

控制与决策

Control and Decision

基于顺序扩展一致性包算法的多无人机分布式任务分配

高程, 都延丽, 步雨浓, 刘燕斌, 杨小草

引用本文:

高程, 都延丽, 步雨浓, 刘燕斌, 杨小草. 基于顺序扩展一致性包算法的多无人机分布式任务分配[J]. *控制与决策*, 2023, 38(11): 3242–3250.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.2178>

您可能感兴趣的其他文章

Articles you may be interested in

基于动态蚁群劳动分工模型的多AUV任务分配方法

A multi-AUV dynamic task allocation method based on antcolony labor division model

控制与决策. 2021, 36(8): 1911–1919 <https://doi.org/10.13195/j.kzyjc.2019.1312>

多无人机协同直播场景下自适应任务卸载决策

Adaptive task offloading decision of multi-UAVs cooperation in live broadcasting scenario

控制与决策. 2021, 36(4): 974–982 <https://doi.org/10.13195/j.kzyjc.2019.1104>

基于两阶段迭代优化的空天观测资源协同任务规划方法

A two-stage iterative optimization method for the coordinated task planning of space and air observation resources

控制与决策. 2021, 36(5): 1147–1156 <https://doi.org/10.13195/j.kzyjc.2019.1193>

多无人机协同直播场景下自适应任务卸载决策

Adaptive task offloading decision of multi-UAVs cooperation in live broadcasting scenario

控制与决策. 2021, 36(4): 974–982 <https://doi.org/10.13195/j.kzyjc.2019.1104>

面向多目标侦察任务的无人机航线规划

UAV trajectory planning for multi-target reconnaissance missions

控制与决策. 2021, 36(5): 1191–1198 <https://doi.org/10.13195/j.kzyjc.2019.1284>

基于顺序扩展一致性包算法的多无人机分布式任务分配

高程¹, 都延丽¹, 步雨浓^{2†}, 刘燕斌¹, 杨小草¹

(1. 南京航空航天大学 航天学院, 南京 211106; 2. 北京机电工程研究所, 北京 100074)

摘要: 以异构多无人机协同执行复杂的耦合多任务为背景, 提出一种求解分布式任务分配问题非死锁的顺序扩展一致性包算法. 首先, 建立考虑任务载荷资源、任务时序、威胁区等约束条件的时序多任务分配模型; 其次, 对一致性包算法的任务包构建过程和冲突消解规则进行扩展, 并设计一种基于有向图深度优先搜索的方法进行任务方案的死锁检测和修正, 以实现无冲突和无死锁的任务分配; 然后, 将关联任务之间的时序约束转化为软时间窗约束, 利用顺序分层的策略进行求解; 最后, 为了提高任务分配结果的可靠性, 采用 Dubins 曲线路径将航路规划耦合到任务分配中. 仿真实验表明, 所提出的算法能够快速有效地求解异构多无人机分布式耦合多任务分配问题, 具备良好的最优性和时效性.

关键词: 多无人机; 多任务分配; 一致性包算法; 分布式; 无死锁; 航路规划

中图分类号: TP29

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.2178

开放科学(资源服务)标识码(OSID):



引用格式: 高程, 都延丽, 步雨浓, 等. 基于顺序扩展一致性包算法的多无人机分布式任务分配[J]. 控制与决策, 2023, 38(11): 3242-3250.

Distributed task allocation of multiple UAVs based on sequential extended consensus based bundle algorithm

GAO Cheng¹, DU Yan-li¹, BU Yu-nong^{2†}, LIU Yan-bin¹, YANG Xiao-cao¹

(1. School of Aerospace, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; 2. Beijing Electromechanical Engineering Research Institute, Beijing 100074, China)

Abstract: Based on the background of complex coupled multiple tasks executed by heterogeneous multiple unmanned aerial vehicles (UAVs), a deadlock-free sequentially extended consensus based bundle algorithm is presented to solve the distributed task allocation problem. Firstly, a timing multi-task allocation model is established in consideration of the constraints of task load resources, task sequence and threat regions. Secondly, the task bundle construction and conflict resolution rules of the consensus based bundle algorithm are extended, and then a deadlock detection and correction method of the task scheme based on digraph depth-first search is designed to achieve conflict-free and deadlock-free task allocation. Furthermore, the timing constraints between related tasks are transformed into soft time window constraints, and the sequential hierarchical strategy is utilized to solve the problem. Finally, the Dubins curve path is employed to integrate the route planning into the task allocation in order to improve the reliability of the task allocation result. Simulation results show that the proposed algorithm can tackle the distributed coupled multi-task allocation problem of heterogeneous multiple UAVs quickly and efficiently, and possesses satisfactory optimality and timeliness.

Keywords: multiple UAVs; multi-task allocation; consensus based bundle algorithm; distributed; deadlock-free; route planning

0 引言

现代战争的复杂多样性使得无人机的作战样式正在从单机作战向多机协同作战的方向转变, 合理的任务分配是提高多无人机协同执行任务系统效能的关键^[1]. 多无人机任务分配问题可以看作是复杂多

约束条件下的组合优化问题, 需要结合特定的任务场景对其进行建模和求解^[2]. 典型的任务分配模型有多旅行商问题(MTSP)^[3]、车辆路由问题(VRP)^[4]、混合整数线性规划(MILP)模型^[5]、协同多任务分配问题(CMTAP)^[6]等. 针对任务分配模型的求解方法主要

收稿日期: 2021-12-17; 录用日期: 2022-05-17.

基金项目: 国防重点实验室基金项目(202005).

责任编辑: 董久祥.

†通讯作者. E-mail: htkg_buyunong@126.com.

*本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

分为集中式和分布式^[7]两种。

集中式的任务分配方法主要包括最优化方法和启发式方法,在任务分配中的应用非常广泛。文献[8]采用改进的遗传算法求解了侦察、打击、毁伤效果评估一体化的CMTAP问题。文献[9]提出了一种基于禁忌表策略的两阶段迭代优化方法,以解决空天观测资源协同任务规划问题。集中式的方法虽然寻优能力较强,但在问题规模较大时易陷入局部最优,中心节点计算负荷较大,系统的鲁棒性较差。

分布式的任务分配方法大多基于市场机制,不依靠中心节点进行计算,系统的可扩展性和鲁棒性更好^[10]。文献[11]将一致性算法的收敛性与拍卖方法的鲁棒性和计算效率相结合,提出了一致性包算法(consensus based bundle algorithm, CBBA),通过不断迭代任务包构建和一致性冲突消解两个阶段得到无冲突的任务分配方案。CBBA算法采用的非负评分方案,满足边际增益递减(DMG)的收敛性,相对于最优目标值可达到至少50%的最优性^[12]。文献[13]讨论了异步CBBA在不完全连通通信条件下的性能。尽管CBBA为分布式任务分配问题提供了一个框架,但它不能直接用于处理复杂任务。文献[14]针对搜救场景提出了一种具有任务耦合约束的CBBA扩展算法(CBBA-TCC),通过引入外部一致性阶段,以解决具有任务耦合约束的异构多无人机任务分配问题。文献[15-16]将扩展一致性包算法与动态任务分配策略相结合,以生成对突发新任务快速响应的可靠任务重规划解决方案。

综合国内外研究现状可知,分布式的任务分配方法计算效率高,具有较好的可扩展性和鲁棒性,已成为近年来国内外学者们研究的热点。但现有的分布式任务分配研究中,大多针对简单场景下的独立任务,未考虑任务之间的时序耦合关系,对分布式时序任务的研究也未考虑多无人机同时执行的任务及其带来的死锁问题;对分布式多约束任务分配问题的研究不够深入,任务分配与航路规划异步进行,可能导致任务分配结果的可靠性降低。

本文针对复杂多约束的分布式时序多任务分配问题,设计一种非死锁的顺序扩展一致性包算法(sequential extended consensus based bundle algorithm, SECBBA)的分布式任务分配方法。对一致性包算法的任务包构建和冲突消解规则进行扩展,并设计一种基于有向图深度优先搜索的方法进行死锁检测和修正,以实现多无人机执行型任务的无冲突和无死锁的任务分配;将任务之间的时序约束转化为软时间窗约束,利用顺序分层的策略进行求解;将航路规划耦

合到任务分配中,提高任务分配结果的可靠性;最后,通过仿真实验验证了该方法的有效性。

1 问题描述和建模

1.1 任务场景

本文研究异构多无人机协同执行复杂耦合多任务的任务分配问题,属于CMTAP问题。假设在任务范围中共有 N_T 个目标区域 $\mathbf{T} = \{T_1, T_2, \dots, T_{N_T}\}$ 、 N_V 架无人机 $\mathbf{U} = \{U_1, U_2, \dots, U_{N_V}\}$ 和 N_W 个威胁区 $\mathbf{W} = \{W_1, W_2, \dots, W_{N_W}\}$,由多无人机绕开威胁区对每个目标区域依次执行侦察(C)、打击(A)、毁伤效果评估(V)任务,将同一目标区域中的不同任务称为关联任务。总任务集表示为 $\mathbf{M} = \{\mathbf{M}_C, \mathbf{M}_A, \mathbf{M}_V\}$,侦察任务集表示为 $\mathbf{M}_C = \{M_1, M_2, \dots, M_{N_T}\}$,其他任务集的表示类似。将无人机 i 能够执行的任务集定义为 \mathbf{J} ,能够执行任务 $j \in \mathbf{J}$ 的无人机集定义为 \mathbf{I} 。

假设 N_V 架无人机分为3类,类型1和类型3的无人机携带侦察探测类载荷资源,分别用于执行侦察任务和评估任务,并且载荷资源在执行任务后不发生变化;类型2的无人机携带有限数量的机载弹药类资源,用于执行打击任务,这类载荷资源属于消耗型资源,执行任务后载荷资源将减少。每个区域的侦察和打击任务由单架或多架无人机来执行,毁伤效果评估任务由单架无人机即可完成。为了提高任务成功的概率,由多架无人机执行的任务设定为同时进行,例如同时侦察、同时打击任务。

1.2 约束条件和目标函数

为了方便地描述各项约束条件,引入决策变量

$$x_{ij} = \begin{cases} 1, & \text{无人机}i\text{执行任务}j; \\ 0, & \text{无人机}i\text{不执行任务}j. \end{cases} \quad (1)$$

对于异构多无人机协同执行察打评任务的任务分配问题,需要考虑的约束条件有

$$\sum_{i=1}^{N_V} x_{ij} = \text{req}_j, \forall j \in \mathbf{J}; \quad (2)$$

$$\sum_{j=1}^{N_T} x_{ij} \leq L_i, \forall i \in \mathbf{I}; \quad (3)$$

$$t_{\text{end}}(M_C^j) \leq t_{\text{start}}(M_A^j), t_{\text{end}}(M_A^j) \leq t_{\text{start}}(M_V^j); \quad (4)$$

$$t_{ij} = t_{kj}, \forall x_{ij}x_{kj} = 1; \quad (5)$$

$$\text{dis}(i) \leq L_{\text{max}}(i), \forall U_i \in \mathbf{U}; \quad (6)$$

$$v(i) \leq v_{\text{max}}(i), r(i) \geq r_{\text{min}}(i), \forall U_i \in \mathbf{U}; \quad (7)$$

$$\text{path}(i, j) \cap \mathbf{W} = \emptyset, \forall i, j \in \mathbf{T} \cup \mathbf{U}. \quad (8)$$

式(2)为任务资源约束,不同的任务对无人机载

荷资源的需求不同;式(3)为任务包上限约束,无人机可执行任务的数量存在上限;式(4)为任务时序约束,即对每个目标区域依次进行侦察、打击和毁伤评估任务;式(5)为任务开始时间约束,执行同一任务的各无人机的任务开始时间相同;式(6)为无人机的最大航程约束;式(7)为无人机运动学约束,主要考虑无人机的最大飞行速度和最小转弯半径;式(8)为威胁区约束,无人机执行任务时要避开威胁区。

以最小化整体任务完成时间和各无人机执行任务的平均时间为目标,构建全局的目标函数如下:

$$F = \alpha \max_{i \leq N_V} \left(\sum_{j=1}^{N_T} t_{ij}^f x_{ij} \right) + \beta \frac{1}{N_V} \sum_{i=1}^{N_V} \sum_{j=1}^{N_T} t_{ij}^f x_{ij}. \quad (9)$$

其中: α 和 β 是权重系数,具体取值根据需求确定; t_{ij}^f 为无人机 i 完成任务 j 所需的时间,计算方法为

$$t_{ij}^f = t_{ij}^{\text{dis}} + t_{ij}^{\text{wait}} + t_{ij}^{\text{dur}}. \quad (10)$$

t_{ij}^{dis} 为无人机从前序任务点到任务 j 的最短时间, t_{ij}^{dur} 为执行任务 j 所需的时间, t_{ij}^{wait} 为等待时间。

2 顺序扩展一致性包算法

本文提出的顺序扩展一致性包算法包括4个阶段:任务包构建、一致性冲突消解、死锁检测和修正以及关联任务的顺序分层求解。算法流程如图1所示。

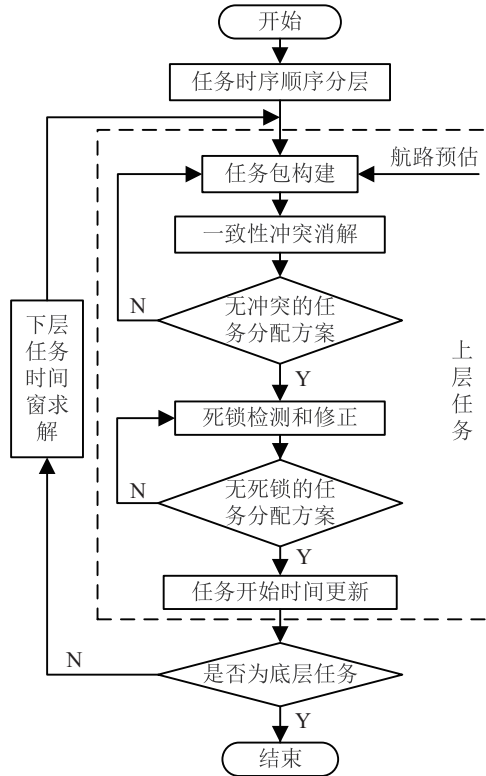


图1 顺序扩展一致性包算法流程

2.1 任务包构建

根据文献[17],在任务包构建阶段,各无人机同时在本地采用DMG评分机制选择任务加入自己的

任务包列表 \mathbf{b}_i 中,并生成任务路径列表 \mathbf{p}_i 、获胜者矩阵 \mathbf{Z}^i 和获胜投标值矩阵 \mathbf{Y}^i 。每一轮选择任务时,无人机首先对满足约束条件的所有任务计算边际增益作为投标值,边际增益 c_{ij} 的计算方法为

$$c_{ij} = \max_{n \leq |\mathbf{p}_i|} S_i^{\mathbf{p}_i \oplus_n \{j\}} - S_i^{\mathbf{p}_i}, \forall j \in \mathbf{J} \setminus \mathbf{b}_i. \quad (11)$$

其中: $\mathbf{p}_i \oplus_n \{j\}$ 表示将任务 j 插入 \mathbf{p}_i 中的第 n 位生成的新路径列表, $\mathbf{J} \setminus \mathbf{b}_i$ 表示从任务集 \mathbf{J} 中去掉 \mathbf{b}_i 中已有的任务, $S_i^{\mathbf{p}_i}$ 表示无人机 i 沿 \mathbf{p}_i 执行任务的总收益。 $S_i^{\mathbf{p}_i}$ 的计算方法为

$$S_i^{\mathbf{p}_i} = \sum_{j \in \mathbf{p}_i} (\text{rwd}_{ij} - \text{cst}_{ij}). \quad (12)$$

rwd_{ij} 为无人机 i 执行任务 j 的收益,采用固定收益和时间折扣收益结合的形式进行计算,即

$$\text{rwd}_{ij} = (R_{j0} + e^{-\lambda_j(t_{ij} - \tau_j^{\text{start}})} R_j) u(t_{ij}). \quad (13)$$

其中: R_{j0} 为执行任务的固定收益; R_j 为时间折扣收益的基础收益; $0 < \lambda_j < 1$ 为任务 j 的时间折扣系数; $u(t_{ij})$ 为与执行任务的时间有关的决策变量,有

$$u(t_{ij}) = \begin{cases} 1, & \tau_j^{\text{start}} \leq t_{ij} \leq \tau_j^{\text{end}} - \tau_j^{\Delta}; \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

其中: $[\tau_j^{\text{start}}, \tau_j^{\text{end}}]$ 为任务 j 的时间窗, τ_j^{Δ} 为任务 j 的耗时。式(12)中 cst_{ij} 为执行任务的代价,计算方法为

$$\text{cst}_{ij} = \gamma_i d_{ij}. \quad (15)$$

其中: γ_i 为无人机 i 的航程代价系数, d_{ij} 为无人机 i 执行任务 j 的航程代价,采用带有避障的Dubins曲线路径进行预估,具体方法将在后文中给出。

在计算得到所有任务的投标值后,各无人机选择边际增益最高的任务加入任务包中。具体的任务包构建算法如下。

算法1 任务包构建。

输入: $\mathbf{b}_i(t-1), \mathbf{p}_i(t-1), \mathbf{Z}^i(t-1), \mathbf{Y}^i(t-1)$;

输出: $\mathbf{b}_i(t), \mathbf{p}_i(t), \mathbf{Z}^i(t), \mathbf{Y}^i(t)$ 。

1) while $|\mathbf{b}_i(t-1)| \leq L_i$ do

2) $c_{ij} = \max_{n \leq |\mathbf{p}_i|} S_i^{\mathbf{p}_i \oplus_n \{j\}} - S_i^{\mathbf{p}_i}, \forall j \in \mathbf{J} \setminus \mathbf{b}_i$;

3) for $\forall j \in \mathbf{J}$

4) $h_{ij} = 0$;

5) if $\text{req}_j = 1$ then

6) $h_{ij} = I\left(c_{ij} > \sum_{k=1}^{N_V} y_{kj}^i\right)$;

7) elseif $\text{req}_j > 1$ then

8) if $\text{req}_j > \sum_{k=1}^{N_V} z_{kj}^i$ then

9) $h_{ij} = 1$;

10) elseif $c_{ij} > \min(y_{kj}^i)$ then

- 11) $h_{ij} = 1;$
- 12) end if
- 13) end if
- 14) end for
- 15) $J_i = \arg \max_j (c_{ij} \times h_{ij});$
- 16) $n_{i,J_i} = \arg \max_n S_i^{p_i \oplus n \{J_i\}};$
- 17) $\mathbf{b}_i(t) = \mathbf{b}_i(t-1) \oplus_{\text{end}} \{J_i\};$
- 18) $\mathbf{p}_i(t) = \mathbf{p}_i(t-1) \oplus_{n_{i,J_i}} \{J_i\};$
- 19) $z_{i,J_i}^i(t) = 1, y_{i,J_i}^i(t) = c_{i,J_i};$
- 20) end while

算法1中, $h_{ij} = 1$ 表示无人机 i 可以对任务 j 投标, $I(\cdot)$ 为条件符号, 条件满足时为1, 否则为0. 首先求取无人机 i 对所有可投标任务的投标值 c_{ij} (第2行). 当 $\text{req}_j = 1$ 时, 无人机 i 只需将自己的投标值与任务 j 的获胜投标值作比较来决定是否投标 (第3行~第6行). 当 $\text{req}_j > 1$ 时, 需要判断竞标的无人机数量是否已满足任务 j 的需求. 若 $\text{req}_j > \sum_{k=1}^{N_V} z_{kj}^i$, 则表明无人机 i 认为任务 j 未分配满荷, 此时直接进行投标. 若 $\text{req}_j = \sum_{k=1}^{N_V} z_{kj}^i$, 则表明无人机 i 认为任务 j 已分配满荷, 无人机 i 需要将自己的投标值与任务 j 的最小获胜投标值作比较来决定是否投标 (第7行~第14行). 在比较投标值时, 若投标值相同, 则选择编号较小的无人机作为获胜者. 一轮投标完成后, 无人机从所有投标的任务中选择投标值最大的任务加入自己的任务包中, 并更新路径列表和获胜信息 (第15行~第19行). 各无人机不断进行任务包的构建直到任务包达到上限或者已无满足约束条件的任务可选.

2.2 一致性冲突消解

任务包构建阶段由各无人机在本地基于局部信息完成, 因此需要各无人机之间通过数据交换进行冲突消解, 以获得全局无冲突的任务分配方案. 本文在原始CBBA算法共识规则的基础上进行了扩展, 以实现多无人机执行型任务的无冲突分配. 假设各无人机在一致性阶段采用同步通信机制, 基于共识的冲突消解算法如下.

算法2 一致性冲突消解.

输入: $\mathbf{Z}^i(t-1), \mathbf{Y}^i(t-1), \mathbf{s}_i(t-1);$

输出: $\mathbf{Z}^i(t), \mathbf{Y}^i(t), \mathbf{s}_i(t).$

- 1) 无人机 i 接收无人机 k 发送的 $\mathbf{Z}^k(t-1), \mathbf{Y}^k(t-1)$ 和 $\mathbf{s}_k(t-1);$
- 2) for $\forall j \in \mathbf{J}$
- 3) if $\text{req}_j = 1$ then
- 4) $\mathbf{Z}^i \rightarrow \mathbf{z}_i, \mathbf{Y}^i \rightarrow \mathbf{y}_i;$

5) 按照原始CBBA的共识规则进行一致性冲突消解;

6) $\mathbf{z}_i \rightarrow \mathbf{Z}^i, \mathbf{y}_i \rightarrow \mathbf{Y}^i;$

7) elseif $\text{req}_j > 1$ then

8) for $\forall m \in \mathbf{I}$ where $z_{mj}^i = 1$

9) if $k = m \ \& \ s_{km} > s_{im}$ then

10) $z_{mj}^i = z_{mj}^k, y_{mj}^i = y_{mj}^k;$

11) end if

12) end for

13) for $\forall m \in \mathbf{I}$ where $z_{mj}^k = 1$

14) if $m \neq i \ \& \ z_{mj}^i = 0 \ \& \ s_{km} > s_{im}$ then

15) if $\text{req}_j > \sum_{m=1}^{N_V} z_{mj}^i$ then

16) $z_{mj}^i = z_{mj}^k, y_{mj}^i = y_{mj}^k;$

17) elseif $\min(y_{nj}^i \forall n) < y_{mj}^k$ then

18) $z_{nj}^i = 0, z_{mj}^i = z_{mj}^k;$

19) $y_{nj}^i = 0, y_{mj}^i = y_{mj}^k;$

20) end if

21) end if

22) end for

23) end if

24) $s_{ik} = t;$

25) end for

算法2中, 无人机 i 首先接收来自相邻无人机 k 的获胜信息和时间戳 (第1行). 对于任务 j , 若 $\text{req}_j = 1$, 则将获胜信息矩阵还原为列表, 按照原始CBBA的共识规则^[11]进行冲突消解 (第3行~第6行); 若 $\text{req}_j > 1$, 则冲突消解分为两个阶段. 第1阶段, 接收方 i 将已存储的获胜信息与发送方 k 进行比较, 以确认这些信息为当前最新 (第8行~第12行). 第2阶段, 接收方从发送方获取新信息并与自己的信息合并. 若发送方 k 认为无人机 $m (m \neq i)$ 执行任务 j , 且接收方 i 中没有此信息时, 接收方 i 需要判断任务 j 分配是否满荷, 若未满荷, 则进行更新; 满荷则需要比较 \mathbf{Y}^i 第 j 列的非零最小投标值与 y_{mj}^k 的值, 当无人机 m 的投标值更大时, 将最小投标值的无人机替换为无人机 m , 并更新获胜信息 (第13行~第22行). 同样的, 当投标值相同时, 选择编号较小的无人机.

在一致性协商结束后, 每个无人机都会检查自己的获胜信息与任务包信息是否冲突, 如果冲突则释放任务包中的冲突任务及其后续任务, 然后进行新一轮的任务包构建和冲突消解.

2.3 死锁检测和修正

通过任务包构建和一致性冲突消解两个阶段的循环迭代, 可以得到满足载荷资源需求的侦察任务分

配方案. 本文中假设同一目标区域的侦察任务同时开始执行, 可能存在“死锁”问题, 即不同无人机执行同一任务时, 出现无限循环等待的现象. 因此有必要对得到的任务分配结果进行死锁检测和修正.

采用有向图 $G = \langle V, E \rangle$ 的方式描述死锁问题, 本文中的死锁只与多无人机执行型任务有关, 因此节点 V 设为仅考虑 $\text{req}_j > 1$ 的任务节点, E 为有向图的边集, E 中的边线 (i, j) 表示存在无人机从节点 i 到节点 j 执行任务. 根据文献[18], G 中的回路表示死锁, 任务序列无死锁等价于有向图 G 为有向无环图 (directed acyclic graph, DAG). 本文采用深度优先搜索 (DFS) 算法对有向图的回路进行检测, 若存在回路, 则采用最近邻原则对任务序列进行修正, 以消除死锁. 死锁的检测和修正算法如下.

算法3 死锁检测和修正.

输入: 无冲突的任务分配结果;

输出: 无死锁的任务分配结果.

- 1) 构建任务序列有向图 $G = \langle V, E \rangle$;
- 2) 利用 DFS 算法检测 G 中是否存在回路, 回路信息保存到 loop 中;
- 3) while loop $\neq \emptyset$ do
- 4) $(i, j) = \arg \min_{i,j} (E_{ij}), \forall i, j \in \text{loop} \& i > j$;
- 5) for $\forall k \in I$
- 6) if $(i, j) \in p_k$ then
- 7) 交换 p_k 中 i 和 j 的位置;
- 8) end if
- 9) end for
- 10) 更新有向图 G ;
- 11) 利用 DFS 算法检测回路, 更新 loop;
- 12) end while

算法3中, 首先构建多无人机执行型任务节点的有向图 G , 并利用 DFS 算法检测 G 中是否存在回路 (第1行和第2行). 若存在回路, 则利用最近邻原则找到交换代价最小的两个节点 i 和 j , 按照编号的大小进行排序, 找到先执行 i 后执行 j 任务的无人机 k , 交换其路径列表中 i 和 j 的位置 (第3行~第9行). 更新有向图 G , 然后重新检测是否存在回路, 若存在回路, 则继续修正, 直到整个任务方案无死锁 (第10行~第12行). 由于死锁的修正过程中仅进行任务执行顺序的调整, 修正后的方案仍然是满足载荷资源约束的方案.

2.4 顺序分层求解关联任务

由于关联任务之间存在时序约束关系, 本文将任务之间的时序约束转化为软时间窗, 然后采用顺序分层的策略按照时序进行求解. 对于优先级相同的任

务 (例如各区域的侦察任务), 同时进行求解, 对于优先级不同的任务, 按照时序约束顺序依次求解, 将上层任务的结束时间作为下层任务的开始时间, 将时序约束转化为软时间窗约束. 在构建软时间窗之前, 首先需要对上层任务的开始时间进行更新, 以满足多无人机执行型任务同时开始的约束. 上层任务开始时间更新的算法如下.

算法4 上层任务开始时间更新.

输入: 未更新的任务开始时间 times_old;

输出: 更新后的任务开始时间 times_new.

- 1) while flag = 1 do
- 2) 更新 p_i 中各任务的开始时间 t_{latest_j} ;
- 3) 求取 p_i 中任务原开始时间与 t_{latest_j} 的差值 $t_{\text{delta}_{i,m}}$;
- 4) if $\sum t_{\text{delta}_{i,m}} > 0$ then
- 5) flag = 1;
- 6) $t_{\text{delta}_{i,m}} = \max(t_{\text{delta}_{i,m-1}}, t_{\text{delta}_{i,m}})$;
- 7) $\text{times_new}_{i,m} = \text{times_old}_{i,m} + t_{\text{delta}_{i,m}}$;
- 8) elseif $\sum t_{\text{delta}_{i,m}} = 0$ then
- 9) flag = 0;
- 10) end if
- 11) end while

算法4中, 首先以任务 j 最晚到达的无人机的预估到达时间为基准, 更新当前任务的最新开始时间 t_{latest_j} , 并求得 p_i 中任务原开始时间与最新开始时间的差值 $t_{\text{delta}_{i,m}}$ (第1行~第3行). 然后综合前序任务的差值与当前任务的差值对 $t_{\text{delta}_{i,m}}$ 进行更新, 并计算更新后的任务开始时间 (第4行~第7行). 由于任务分配方案中不存在死锁, 在迭代有限次后, $t_{\text{delta}_{i,m}}$ 中的值全为0, 表示算法已收敛 (第8行~第11行). 在更新所有上层侦察任务的开始时间后, 将各区域下层打击任务的开始时间替换为上层任务的结束时间, 构建同一区域的关联打击任务的软时间窗, 以满足侦察和打击任务之间的时序约束. 接着对打击任务按照扩展 CBBA 算法进行求解, 并进行死锁检测和修正. 最后, 按照同时打击的约束更新打击任务的开始时间, 构建评估任务的软时间窗, 并进行评估任务的无冲突分配. 由于评估任务均为单无人机执行型任务, 因此无需进行死锁检测和开始时间更新.

2.5 耦合航路规划

对于存在威胁区的任务环境, 后期航路规划得到的航程与欧式距离之间可能存在较大误差, 因此, 本文将 Dubins 曲线路径规划耦合到任务分配中, 以提高任务分配结果的可靠性. 有直线段的 Dubins 曲线

可表示为 $D = \{LSL, RSR, RSL, LSR\}$, 其中 L 或 R 表示逆或顺时针的圆弧, S 为直线. 本文中采用欧几里得原理构造 CLC 直线^[19]的方法求解 Dubins 路径的长度, 半径不同的 Dubins 路径待求解各变量如图 2 所示, 图 2 为 RSR 型曲线.

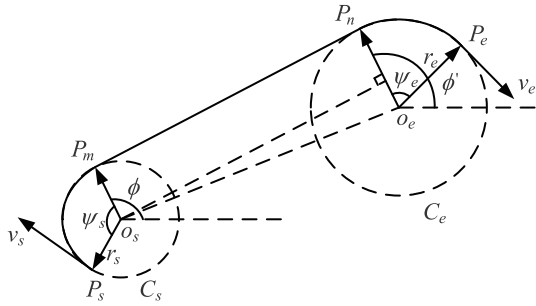


图 2 Dubins 路径

给定起始姿态点 $P_s(x_s, y_s, \phi_s)$ 和终止姿态点 $P_e(x_e, y_e, \phi_e)$, 则起始圆 C_s 和结束圆 C_e 的圆心 $O_s(x_{cs}, y_{cs})$ 和 $O_e(x_{ce}, y_{ce})$ 可以由下式求得:

$$\begin{aligned} x_{cs} &= x_s - r_s \cos(\phi_s \pm \pi/2), \\ y_{cs} &= y_s - r_s \sin(\phi_s \pm \pi/2), \end{aligned}$$

$$\begin{aligned} x_{ce} &= x_e - r_e \cos(\phi_e \pm \pi/2), \\ y_{ce} &= y_e - r_e \sin(\phi_e \pm \pi/2). \end{aligned} \quad (16)$$

其中: \pm 取决于运动方向是顺时针还是逆时针, 顺时针是 +, 逆时针是 -. 然后按照表 1 中的规则求解起始圆的 ϕ 和终止圆的 ϕ' , $\text{mod}(\cdot)$ 为取余运算符, $\text{sign}(\cdot)$ 为符号函数, 由此计算圆弧段的角度 ψ_s 和 ψ_e 为

$$\begin{aligned} \psi_s &= \text{mod}\left(2\pi + \frac{\pi}{2} \pm (\phi_s - \phi), 2\pi\right), \\ \psi_e &= \text{mod}\left(2\pi + \frac{3\pi}{2} \pm (\phi_e - \phi'), 2\pi\right). \end{aligned} \quad (17)$$

其中: \pm 取决于运动方向是顺时针还是逆时针, 顺时针是 +, 逆时针是 -. 然后计算圆 C_s 和 C_e 上切点 P_m 和 P_n 的位置, 有

$$\begin{aligned} x_{P_m} &= x_{cs} + r_s \cos \phi, \quad y_{P_m} = y_{cs} + r_s \sin \phi; \\ x_{P_n} &= x_{ce} + r_e \cos \phi', \quad y_{P_n} = y_{ce} + r_e \sin \phi'. \end{aligned} \quad (18)$$

最后, 可以求得整个 Dubins 路径的长度为

$$L_{\text{Dubins}} = r_s \psi_s + \sqrt{(x_{P_m} - x_{P_n})^2 + (y_{P_m} - y_{P_n})^2} + r_e \psi_e. \quad (19)$$

表 1 ϕ 和 ϕ' 的求解规则

曲线	起始圆的 ϕ	终止圆的 ϕ'
RSR 型	$\text{mod}\left(\arcsin \frac{r_e - r_s}{\sqrt{(x_{cs} - x_{ce})^2 + (y_{cs} - y_{ce})^2}} + \arctan\left(\frac{y_{ce} - y_{cs}}{x_{ce} - x_{cs}}\right) + \text{sign}(x_{ce} - x_{cs})\frac{\pi}{2} + 2\pi, 2\pi\right)$	ϕ
LSL 型	$\text{mod}\left(-\arcsin \frac{r_e - r_s}{\sqrt{(x_{cs} - x_{ce})^2 + (y_{cs} - y_{ce})^2}} + \arctan\left(\frac{y_{ce} - y_{cs}}{x_{ce} - x_{cs}}\right) + \text{sign}(x_{ce} - x_{cs})\frac{\pi}{2} + 2\pi, 2\pi\right)$	ϕ
RSL 型	$\text{mod}\left(-\arcsin \frac{r_e + r_s}{\sqrt{(x_{cs} - x_{ce})^2 + (y_{cs} - y_{ce})^2}} + \arctan\left(\frac{y_{ce} - y_{cs}}{x_{ce} - x_{cs}}\right) + \text{sign}(x_{ce} - x_{cs})\frac{\pi}{2} + 2\pi, 2\pi\right)$	$\text{mod}(\phi + \pi, 2\pi)$
LSR 型	$\text{mod}\left(\arcsin \frac{r_e + r_s}{\sqrt{(x_{cs} - x_{ce})^2 + (y_{cs} - y_{ce})^2}} + \arctan\left(\frac{y_{ce} - y_{cs}}{x_{ce} - x_{cs}}\right) + \text{sign}(x_{ce} - x_{cs})\frac{\pi}{2} + 2\pi, 2\pi\right)$	$\text{mod}(\phi + \pi, 2\pi)$

3 仿真实验及分析

本节在典型的协同多任务场景下, 通过仿真实验验证所提出的 SECBBA 算法的有效性, 并与 CBBA-TCC 算法以及改进遗传算法 (IGA) 进行算法性能的对比. 设定各无人机之间为全连通通信网络. 仿真平台为具有 Intel Core i5-6300HQ 2.30 GHz 处理器和 8 G 内存的 PC 机, 软件环境为 MATLAB R2019a.

3.1 典型场景任务分配结果

任务场景设定为在 $50 \text{ km} \times 50 \text{ km}$ 的范围内有 10 个目标区域和 4 个威胁区, 分配携带 3 种不同载荷的 16 架无人机避开威胁区到目标区域依次执行侦察任务、打击任务和毁伤效果评估任务. 无人机的参数如表 2 所示, 3 种类型无人机的最大航程均为 100 km, 最大速度分别为 50 m/s、45 m/s、50 m/s, 最小转弯半径分别为 200 m、220 m、200 m. 目标区域的参数如表 3 所示, 威胁区的参数如表 4 所示.

表 2 无人机参数

编号	载荷资源	位置/km	编号	载荷资源	位置/km
1	[1, 0, 0]	(2, 7)	9	[0, 3, 0]	(19, 6)
2	[1, 0, 0]	(14, 5)	10	[0, 3, 0]	(35, 4)
3	[1, 0, 0]	(22, 8)	11	[0, 3, 0]	(38, 1)
4	[1, 0, 0]	(26, 7)	12	[1, 3, 0]	(48, 2)
5	[1, 0, 0]	(30, 6)	13	[0, 0, 1]	(12, 0)
6	[1, 0, 0]	(41, 6)	14	[0, 0, 1]	(19, 0)
7	[0, 3, 0]	(6, 2)	15	[0, 0, 1]	(32, 0)
8	[0, 3, 0]	(10, 1)	16	[0, 0, 1]	(43, 0)

表 3 目标区域参数

编号	载荷需求	位置/km	编号	载荷需求	位置/km
1	[2, 1, 1]	(3, 24)	6	[2, 2, 1]	(28, 22)
2	[2, 2, 1]	(8, 37)	7	[1, 1, 1]	(34, 13)
3	[2, 1, 1]	(17, 12)	8	[2, 2, 1]	(39, 27)
4	[2, 1, 1]	(20, 26)	9	[2, 1, 1]	(43, 13)
5	[1, 1, 1]	(26, 45)	10	[1, 1, 1]	(47, 38)

表4 威胁区参数

编号	位置/km	半径/km	编号	位置/km	半径/km
1	(11, 20)	5	3	(27, 14)	3
2	(21, 35)	4	4	(36, 38)	4

执行任务的固定收益设为5,航程代价系数设为0.1,侦察、打击、评估任务时间折扣收益的基础收益分别设为10、5、5,时间折扣系数分别设为0.05、0.1、0.1,任务耗时分别设为30s、25s、20s.任务分配的结果如表5所示,其中,任务路径1和任务路径2分别为死锁修正前后的任务路径.在初始的任务分配结果中,无人机3和4的任务路径存在死锁冲突,导致目标区域4和6的侦察任务无法同时执行.死锁修正前后各无人机执行侦察任务的时序对比如图3所示;经过算法3的死锁检测和修正后,无人机4的任务路径调整为“4,6,2”,调整后的任务路径无死锁,采用算法4

表5 任务分配结果

无人机编号	任务路径1	任务路径2	任务开始时间/s
1	1, 2, 5	1, 2, 5	574.5, 1207.6, 1631.5
2	3, 1	3, 1	152.3, 574.5
3	3, 4, 6	3, 4, 6	152.3, 468.7, 677.6
4	6, 4, 2	4, 6, 2	468.7, 677.6, 1207.6
5	7, 9, 8	7, 9, 8	161.2, 371.2, 692.4
6	9, 8, 10	9, 8, 10	371.2, 692.4, 994.5
7	1, 2	1, 2	604.5, 1237.6
8	2	2	1237.6
9	3, 6, 8	3, 6, 8	182.3, 707.6, 1001.1
10	6, 4, 5	6, 4, 5	707.6, 931.3, 1661.5
11	7, 8	7, 8	281.1, 1001.1
12	9, 10	9, 10	401.2, 1024.5
13	1, 2, 5	1, 2, 5	629.5, 1262.6, 1686.5
14	3, 6, 4	3, 6, 4	243.3, 732.6, 956.3
15	7, 8	7, 8	306.1, 1026.1
16	9, 10	9, 10	426.2, 1049.5

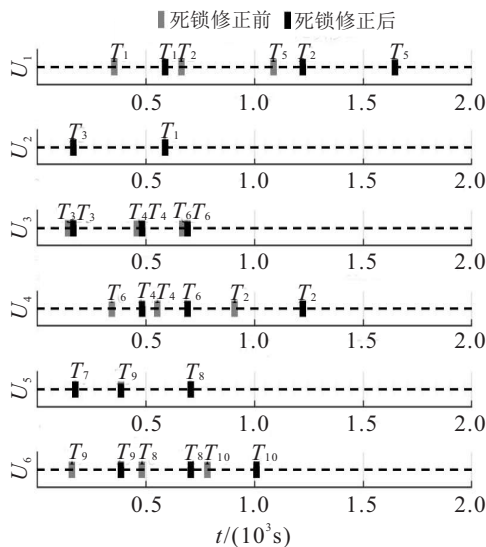


图3 死锁修正前后各无人机侦察任务序列对比

进行任务开始时间更新后,各目标区域的侦察任务同时执行.算法3的求解平均耗时为0.002s,具有较好的时效性.任务分配结果的三维时空和二维平面的示意图分别如图4和图5所示.由任务分配结果的相关图表可以看出,同一目标区域的侦察、打击和评估任务之间满足时序约束,任务分配结果无死锁,同时在任务分配中考虑了威胁区的影响,同步规划出了满足曲率约束的可飞航路.因此,所设计的顺序扩展一致性包算法能够实现复杂的协同耦合多任务分布式分配.

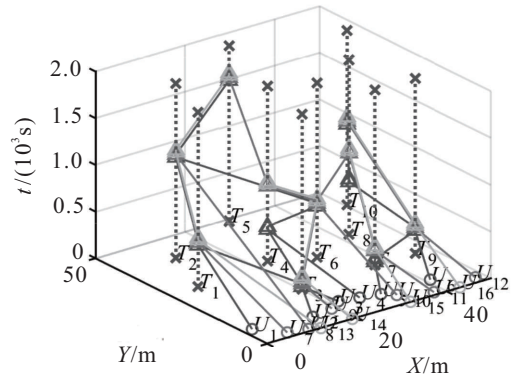


图4 任务分配结果时空示意图

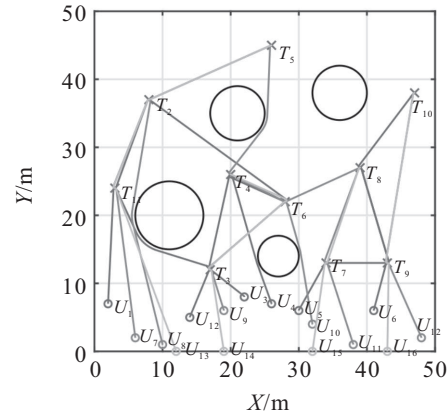


图5 任务分配结果二维示意图

3.2 算法性能对比

为了有效地评估 SECBA 算法的最优性和时效性,通过2个算例与其他算法进行算法性能的对比如图6和图7所示.3种算法的平均目标函数值分别为1084.8、1251.6、1403.4,

其中, CBBA-TCC算法的任务价值递减系数设为0.001, IGA算法的种群规模设为200,最大迭代次数设为600,交叉概率为0.8,变异概率为0.2,使用精英保留策略,保留的最优个体数量为2.采用式(9)中的最小化全局目标函数对任务分配方案进行评估,权重系数 α 和 β 分别设为0.55和0.45.

例1 设定8架无人机对1~12个目标执行任务,目标和无人机的位置均随机生成.各算法的目标函数值和求解耗时间对比如图6和图7所示.3种算法的平均目标函数值分别为1084.8、1251.6、1403.4,

SECBBA的平均目标函数值相比于CBBA-TCC和IGA分别提升了13.3%和22.7%。任务规模较小时,3种算法都有着较好的最优性。随着任务规模的增大,IGA算法虽然寻优能力更强,但易陷入局部最优,最优性不够稳定。而SECBBA算法利用顺序分层的策略,将上层任务分配积累的时间优势传递到下层,从而缩短整体的任务执行时间,因此SECBBA算法的整体最优性优于另外两种算法。由求解耗时的对比图可以看出,随着任务规模的增大,IGA算法的求解耗时逐渐增大,而SECBBA算法和CBBA-TCC算法的求解耗时几乎不变,并且有着明显的优势。因此,本文所提出的算法能够适应不同规模的任务分配,并且保持良好的最优性和时效性。

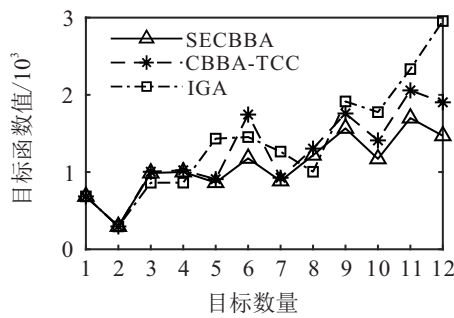


图6 算法目标函数值对比

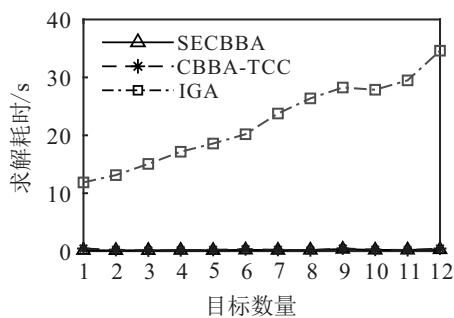


图7 算法求解耗时对比

例2 为了对算法的最优性和时效性作进一步评估,固定目标数量和无人机数量,设定8架无人机对10个目标执行任务,进行50次Monte Carlo仿真实验,并基于Monte Carlo仿真实验结果对算法性能进行评估^[20],评分机制为

$$Q = \sum_{i=1}^{N_m} \left(a \frac{F_{SECBBA}^{ave}}{F_i} + b \frac{t_{SECBBA}^{ave}}{t_i} \right), a + b = 1. \quad (20)$$

其中: F_{SECBBA}^{ave} 和 t_{SECBBA}^{ave} 分别为SECBBA算法的平均目标函数值和平均求解耗时, N_m 为Monte Carlo仿真次数, F_i 和 t_i 分别为各算法第 i 次仿真的目标函数值和求解耗时,分别设置权重系数 a 为0.7~0.3,间隔为0.1,对算法的最优性和时效性进行评估。3种算法的目标函数值和求解耗时如图8和图9所示,整体性

能评估得分如图10所示。SECBBA算法的平均目标函数值为1424.7,相比于其他两种算法分别提升了9.8%和23.2%,在相同任务规模的情况下,SECBBA算法求解的最优性优于其他两种算法,鲁棒性更强。SECBBA算法作为一种分布式任务分配方法,其求解的时效性相比于集中式任务分配方法提升明显,更加适合动态场景下高实时性要求的任务分配。图10中SECBBA算法在5种权重系数下的整体性能均优于其他两种算法,因此,SECBBA算法能够在保证求解耗时的前提下,得到目标函数值更优的任务分配结果。

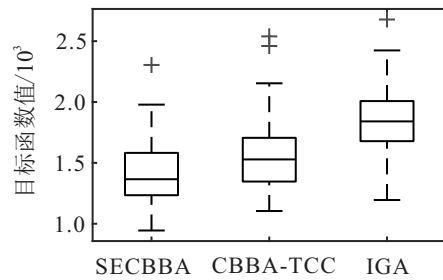


图8 算法目标函数值对比

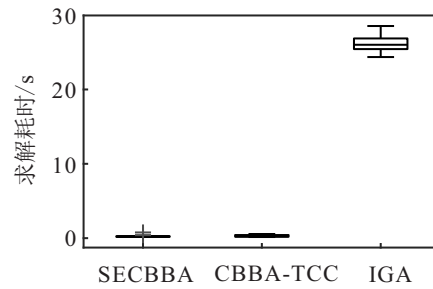


图9 算法求解耗时对比

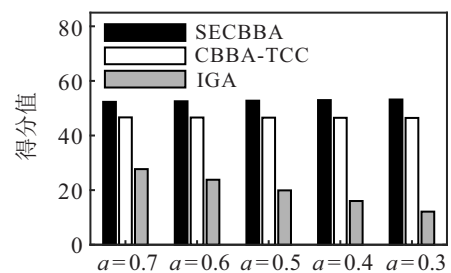


图10 算法整体性能对比

4 结论

本文针对异构无人机协同执行复杂的时序耦合多任务场景中,具有任务载荷需求约束、任务时序约束等多约束条件的分布式任务分配问题,设计了一种无死锁的顺序扩展一致性包算法进行求解。首先对一致性包算法进行了扩展,然后针对可能存在的死锁问题,设计了一种基于有向图深度优先搜索的死锁检测和修正方法,实现了多无人机执行型任务的无冲突分配。利用顺序分层的策略将关联任务之间的时序

约束转化为各自的时间窗约束,对时序多任务分配问题进行了求解.将航路规划耦合到任务分配中,得到了更为可靠的任务分配结果.最后通过仿真实验分析并与其他方法进行对比,验证了本文所提出的SECBBA算法具有较好的最优性和时效性,能够适应复杂动态场景下的任务分配要求.由于本文提出的SECBBA算法具有较好的时效性,后续的研究工作主要考虑在求解时间允许的范围内进一步提高算法的最优性.

参考文献(References)

- [1] 季自力, 张申, 王文华. 联合集群作战力量将成为未来战场的主角[J]. 军事文摘, 2021(15): 58-63.
(Ji Z L, Zhang S, Wang W H. The joint cluster combat force will become the protagonist of the future battlefield[J]. Military Digest, 2021(15): 58-63.)
- [2] Coutinho W P, Battarra M, Fliege J. The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review[J]. Computers & Industrial Engineering, 2018, 120: 116-128.
- [3] Kitjacharoenchai P, Ventresca M, Moshref-Javadi M, et al. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach[J]. Computers & Industrial Engineering, 2019, 129: 14-30.
- [4] Flores-Luyo L, Agra A, Figueiredo R, et al. Heuristics for a vehicle routing problem with information collection in wireless networks[J]. Journal of Heuristics, 2020, 26(2): 187-217.
- [5] Faria A F, de Souza S R, de Sá E M. A mixed-integer linear programming model to solve the multidimensional multi-way number partitioning problem[J]. Computers & Operations Research, 2021, 127: 105133.
- [6] Shima T, Rasmussen S J, Sparks A G, et al. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms[J]. Computers & Operations Research, 2006, 33(11): 3252-3269.
- [7] Zhao W Q, Meng Q G, Chung P W H. A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario[J]. IEEE Transactions on Cybernetics, 2016, 46(4): 902-915.
- [8] Jia Z Y, Yu J Q, Ai X L, et al. Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm[J]. Aerospace Science and Technology, 2018, 76: 112-125.
- [9] 李夏苗, 廖文昆, 伍国华, 等. 基于两阶段迭代优化的空天观测资源协同任务规划方法[J]. 控制与决策, 2021, 36(5): 1147-1156.
(Li X M, Liao W K, Wu G H, et al. A two-stage iterative optimization method for the coordinated task planning of space and air observation resources[J]. Control and Decision, 2021, 36(5): 1147-1156.)
- [10] Chen P, Yan F, Liu Z, et al. Communication-constrained task allocation of heterogeneous UAVs[J]. Acta Aeronautica et Astronautica Sinica, 2021, 42(8): 313-326.
- [11] Choi H L, Brunet L, How J P. Consensus-based decentralized auctions for robust task allocation[J]. IEEE Transactions on Robotics, 2009, 25(4): 912-926.
- [12] Gallud X, Selva D. Agent-based simulation framework and consensus algorithm for observing systems with adaptive modularity[J]. Systems Engineering, 2018, 21(5): 432-454.
- [13] Nayak S, Yeotikar S, Carrillo E, et al. Experimental comparison of decentralized task allocation algorithms under imperfect communication[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 572-579.
- [14] Ye F, Chen J, Sun Q, et al. Decentralized task allocation for heterogeneous multi-UAV system with task coupling constraints[J]. The Journal of Supercomputing, 2021, 77(1): 111-132.
- [15] Li X M, Tang J Y, Dai J J, et al. Dynamic coalition task allocation of heterogeneous multiple agents[J]. Journal of Northwestern Polytechnical University, 2020, 38(5): 1094-1104.
- [16] Chen J, Qing X G, Ye F, et al. Consensus-based bundle algorithm with local replanning for heterogeneous multi-UAV system in the time-sensitive and dynamic environment[J]. The Journal of Supercomputing, 2022, 78(2): 1712-1740.
- [17] Tang J Y, Li X M, Dai J J, et al. Coalition task allocation of heterogeneous multiple agents with complex constraints[J]. Control Theory & Applications, 2020, 37(11): 2413-2422.
- [18] Yuan Y P, Shen X T, Pan W, et al. Constrained likelihood for reconstructing a directed acyclic Gaussian graph[J]. Biometrika, 2018, 106(1): 109-125.
- [19] Shanmugavel M, Tsourdos A, White B, et al. Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs[J]. Control Engineering Practice, 2010, 18(9): 1084-1092.
- [20] Wu W N, Cui N G, Guo J F. Distributed task assignment method based on local information consensus and target estimation[J]. Control Theory & Applications, 2018, 35(4): 566-576.

作者简介

高程(1997—), 男, 硕士生, 从事集群飞行器任务规划的研究, E-mail: gaocheng@nuaa.edu.cn;

都延丽(1977—), 女, 副教授, 博士, 从事集群飞行器任务规划与控制等研究, E-mail: duyanli@nuaa.edu.cn;

步雨浓(1989—), 女, 工程师, 硕士, 从事集群飞行器任务规划的研究, E-mail: htkg_buyunong@126.com;

刘燕斌(1980—), 男, 副教授, 博士, 从事飞行器任务规划和控制等研究, E-mail: liuyb@nuaa.edu.cn;

杨小草(1997—), 女, 硕士生, 从事集群飞行器航路规划的研究, E-mail: yangxiaocao@nuaa.edu.cn.