-任务的组合数量增加意味着粒子群算法和遗传算法的搜索空间变大, 计算量增加且算法搜索最优解难度变大; 而通过本文算法逻辑的设计使寻优具有导向性, 算法经过有限次的迭代即可收敛到一个稳定且质量较高的纳什均衡解。

其次分别设置在任务数量 $m \in \{1, 2, 3, 4, 5, 6\}$, 对应的无人机数量 $n \in \{3, 6, 9, 10, 12, 14\}$ 的场景下。各算法重复运行 50 次, 图 7 是不同算法相对于枚举法的目标函数衰减量仿真结果, 图 8 给出了各算法运行时间示意图。需要特别说明的是由于粒子群算法和遗传算法的运行时间与设置的迭代次数相关, 本文取联盟总收益稳定收敛时平均所需迭代次数。

由图 7 和图 8 可得当无人机和任务数量较少时, 基于联盟形成博弈的分布式任务分配算法可以

获得与枚举法相同的收益. 随着无人机和任务数量的增加, 本文算法仍能在收益上表现出与遗传算法相似较好的性能, 得到较高的总联盟收益值, 同时在实时性上更加优越. 从图中可以看出当任务数量 $m = 6$ 、无人机数量 $n = 14$ 时, 本文算法收益可近似达到全局最优的 96% 以上, 同时算法运行时间接近遗传算法所需运行时间的 1/2. 粒子群算法收敛速度较快, 但最终解的质量平均质量有待提高, 过早收敛现象使其容易陷入局部最优, 算法性能依赖于参数的设计.

仿真表明, 本文提出的基于联盟形成博弈的异构无人机集群分布式任务分配算法能够综合考虑资源的异构性, 保证较低复杂度的同时得到较优的任务分配方案, 极大提高了任务分配的实时性, 充分发挥集群效能.

5 结论

本文给出了一种基于联盟形成博弈的异构无人机集群分布式任务分配算法, 考虑了大规模无人机集群机载资源的异构性以及任务的异构需求.

基于异构资源的改进 K-medoids 聚类方法能够更合理的对问题进行预处理, 降低原问题的规模; 建立的联盟形成博弈模型将原任务分配问题转化为联盟划分问题, 其具有势博弈的特性为任务分配算法提供理论基础; 利用设计的基于联盟形成博弈的分布式任务分配算法进行求解, 无需全局信息交互, 提高了时效性和对通信故障的容错性.

恶劣的工作环境或突发事件的发生有时会导致无人机故障, 导致整体性能的降低, 众多国内外学者围绕故障检测以及容错控制问题展开研究^[19]. 从控制层面已有很多较为成熟的针对各种执行器故障的容错自适应方案^[20], 此时还需要考虑集群对环境态势和动态任务迅速做出反应, 在有限时间内完成联盟的重构与优化, 达成新的无冲突的任务重分配方案. 因此, 研究任务重分配是后续研究重点.

参考文献

- 1 Poudel S, Moh S. Task assignment algorithms for unmanned aerial vehicle networks: a comprehensive survey. *IEEE Trans Veh Commun*, 2022, 35: 100469
- 2 Li H Y, Chen J T, Ren H R, et al. Random-sampling-based multi-UAV cooperative search planning for high-rise firefighting. *Sci Sin Inform*, 2022, 52: 1610–1626 [李鸿一, 陈锦涛, 任鸿儒, 等. 基于随机采样的高层消防无人机协同搜索规划. *中国科学: 信息科学*, 2022, 52: 1610–1626]
- 3 Zhou W Q, Zhu J H, Kuang M C. An unmanned air combat system based on swarm intelligence. *Sci Sin Inform*, 2020, 50: 363–374 [周文卿, 朱纪洪, 匡敏驰. 一种基于群体智能的无人空战系统. *中国科学: 信息科学*, 2020, 50: 363–374]
- 4 Lei Y Q, Duan H B. Decision-making of multi-UAV combat game via enhanced competitive learning pigeon-inspired optimization. *Sci Sin Tech*, 2024, 54: 136–148 [雷阳琦, 段海滨. 基于强化竞争学习鸽群优化的多无人机博弈决策. *中国科学: 技术科学*, 2024, 54: 136–148]
- 5 Ju K, Mao Z H, Jiang B, et al. Task allocation and reallocation for heterogeneous multiagent systems based on potential game. *Act Autom Sin*, 2022, 48: 2416–2428 [鞠锴, 冒泽慧, 姜斌, 等. 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: 2416–2428]
- 6 Cui W, Li R, Feng Y, et al. Distributed task allocation for a multi-UAV system with time window constraints. *Drones*, 2022, 6: 226
- 7 Chen X, Wei X M, Xu G Y. Multiple unmanned aerial vehicle decentralized cooperative air combat decision making with fuzzy situation. *J Shanghai Jiaotong Univ*. 2014, 48: 907–913+921 [陈侠, 魏晓明, 徐光延. 多无人机模糊态势的分布式协同空战决策. *上海交通大学学报*, 2014, 48: 907–913+921]
- 8 Yan F, Zhu X, Zhou Z, et al. Real-time task allocation for a heterogeneous multi-UAV simultaneous attack. *Sci Sin Inform*, 2019, 49: 555–569 [严飞, 祝小平, 周洲, 等. 考虑同时攻击约束的多异构无人机实时任务分配. *中国科学: 信息科学*, 2019, 49: 555–569]
- 9 Lyu Y, Zhou R, Li X, et al. Multi-task assignment algorithm based on multi-round distributed auction. *J Beijing Univ Aeronaut Astronaut*, 2023, 1–14 [吕晔, 周锐, 李兴, 等. 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*, 2023, 1–14]
- 10 Chen Y, Sun Y, Yu H, et al. Joint task and computing resource allocation in distributed edge computing systems via multi-agent deep reinforcement learning. *IEEE Trans Netw Sci Eng*, 2024, 11: 3479–3494

- 11 Xu Y, Jiang B, Yang H. Two-level game-based distributed optimal fault-tolerant control for nonlinear interconnected systems. *IEEE Trans Neural Netw Learn Syst*, 2020, 31: 4892–4906
- 12 Wu H, Shang H. Potential game for dynamic task allocation in multi-agent system. *ISA Trans*, 2020, 102: 208–220
- 13 Czarnecki E, Dutta A. Scalable hedonic coalition formation for task allocation with heterogeneous robots. *Intel Serv Robotics*, 2021, 14: 501–517
- 14 Zhang M, Li J, Wu X, et al. Coalition game based distributed clustering approach for group oriented unmanned aerial vehicle networks. *Drones*, 2023, 7: 91
- 15 Zhang T, Wang Y, Ma Z, et al. Task assignment in UAV-enabled front jammer swarm: a coalition formation game approach. *IEEE Trans Aerosp Electron Syst*, 2023, 59: 9562–9575
- 16 Qi N, Huang Z, Zhou F, et al. A task-driven sequential overlapping coalition formation game for resource allocation in heterogeneous UAV networks. *IEEE Trans Mobile Comput*, 2023, 22: 4439–4455
- 17 Wang J, Jia G, Lin J, et al. Cooperative task allocation for heterogeneous multi-UAV using multi-objective optimization algorithm. *J Cent South Univ*, 2020, 27: 432–448
- 18 Gao C, Du Y L, Bu Y N, et al. Heterogeneous UAV swarm grouping deployment for complex multiple tasks. *J Syst Eng Electron*, 2024, 46: 972–981 [高程, 都延丽, 步雨浓, 等. 面向复杂多任务的异构无人机集群分组调配. *系统工程与电子技术*, 2024, 46: 972–981]
- 19 Ma Y, Jiang B, Tao G, et al. Uncertainty decomposition-based fault-tolerant adaptive control of flexible spacecraft. *IEEE Trans Aerosp Electron Syst*, 2015, 51: 1053–1068
- 20 Mao Z, Jiang B, Shi P. Fault-tolerant control for a class of nonlinear sampled-data systems via a Euler approximate observer. *Automatica*, 2010, 46: 1852–1859

Distributed task allocation algorithm for heterogeneous unmanned aerial vehicle swarm based on coalition formation game

Shuxin XUE^{1,3}, Yajie MA^{1,3*}, Bin JIANG^{1,3}, Wenbo LI^{1,2} & Chengrui LIU^{1,2}

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

2. *National Key Laboratory of Space Intelligent Control, Beijing Institute of Control Engineering, Beijing 100094, China;*

3. *Engineering Research Center of Autonomous Control Technology of Aircraft, Ministry of Education, Nanjing 210016, China*

* Corresponding author. E-mail: yajiema@nuaa.edu.cn

Abstract This paper presents a distributed allocation algorithm based on a coalition formation game to solve the complex task allocation problem when heterogeneous resources are carried by large-scale UAV swarm and tasks have heterogeneous requirements. Firstly, in response to the large scale of the cluster and the heterogeneity of resources, the K-medoids clustering method is improved by incorporating consideration of heterogeneous resources to reasonably reduce the scale of the original problem. Therefore, the allocation difficulty is reduced through preprocessing. Considering task requirements, resources carried by UAV, and path costs, a task allocation model is established and the original problem is transformed into a coalition partitioning problem. Then a distributed task allocation algorithm based on a coalition formation game is used to solve it. Finally, the simulation results of assigning 30 tasks with heterogeneous requirements to 100 unmanned aerial vehicles carrying 3 types of heterogeneous resources show that the proposed algorithm can improve the real-time performance of task allocation, reduce resource redundancy, and fully leverage cluster efficiency.

Keywords task allocation, heterogeneous UAV swarms, heterogeneous resources, clustering algorithm, coalition formation game