

复杂约束条件下异构多智能体联盟任务分配

唐嘉钰^{1†}, 李相民¹, 代进进¹, 薄 宁²

(1. 海军航空大学 岸防兵学院, 山东 烟台 264001; 2. 中国人民解放军 91213 部队, 山东 烟台 264001)

摘要: 分布式任务决策是提高多智能体系统自主性的关键. 以异构多智能体协同执行复杂任务为背景, 首先建立了一种考虑任务载荷资源约束、任务耦合关系约束及执行窗口约束等条件的异构多智能体分布式联盟任务分配模型; 其次, 对一致性包算法(CBBA)进行了扩展, 提出了基于改进冲突消解原则的一致性联盟算法(CBCA), 以实现异构多智能体协同无冲突任务分配, 并进一步证明了在一定条件下CBCA算法收敛于改进顺序贪婪算法(ISGA). 最后通过数值仿真, 验证了CBCA算法求解复杂约束条件下异构多智能体联盟任务分配问题的可行性和快速性.

关键词: 多智能体系统; 分布式决策; 一致性包算法; 联盟生成; 无人机

引用格式: 唐嘉钰, 李相民, 代进进, 等. 复杂约束条件下异构多智能体联盟任务分配. 控制理论与应用, 2020, 37(11): 2413–2422

DOI: 10.7641/CTA.2020.90868

Coalition task allocation of heterogeneous multiple agents with complex constraints

TANG Jia-yu^{1†}, LI Xiang-min¹, DAI Jin-jin¹, BO Ning²

(1. College of Coastal Defense, Naval Aviation University, Yantai Shandong 264001, China;

2. Unit 91213 of the PLA, Yantai Shandong 264001, China)

Abstract: Distributed decision making is the key of enhancing the autonomy of multi-agent systems. Firstly, based on the problem of multi agents cooperating to perform complicated tasks, a decentralized coalition task assignment model of heterogeneous multiple agents with consideration of specific equipment requirements of tasks, coupled constraints and the time window constraint is proposed. Secondly, the consensus-based bundle algorithm (CBBA) is extended to the consensus-based coalition algorithm (CBCA) with the improved principles of conflicts resolution to achieve the conflict-free solution of the task assignment problem. Further CBCA is proved to be convergent to the improved sequential greedy algorithm (ISGA). Finally, simulation results of multiple unmanned search and combat aerial vehicles performing search and attack missions demonstrate the feasibility and rapidity of the proposed algorithm.

Key words: multi agent systems; distributed decision making; consensus-based bundle algorithm(CBBA); coalition formation; unmanned aerial vehicles

Citation: TANG Jiayu, LI Xiangmin, DAI Jinjin, et al. Coalition task allocation of heterogeneous multiple agents with complex constraints. *Control Theory & Applications*, 2020, 37(11): 2413–2422

1 引言

多智能体协同任务分配是实现多智能体协同控制的基础, 其求解方法分为集中式和分布式^[1–2].

依据文献[3]提出的分类法, 多智能体协同任务分配问题等同于交叉调度的单任务多机器人时间扩展任务分配问题(cross-schedule dependent single task multi-robot time-extended task allocation, XDSTMRTA), 其中交叉调度是指智能体是否执行某个任务受其他智能体任务执行情况影响; 单任务是指

智能体一次只能执行一个任务; 时间扩展则是指智能体在执行任务之前预先进行任务规划. 对XDSTMRTA问题的研究中, 文献[2]采用集中式混合整数线性规划算法(mixed-integer linear programming, MILP)解决了搜救任务中小组任务分配问题. 文献[4]在MILP算法基础上进一步考虑了任务之间的时序关系、起始时间的同步需求及智能体等待惩罚等约束条件, 对单任务多机器人时间扩展任务分配问题进行了研究. 在集中式框架中, 中心节点计算获得最优解, 并

收稿日期: 2019–10–17; 录用日期: 2020–06–29.

[†]通信作者. E-mail: 342118507@qq.com; Tel.: +86 18605456315.

本文责任编辑: 高会军.

国家自然科学基金项目(61305136)资助.

Supported by the National Natural Science Foundation of China (61305136).

通过有效通信方式将最优解发送给智能体. 与分布式框架相比, 集中式分配框架存在中央节点负荷大、系统鲁棒性差等缺点^[5]. 在对抗环境中, 敌方可能会将目标锁定在中心节点, 分布式算法可以为节点或通信基础设施失效引起的单点故障提供稳健性.

基于市场的分布式求解方法已广泛应用于多机器人协同任务分配问题. 分布式求解框架中, 为避免各智能体自主分配结果出现冲突, 多智能体系统需要在态势感知和(或)任务规划上达成一致^[6]. 文献[7-9]将分布式竞标与集中式拍卖相结合提出了基于拍卖的多无人机任务分配方法, 通过一定的一致性协商机制对多无人机的竞标进行冲突消解, 从而确定获胜无人机.

为了削弱拍卖者中心节点作用, 提高分布式多智能体系统鲁棒性, Choi H L等人提出了基于一致性的拍卖算法(consensus based auction algorithm, CBAA)解决分布式智能体系统中单任务分配问题^[10], 试验表明该算法减轻了数据通信负担. CBAA算法解决了每个任务需要一个智能体完成的单一任务分配问题. 实际上, 一个智能体可以执行位置相近的多个任务. 因此, Choi H L等人将CBAA算法扩展为一致性包算法(consensus based bundle algorithm, CBBA). CBBA算法中, 各智能体将任务序列排列组合, 并选择效益最大的任务组合作为任务包(bundle). 随着相邻智能体不断构建更有价值的任务包, 各智能体不断删减或修正自身任务包. CBBA算法采取与CBAA算法相同的协商机制以获得无冲突任务分配方案^[10].

文献[11]结合具体无人机应用场景, 将无人机动力学约束及任务间独立性约束与CBBA算法相结合, 提出了耦合约束条件下一致性包算法(coupled-constraint CBBA). 文献[12]证明了CBBA算法可收敛到纳什均衡(Nash equilibrium)但非帕累托最优(Pareto optimal)解, 同时指出CBBA算法可为单机器人单任务分配问题^[2](single-robot single task task allocation, SRSTTA)提供次优解决方案. 当CBBA算法非负评分机制满足边际增益递减(diminishing marginal gain, DMG)的收敛特性时, CBBA算法相对于最优目标值可达到至少50%的最优性^[13]. 同时, 由于CBBA中投标算法运行时间为多项式时间, 当智能体和任务数量增多时, CBBA算法的可扩展性保证了其在实时动态环境下的适用性^[13].

综合国内外研究现状可知, 目前分布式多智能体任务分配研究主要有以下不足:

1) 目前研究方法主要针对单任务单智能体场景, 但在实际中单个复杂任务需要多智能体合作完成, 例如多无人机协同搜索攻击等作战应用中, 对单个目标

的打击任务需要侦察型和攻击型无人机协同完成;

2) 现有分布式多智能体协同任务分配研究中, 各任务之间相互独立, 未考虑任务之间耦合约束关系.

基于国内外研究现状, 本文针对任务载荷资源约束、任务耦合关系约束及执行窗口约束等条件下的异构多智能体联盟任务分配问题, 对分布式一致性包理论进行扩展, 提出了一致性联盟算法. 首先, 各智能体综合考虑任务时变收益、行程代价损耗, 依据边际效益最大的原则选择有效任务构造自身任务包; 其次, 基于改进协调原则实现异构多智能体联盟任务分配决策的冲突调解; 最后通过仿真试验验证了该方法的可行性.

2 多智能体联盟任务分配问题描述

联盟任务分配问题的目标是将包含 N_T 个任务的集合 $J \triangleq \{1, \dots, N_T\}$ 分配给 N_a 个智能体的集合 $I \triangleq \{1, \dots, N_a\}$, 智能体 $i \in I$ 最多可执行 L_i 个任务. 任务 $j \in J$ 由四元素组 $\langle \text{pos}_j, \text{tae}_j, \text{num}_j, \text{tc}_j \rangle$ 表示, 其中: pos_j 为任务坐标; tae_j 为执行该任务所需载荷资源; num_j 为完成任务 j 所需智能体数量; $\text{tc}_j = \{\tau_j^{\text{start}}, \tau_j^{\text{end}}, \text{dur}_j, \lambda_j\}$ 为任务时间约束集合, 其中: τ_j^{start} 和 τ_j^{end} 分别为任务 j 执行最早和最晚时间节点, dur_j 为任务持续时间, $0 \leq \lambda_j < 1$ 为任务时间折扣因子.

多智能体联盟分布式任务分配问题数学模型^[14-15]为

$$\begin{aligned} & \max \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_T} r_{ij} x_{ij} \right), \\ & \text{s.t.} \quad \sum_{j=1}^{N_T} x_{ij} \leq L_i, \\ & \quad \sum_{i=1}^{N_a} x_{ij} \leq \text{num}_j, \\ & \quad x_{ij} \in \{1, 0\}, \forall (i, j) \in I \times J, \end{aligned} \quad (1)$$

式中: r_{ij} 为智能体 i 执行任务 j 的收益; $x_{ij} \in \{1, 0\}$ 为任务分配决策变量, $x_{ij} = 1$ 表明智能体 i 执行任务 j , 否则 $x_{ij} = 0$.

异构多智能体联盟任务分配包括以下约束条件:

1) 任务需要智能体携带特定的载荷资源完成. 设联盟分配问题中有 N_q 类载荷资源 $E \triangleq \{e_1, \dots, e_{N_q}\}$, 其中 age_i 为智能体 i 所携带的资源列表, 且 $\text{age}_i \subseteq E$. 类似地, 设执行任务 j 所需资源列表为 tae_j , $\text{tae}_j \subseteq E$. $\text{age}_i \cap \text{tae}_j \neq \emptyset$ 表示 i 可参与执行任务 j ; $\text{tae}_j = \emptyset$ 则表示任务 j 对智能体携带资源无要求, 所有智能体都可投标.

当

$$\text{tae}_j \setminus \{\text{age}_i | x_{ij} = 1\} = \emptyset, \forall i \in I \quad (2)$$

成立时, 可认为任务分配结果满足任务 j 的载荷资源需求, 其中 $\text{tae}_j \setminus \{\text{age}_i | x_{ij} = 1\}$ 表示 tae_j 中去掉已分配智能体 $\{\text{age}_i | x_{ij} = 1\}$ 所携带载荷资源后的剩余载荷资源.

2) 复杂任务 j 可分为 Sub_j 个子任务, 分别需要不同类型载荷资源, 每个子任务只需一种载荷资源. 子任务分配之间存在着耦合约束关系, 为描述耦合约束关系, 定义子任务关系矩阵为方阵 $D = \{d_{wq}\}$, $w = 1, \dots, \text{Sub}_j$; $q = 1, \dots, \text{Sub}_j$, 其中: d_{wq} 表示子任务 w 和子任务 q 之间的耦合关系约束, $d_{wq} = 1$ 表示子任务 w 和子任务 q 之间存在时序关系约束, $d_{wq} = 0$ 表示两个子任务之间无时序约束关系. 当子任务之间存在时序关系约束时, w 和 q 的起始时间存在最小时间间隔 $\Delta t_{\min}$, 即

$$T(w, q) = t_{wq} = \begin{cases} \Delta t_{\min}, & d_{wq} = 1, \\ \infty, & d_{wq} = 0, \end{cases} \quad (3)$$

其中 T 为子任务时序间隔矩阵.

以多无人机智能体协同执行侦察攻击型任务为例, 子任务主要可分为侦察子任务和攻击子任务, 对应子任务关系矩阵为

$$D = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

对应的时序间隔矩阵为

$$T = \begin{bmatrix} 0 & \Delta t_{\min} \\ \Delta t_{\min} & 0 \end{bmatrix}.$$

矩阵 D 和 T 的行(列)元素按顺序分别表示侦察子任务和攻击子任务, $\Delta t_{\min}$ 为执行攻击子任务前完成侦察子任务所需的最短时间.

3 一致性联盟算法

CBBA算法主要考虑单个任务由一个智能体执行的应用场景, 在需要多智能体协同执行复杂任务的场景下具有局限性. 本节在CBBA算法基础上提出了一致性联盟算法 (consensus based coalition algorithm, CBCA), 旨在解决异构多智能体协同执行复杂任务时联盟分组问题. CBCA算法分为两阶段: 智能体任务包构建和基于一致性的冲突消解.

3.1 算法关键要素

在CBCA算法中, 首先明确定义智能体 i 任务分配信息的关键要素:

1) 执行路径列表 p_i .

执行路径列表 $p_i \triangleq \{p_{i1}, \dots, p_{i|p_i|}\}$ 中存放分配给智能体 i 的待执行任务序号, 列表中元素按照 i 计划执行顺序排列, $|p_i|$ 表示列表长度. 空路径列表为

$$p_i = \emptyset.$$

2) 任务包 b_i .

任务包 $b_i \triangleq \{b_{i1}, \dots, b_{i|b_i|}\}$ 表示分配给智能体 i 的任务集合, 集合中元素按加入任务包的先后顺序排列, $|b_i|$ 代表任务包长度, 且有 $|b_i| \leq |p_i|$, $b_i = \emptyset$ 表示空任务包.

3) 获胜智能体矩阵 Z^i .

Z^i 为 $N_a \times N_T$ 维矩阵, 其元素 $z_{kj}^i = 1$ 表示智能体 i 认为智能体 k 是任务 j 的投标获胜者; 否则 $z_{kj}^i = 0$. Z^i 中第 j 列非零元素数量之和 $\text{Sum}_j = \sum_{k=1}^{N_a} z_{kj}^i$ 代表了智能体 i 认为执行任务 j 的智能体总数.

4) 获胜投标值矩阵 Y^i .

Y^i 存储智能体 i 视角下各项任务联盟成员的投标值, 矩阵中各元素与 Z^i 一一对应. 当智能体 i 认为任务 j 没有获胜者时, $y_{ij} = 0$.

5) 时间戳列表 s_i .

时间戳列表 $s_i \triangleq \{s_{i1}, \dots, s_{iN_a}\}$ 记录智能体 i 从相邻其他智能体中获得更新信息的时刻, 其中 s_{ik} 表示智能体 i 从智能体 k 获得最新信息的时刻. 该向量是冲突消除阶段的重要指标, 表征智能体从其他智能体获得信息的新旧程度.

3.2 任务包构建

在任务包构建过程中, 各智能体迭代选择任务添加到自身任务包中. 每个智能体以最大化自身任务收益增量为选择原则. 设智能体 i 初始任务包为 b_i , i 选择收益边际增益最大的任务 $j^* \in J$ 加入到 b_i 中. 任务选择过程持续到任务包已满或无可选任务. 但智能体只对满足时序约束和载荷资源约束的有效任务进行投标.

智能体 i 任务分配总收益为

$$S_i^{p_i} = \sum_j (r_{ij} - \text{cost}_j), \quad (4)$$

式中 $r_{ij}(p_i)$ 为任务执行收益. 考虑任务动态时效性, 采用时间折扣收益来计算, 即

$$r_{ij}(p_i) = \begin{cases} \bar{r}_j e^{-\lambda_j(\tau_{ij}(p_i) - \tau_j^{\text{start}})}, & \tau_{ij}(p_i) \in [\tau_j^{\text{start}}, \tau_j^{\text{end}}] \\ & \text{且 } \text{age}_i \cap \text{tae}_j \neq \emptyset, \\ 0, & \text{其他,} \end{cases} \quad (5)$$

式中: $\bar{r}_j$ 为任务 j 的静态收益; $0 < \lambda_j < 1$ 为任务 j 的时间折扣因子; τ_j^{start} 为智能体 i 沿路径 p_i 执行任务 j 的估计时间; cost_j 为智能体行程损耗代价, 且有

$$\text{cost}_{ij} = d_{ij}\gamma, \quad (6)$$

式中: d_{ij} 为智能体 i 与任务 j 之间的直线距离, γ 为行程代价系数.

依据新任务 j' 加入 b_i 前后的收益, 可获得 j' 的边际增益为

$$c_{ij'}[b_i] = \begin{cases} 0, & j' \in b_i, \\ \max_{n \leq |p_i|+1} S_i^{p_i \oplus n\{j'\}} - S_i^{p_i}, & \text{其他}, \end{cases} \quad (7)$$

式中: n 表示 j' 在 p_i 中所有可能位置; $|p_i|$ 为列表长度; $p_i \oplus n\{j'\}$ 表示将 j' 插入到路径列表 p_i 中第 n 位, 原路径列表中第 n 位及其以后的元素保持原顺序不变, 依次后移.

在多智能体联盟任务分配中, 考虑到单项复杂任务需要多智能体协同完成, 因此CBBA算法中单纯比较边际增益与当前任务获胜投标值的方法并不适用.

当 $\text{num}_j = 1$ 时, 任务 j 为单智能体执行型, 投标过程与CBBA算法相同. 当 $\text{num}_j > 1$ 时, 任务 j 需要多智能体组成联盟协同完成, 此时任务 j 可分为两种情况: 1) 竞标执行任务 j 的智能体数量已满足联盟智能体数量需求, 即 $\text{Sum}_j = \text{num}_j$, 此时任务 j 分配满荷; 2) 分配执行任务 j 的智能体数量未满足数量需求, 即 $\text{Sum}_j < \text{num}_j$.

任务 j 分配已满荷时, 智能体将自身边际增益与当前任务最小获胜投标值作比较, 当

$$c_{ij} > \min(y_{kj}^i), \forall k \in I \quad (8)$$

时, 智能体 i 可以投标 ($h_{ij} = 1$), 并将最小获胜投标值替换为自身投标值; 否则, 智能体 i 放弃任务 j . 当智能体 i 投标值与最小获胜投标值相同时, 选择身份标识小的智能体.

智能体不断构造自身任务包直至任务包已满或没有满足约束条件的有效任务. 具体CBCA任务包构造算法如算法1所示.

算法1 CBCA算法任务包构造阶段.

输入: $Z^i(t-1)$, $Y^i(t-1)$, $b_i(t-1)$ 和 $p_i(t-1)$

输出: $Z^i(t)$, $Y^i(t)$, $b_i(t)$ 和 $p_i(t)$

While $|b_i(t-1)| < L_i$ do
 $c_{ij} = \max_{n \leq |p_i|} S_i^{p_i \oplus n\{j\}} - S_i^{p_i}, \forall j \in J$

for $\forall j \in J$

if $\text{num}_j > 1$ then

if $\text{num}_j > \sum_{k=1}^{N_a} z_{kj}^i$ then

$h_{ij} = 1$

else if $c_{ij} > \min(y_{kj}^i), \forall k \in I$ then

$h_{ij} = 1$

else

$h_{ij} = 0$

end

else

$h_{ij} = I(c_{ij} > \sum_{k=1}^{N_a} y_{kj}^i)$

end

end

$J_i = \arg \max_j c_{ij} h_{ij}$

$n_{i,J_i} = \arg \max_n S_i^{p_i \oplus n\{J_i\}}$

$b_i(t) = b_i(t-1) \oplus_{\text{end}} \{J_i\}$

$p_i(t) = p_i(t-1) \oplus_{n_{i,J_i}} \{J_i\}$

$y_{i,J_i}^i(t) = c_{i,J_i}$

$z_{i,J_i}^i(t) = 1$

end while

3.3 基于一致性的冲突消解

任务包构建过程结束后, 为获得无冲突任务分配结果, 智能体需要与相邻智能体共享获胜智能体矩阵、获胜投标矩阵和时间戳列表等信息, 并按照一定的行动规则更新相应信息以获得无冲突任务分配. 本文假设在一致性协商阶段, 各智能体之间采用同步通信机制, 关于异步通信机制下多智能体任务分配研究详见文献[16].

在冲突消解阶段, 智能体利用时间戳列表 s_i 来确定接收到的最新信息. 每当智能体 i 接收到来自相邻智能体 k 的信息时, 对该信息标记如下时间戳:

$$s_{ik} = \begin{cases} \tau_r, & g_{ik} = 1, \\ \max_{l: g_{il}=1} s_{lk}, & \text{其他}, \end{cases} \quad (9)$$

式中: $g_{ik} = 1$ 表示智能体 i 与智能体 k 之间存在通信链接, 否则 $g_{ik} = 0$, 特别地, $g_{ii} = 1$; τ_r 为智能体 i 接受到信息的时刻. 具体冲突消解过程如算法2所示.

算法2 CBCA算法冲突消解阶段.

输入: $Z^k(t-1)$, $Y^k(t-1)$ 和 $s_k(t-1)$

输出: $Z^i(t)$, $Y^i(t)$ 和 $s_i(t)$

智能体 i 将 $Z^i(t-1)$, $Y^i(t-1)$ 和 $s_i(t-1)$ 发送给智能体 k .

智能体 i 接收智能体 k 发送的 $Z^k(t-1)$, $Y^k(t-1)$ 和 $s_k(t-1)$.

for $\forall j \in J$

if $\text{num}_j = 1$ then

进行CBBA算法冲突消解

else

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

if $k = m$ and $s_{km} > s_{im}$ then

$z_{mj}^i = z_{mj}^k$

$y_{mj}^i = y_{mj}^k$

```

end
end
for  $\forall m \in I$  where  $z_{mj}^k = 1$ 
  if  $m \neq i$  and  $z_{mj}^i = 0$  and  $s_{km} > s_{im}$  then
    if  $\sum_{m=1}^{N_a} z_{mj}^i < \text{num}_j$  then
       $z_{mj}^i = z_{mj}^k$ 
       $y_{mj}^i = y_{mj}^k$ 
    else
      if  $\min(y_{nj}^i \forall n) < y_{mj}^k$  then
         $z_{nj}^i = 0, z_{mj}^i = z_{mj}^k$ 
         $y_{nj}^i = 0, y_{mj}^i = y_{mj}^k$ 
      end
    end
  end
end
end
end
end
end
 $s_{ik} = t$ 
end

```

当 $\text{num}_j = 1$ 时, 智能体采用CBBA第2阶段的冲突消解方式(算法2中第4–6行). $\text{num}_j > 1$ 时, CBCA冲突消解(算法2中第7–26行)可分为两部分: 首先接收者 i 将自身获胜投标信息与发送者 k 的获胜投标信息进行比较(算法2中第7–11行), $s_{km} > s_{im}$ 表明发送者 k 信息更新, 通过第1阶段与发送者 k 信息对比后, 接收者 i 可确认自身存储信息是当前最新的; 随后依据对比结果, 智能体 i 对自身信息进行调整(算法2中第12–26行), 当发送者 k 认为智能体 $m(m \neq i)$ 执行任务 j , 且接收者 i 认为 m 不执行任务 j 时(算法2中13–14行), 接收者 i 需要判断是否有足够智能体执行任务 j (算法2中第15行), 若接收者 i 认为分配给任务 j 的智能体数量未满足任务需求, 则进行更新(算法2中第16–17行). 由于在任务包构造过程中智能体只对符合约束条件的任务进行投标, 因此当 i 将联盟内成员替换时, 新成员与旧成员之间均符合任务载荷资源需求. $\sum_{m=1}^{N_a} z_{mj}^i < \text{num}_j$ 表明任务 j 有多余分配空间, 并且当 $\text{age}_m \in \text{tae}_j \setminus \{\text{age}_i | z_{ij}^i = 1\}$ 时, 任务 j 需要携带载荷资源 age_m 的智能体 m ; 若任务 j 分配已满荷, 则需要判断现有联盟内成员的投标值是否小于 m 的投标值(算法2中第19行), 且二者有重叠载荷资源类型, 即

$$\min(y_{nj}^i \forall n) < y_{mj}^k, \text{age}_m \cap \text{age}_n \neq \emptyset$$

成立时, 将当前任务联盟内投标值较低的智能体替换为智能体 m . 为避免造成死锁, 当投标值相同时, 选择标识较小的智能体.

3.4 算法收敛性

文献[10]中已证明CBBA分配策略收敛至集中式顺序贪婪算法(sequential greedy algorithm, SGA). 为满足单个任务多智能体执行的任务需求, 本文提出了改进顺序贪婪算法(improved sequential greedy algorithm, ISGA), 如算法3所示, 其中

$$N_{\min} = \sum_{i=1}^{N_a} \sum_{j=1}^{N_T} z_{ij}^i.$$

算法3 改进顺序贪婪算法.

输入: 智能体集 I , 任务集 J .

输出: 智能体 $i \in I$ 的任务包 b_i .

$c_{ij}^{(1)} = c_{ij}[\{\emptyset\}], \forall (i, j) \in I \times J$

$\eta_i = 1, \forall i \in I$

for $n = 1$ to $N_{\min}$ do

if $\text{num}_j = 1$ then

同SGA算法

else

for $m = 1$ to num_j do

$$(i_{n,m}^*, j_n^*) = \arg \max_{(i,j) \in I \times J} c_{ij}^{(n)}$$

$$\eta_{i_{n,m}^*} = \eta_{i_{n,m}^*} + 1$$

$$b_{i_{n,m}^*}^{(n)} = b_{i_{n,m}^*}^{(n-1)} \oplus \{j_n^*\}$$

$$b_i^{(n)} = b_i^{(n-1)}, \forall i \neq i_{n,m}^*$$

if $\eta_{i_{n,m}^*} = L_{i_{n,m}^*}$ then

$$I_{n+1} = I_n \setminus \{i_{n,m}^*\}$$

$$c_{i_{n,m}^*, j}^{(n+1)} = 0, \forall j \in J$$

else

$$I_{n+1} = I_n$$

end

end

$$J_{n+1} = J_n \setminus \{j_n^*\}$$

$$c_{i, j_n^*}^{(n+1)} = 0, \forall i \in I_{n+1}$$

$$c_{i, j}^{(n+1)} = c_{ij}[b_i^{(n)}], \forall (i, j) \in I_{n+1} \times J_{n+1}$$

end

end

由参考文献[10]可知, 采用分布式一致协商算法完成网络直径为 D 的分布式多智能体系统任务分配, 且每个智能体采用DMG评分机制, 当冲突消解阶段迭代次数为 t 时, 有

$$z_{i^*, j_k^*}^i(t) = 1, \forall i \in I, i^* \in I_k^*, \forall k \leq n, \quad (10a)$$

$$y_{i^*, j_k^*}^i(t) = c_{i^*, j_k^*}^k, \forall i \in I, i^* \in I_k^*, \forall k \leq n, \quad (10b)$$

其中: (I_k^*, j_k^*) 为ISGA算法在第 k 次迭代得到的解; $c_{i^*, j_k^*}^k(i^* \in I_k^*)$ 为对应分值. 则下列5点结论成立:

1) CBCA算法中智能体 i 任务包前 $|L_i(n)| \triangleq$

$|b_i(n)|$ 项元素与ISGA算法中第 n 次迭代后任务包元素相同,即

$$b_i^{1:L_i(n)} = b_i^{(n)}; \quad (11)$$

2) 智能体 $i^* \in I_{n+1}^*$ 执行任务 j_{n+1}^* 的收益为

$$y_{i^*,j_{n+1}^*}^{i^*}(t) = c_{i^*,j_{n+1}^*}^{(n+1)}, \quad (12)$$

且有 $y_{i^*,j_{n+1}^*}^{i^*}(t) \geq y_{ij}^i(t), \forall (i,j) \in I_{n+1} \times J_{n+1}$;

3) $\forall s \geq t, \forall k \leq n$, 有 $z_{i^*,j_k^*}^{i^*}(s) = z_{i^*,j_k^*}^{i^*}(t), y_{i^*,j_k^*}^{i^*}(s) = y_{i^*,j_k^*}^{i^*}(t)$;

4) 智能体 $i^* \in I_{n+1}^*$ 执行任务 j_{n+1}^* 的收益在后续迭代过程中不会改变,并且不会有其他智能体出价更高,即

$$y_{i^*,j_{n+1}^*}^{i^*}(s) = y_{i^*,j_{n+1}^*}^{i^*}(t) \geq y_{i,j_{n+1}^*}^i(s), \forall s \geq t, \forall i \in I;$$

5) t 时刻经过 D 轮通信后,多智能体系统一致认为由智能体 $i \in I_{n+1}^*$ 协同执行任务 j_{n+1}^* ,即

$$y_{i^*,j_{n+1}^*}^{i^*}(t+D) = y_{i^*,j_{n+1}^*}^{i^*}(t), \quad (13a)$$

$$z_{i^*,j_{n+1}^*}^{i^*}(t+D) = z_{i^*,j_{n+1}^*}^{i^*}(t), \forall i. \quad (13b)$$

引理1 nD 轮通信后,CBCA算法收敛至 n 次迭代后ISGA算法,即

$$z_{i^*,j_k^*}^{i^*}(nD) = 1, \forall i \in I, i^* \in I_k^*, \forall k \leq n, \quad (14a)$$

$$y_{i^*,j_k^*}^{i^*}(nD) = c_{i^*,j_k^*}^k, \forall i \in I, i^* \in I_k^*, \forall k \leq n. \quad (14b)$$

证 采用递推法证明引理1.在ISGA算法中,由于 $\arg \max_{j \in J} c_{i^*,j}[\{\emptyset\}] = j_1^*, \forall i^* \in I_1^*$ 在CBCA算法第1次任务包构造过程中智能体 $i^* \in I_1^*$ 将任务 j_1^* 放入任务包中第1位.因为 $\forall (i,j) \in I \times J, \forall b$ 有

$$c_{i^*,j_1^*}[\{\emptyset\}] \geq c_{i,j}[b].$$

因此在后续迭代中不会有智能体对 j_1^* 出价更高.随后冲突消解阶段经过 D 轮通信后,所有智能体一致认为由智能体 $i \in I_1^*$ 协同执行任务 j_1^* .由结论3)和结论5),若

$$z_{i^*,j_k^*}^{i^*}(mD) = 1, \forall i \in I, i^* \in I_k^*, \forall k \leq m,$$

$$y_{i^*,j_k^*}^{i^*}(mD) = c_{i^*,j_k^*}^{(k)}, \forall i \in I, i^* \in I_k^*, \forall k \leq m,$$

则有

$$z_{i^*,j_k^*}^{i^*}(mD+D) = 1, \forall i \in I, i^* \in I_k^*, \forall k \leq m+1,$$

$$y_{i^*,j_k^*}^{i^*}(mD+D) = c_{i^*,j_k^*}^{(k)}, \forall i \in I, i^* \in I_k^*, \forall k \leq m+1.$$

因此, nD 轮通信后, $\forall k \leq n$,多智能体系统一致认为由智能体 $i \in I_k^*$ 协同执行任务 j_k^* . 证毕.

定理1 当静态网络半径为 D 的多智能体系统采用同步通信机制,且各智能体采用DMG评分机制时,CBCA算法收敛于ISGA算法,即

$$z_{i^*,j_k^*}^{i^*}(t) = 1, \forall i \in I, i^* \in I_k^*, \forall k \leq N_{\min}, \quad (15a)$$

$$y_{i^*,j_k^*}^{i^*}(t) = c_{i^*,j_k^*}^k, \forall i \in I, i^* \in I_k^*, \forall k \leq N_{\min}. \quad (15b)$$

证 依据结论5)和引理1, $\forall n \leq N_{\min}$,冲突消解阶段经过 nD 轮通信后,所有智能体都认同ISGA中前 n 项解.由结论3)可知,这些解在后续迭代中不会发生改变.因此经过 $N_{\min}D$ 轮迭代后,CBCA算法收敛于ISGA算法. 证毕.

4 仿真结果及分析

以多架侦察型无人机(unmanned search aerial vehicle, USAV)和攻击型无人机(unmanned combat aerial vehicle, UCAV)依次执行侦察与攻击任务为例,对文中提出的CBCA算法进行仿真验证,算例1对CBCA算法的可行性进行验证,算例2对CBCA算法和CBBA算法进行比较,算例3将分布式CBCA算法与分布式合同网协议算法进行比较.

设定多无人机之间为全连通通信网络.仿真实验均在Intel(R)Core(TM)i7-2630QM CPU 2.00 GHz 计算机上运行.

4.1 CBCA算法仿真结果分析

算例1中共有5架无人机对9个目标进行攻击,任务空间范围设定为 $5 \text{ km} \times 5 \text{ km} \times 2 \text{ km}$,仿真中无人机和目标初始状态信息如表1和表2所示.目标依据所需执行任务共分为3类:侦察类目标(需2架USAV)、攻击类目标(需2架UCAV)及侦察攻击类目标(需2架USAV和1架UCAV). $\Delta t_{\min} = 10 \text{ min}$.

表1 无人机参数设定

Table 1 Initial settings of UAVs

UAV	能力类型	可执行最大任务数 L_i	初始位置/km	速度/ $(\text{m} \cdot \text{s}^{-1})$	γ_i
1	USAV	5	(2.3, 3.4, 1.0)	30	0.5
2	UCAV	5	(3.7, 4.8, 1.2)	50	0.8
3	USAV	5	(4.4, 0.4, 1.0)	30	0.5
4	USAV	5	(1.2, 4.9, 1.5)	50	0.5
5	UCAV	5	(4.5, 2.7, 1.0)	50	0.8

各UAV分配的任务路径如表3所示,任务时序如图1所示.对察打型目标Task1,首先UAV1和UAV3进行侦察,侦察任务结束后UAV5执行攻击任务;对察打型目标Task5,同样首先UAV1和UAV3进行侦察,侦察任务结束后UAV5执行攻击任务;此外均有两架USAV执行纯侦察类任务;两架UCAV执行纯攻击类任务.此时算法收益值为1805.01.

表 2 任务空间参数设定

Table 2 Initial settings of tasks

任务Task	初始位置/km	任务类型	静态收益 $\bar{r}_j$	时间折扣因子 λ_j	任务耗时 dur_j/min	任务时间窗口/min
1	(2.5, 3.0, 0)	侦察、攻击	95	0.7	15	[0, 30]
2	(2.4, 1.6, 0)	攻击	100	0.8	10	[20, 70]
3	(3.2, 3.0, 0)	侦察	85	0.6	10	[10, 100]
4	(5.0, 3.3, 0)	攻击	100	0.8	5	[40, 90]
5	(3.5, 4.9, 0)	侦察、攻击	95	0.7	15	[30, 100]
6	(4.2, 4.8, 0)	侦察	85	0.6	10	[30, 100]
7	(4.1, 4.9, 0)	攻击	100	0.8	5	[60, 100]
8	(0.5, 1.0, 0)	侦察	85	0.6	10	[70, 100]
9	(2.9, 0.9, 0)	侦察	85	0.6	10	[80, 120]

表 3 联盟任务分配结果

Table 3 Assignment results of CBCA

UAV	任务路径 p_i	任务起始时间/min	任务收益
1	1, 3, 5, 8, 9	3.24, 20, 45.63, 73, 90	98.4472, 83.9587, 98.5652, 83.8367, 83.4331
2	2, 4, 7	33.4200, 67.4300, 80	97.9972, 98.1883, 98.9823
3	1, 3, 5, 6, 9	3.24, 20, 45.63, 61.2445, 90	99.2818, 83.7130, 98.4388, 37.9006, 82.9547
4	6, 8	60, 73	83.7221, 83.5556
5	1, 2, 5, 4, 7	13.24, 33.4200, 55.63, 67.4300, 80	98.0572, 98.2230, 98.9910, 98.6584, 98.1057

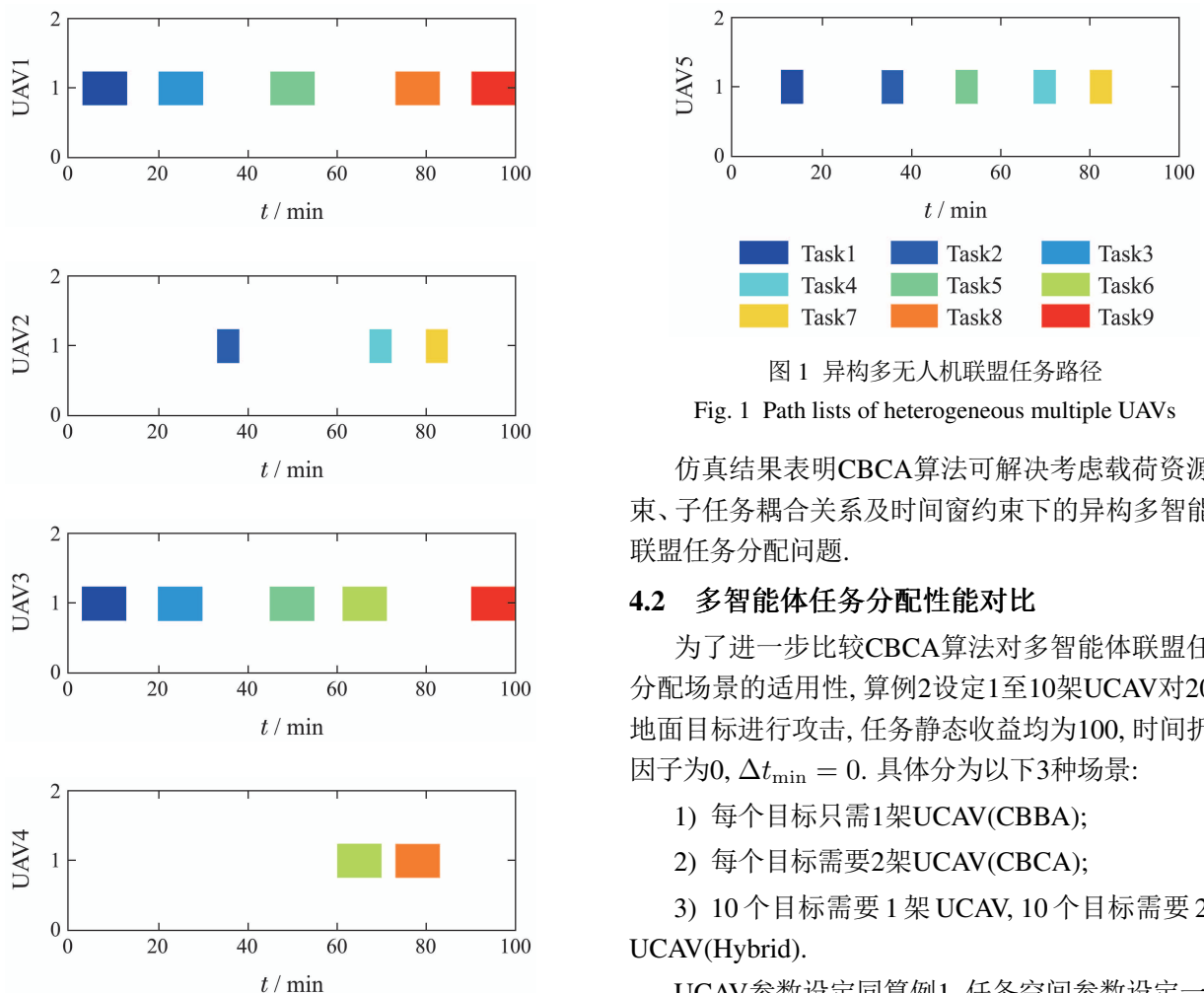


图 1 异构多无人机联盟任务路径

Fig. 1 Path lists of heterogeneous multiple UAVs

仿真结果表明CBCA算法可解决考虑载荷资源约束、子任务耦合关系及时间窗约束下的异构多智能体联盟任务分配问题。

4.2 多智能体任务分配性能对比

为了进一步比较CBCA算法对多智能体联盟任务分配场景的适用性, 算例2设定1至10架UCAV对20个地面目标进行攻击, 任务静态收益均为100, 时间折扣因子为0, $\Delta t_{\min} = 0$. 具体分为以下3种场景:

- 1) 每个目标只需1架UCAV(CBBA);
- 2) 每个目标需要2架UCAV(CBCA);
- 3) 10个目标需要1架UCAV, 10个目标需要2架UCAV(Hybrid).

UCAV参数设定同算例1, 任务空间参数设定一致,

分别记录了3种场景中算法总收益值、一致协商通信次数(所有UAV完成双向通信记为一次)和算法运行时间,对比结果如图2-4所示。

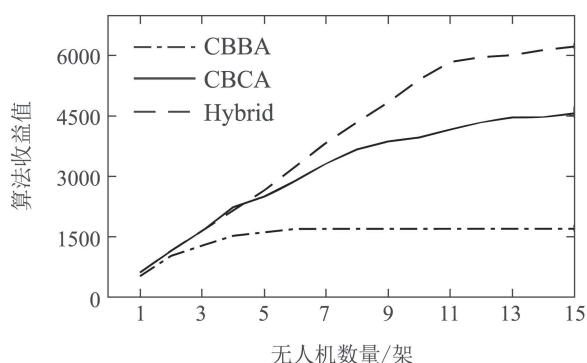


图2 算法总收益值对比结果

Fig. 2 Comparison of total score

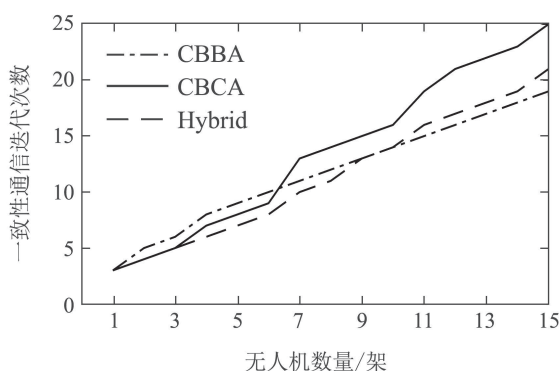


图3 一致协商通信次数对比结果

Fig. 3 Comparison of consensus communication steps

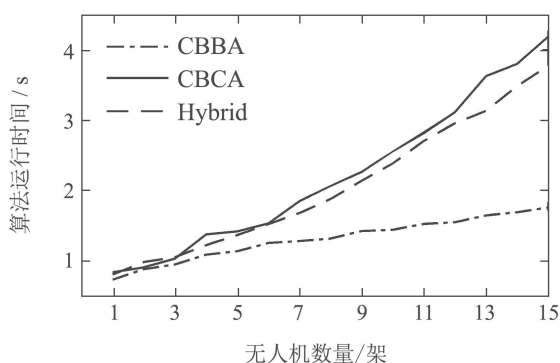


图4 算法运行时间对比结果

Fig. 4 Comparison of computational run time

由图2对比结果可知,随着UAV数量不断增加,各场景对应的算法收益值不断增加;但当UAV数量增加到一定程度时,由于任务数量有限,智能体数量过饱和导致随着UAV数量增加,CBBA算法收益值保持不变。由图3和图4可以看出,CBCA算法平均运行时间更长,为获得一致无冲突分配结果所进行的协商次数更多。造成这一结果的原因是联盟任务分配复杂度更高。

4.3 算法性能评估

为了进一步评估CBCA算法质量,将CBCA算法与基于合同网协议(contract net protocol, CNP)的分布式任务分配算法^[17]进行比较,分别记录算法平均总收益值和平均运行时间,从而对CBCA算法性能进行评估。

算例3中5架无人机对9个目标进行攻击,随机产生无人机和目标的初始位置,任务空间范围设定为 $5\text{ km} \times 5\text{ km} \times 2\text{ km}$,任务静态收益均为100,时间折扣因子为0, $\Delta t_{\min} = 10\text{ min}$ 。设定蒙特卡洛仿真次数为50次。对比结果如图5和图6所示。

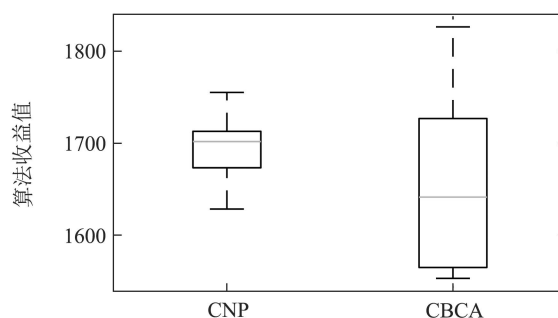


图5 CNP算法与CBCA算法平均收益值对比

Fig. 5 Comparison of average total score based on CNP algorithm and CBCA

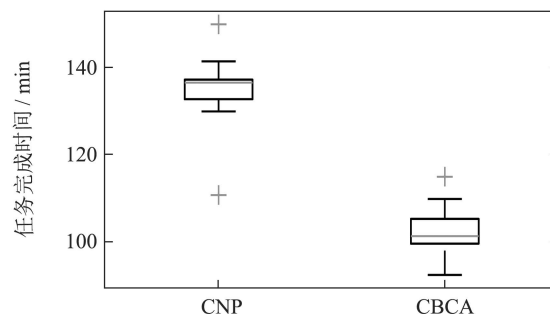


图6 CNP算法与CBCA算法平均任务完成时间对比

Fig. 6 Comparison of average mission time based on CNP algorithm and CBCA

算例3中CBCA算法平均收益值为1641.32,平均任务完成时间为101.3 min;合同网算法平均收益值为1701.803,平均任务完成时间为136.5 min。仿真结果表明,CNP算法任务分配收益值略高于CBCA算法,但CBCA算法平均任务完成时间明显缩短,更符合快速高效的打击任务需求。造成这一结果的主要原因在于:1) CNP算法中拍卖智能体作为管理者,接收满足资源需求的竞拍智能体标书,依据全局最大效能原则构造联盟,而基于贪婪策略的分布式CBCA算法中每个智能体并不掌握周围相邻智能体的信息,以个体边际效益最大为投标原则,因而导致CBCA算法收益值相对较低;2) CNP算法中拍卖智能体需向通信范围内

的智能体发送目标位置信息和所需资源信息作为合同, 收到合同的智能体根据合同判断是否有能力执行该任务. 若具备该目标所需资源, 则成为竞拍智能体并向拍卖智能体投递包括预期路径长度和所携资源等信息的标书. 拍卖智能体根据收到的标书决定联盟成员, 并发送中标和落标信息. 智能体间的广播招标与投标导致CNP算法通信量较大, 任务分配效率降低, 而CBCA算法中各智能体仅需广播自身任务分配信息, 算法通信量较小, 适用于实时动态的战场环境.

进一步地, 采用基于蒙特卡洛多次试验结果的算法评估准则对算法整体优劣性进行评估^[1], 即

$$Q = \sum_{i=1}^{N_m} Q_i, \quad (16a)$$

$$Q_i = \alpha \times \frac{R_i}{R_{\max}} + \beta \times \frac{t_{\min}}{t_i}, \quad (16b)$$

式中: N_m 为蒙特卡洛试验次数; Q_i 为算法单次试验分数; Q 为算法总试验分数; α 和 β 分别为算法规划结果最优性和规划快速性权重系数; $R_{\max}$ 为蒙特卡洛试验中最大收益值; $R(i)$ 为第 i 次试验任务分配收益值; $t_{\min}$ 为蒙特卡洛试验中算法最短运行时间; $t(i)$ 为第 i 次试验算法运行时间. 设定 $N_m = 50$, 并分别设置3种权重系数组合:

- 1) $\alpha = 0.6, \beta = 0.4$;
- 2) $\alpha = 0.5, \beta = 0.5$;
- 3) $\alpha = 0.4, \beta = 0.6$.

算法评估结果如图7所示.

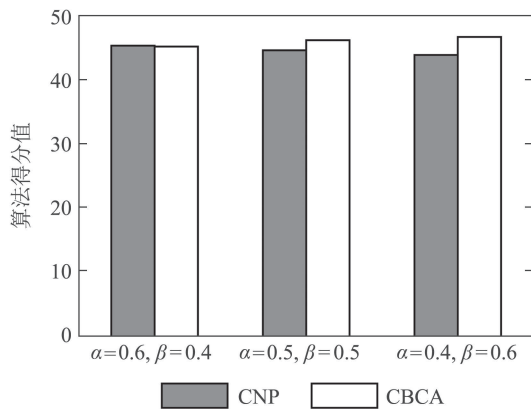


图 7 CNP算法与CBCA算法性能对比结果

Fig. 7 Performance comparison of CNP Algorithm and CBCA

由上述结果可知, CBCA算法总体性能优于CNP算法, 当算法规划快速性权重由0.4增至0.6时, CBCA算法相对于CNP算法得分差值由-0.135增加至2.82; 当算法快速性权重与结果最优性权重相同时, CBCA算法相对于CNP算法得分差值为1.592. 因此, 在保证求解质量的前提下, CBCA算法与CNP算法相比实时性较强.

5 结论

针对复杂任务需要携带不同载荷资源的异构多智能体协同完成且任务之间存在耦合关系的问题, 对一致性包算法的数据结构和冲突消解规则进行了改进, 提出了一致性联盟算法. 仿真结果表明该方法可有效解决考虑任务窗口约束、任务时序约束及任务载荷资源需求等条件下的异构多智能体联盟任务分配问题; 在保证求解质量的同时算法实时性较强. 一致性联盟算法在动态不确定任务环境下的应用将是下一步研究的内容.

参考文献:

- [1] WU Weinan, CUI Naigang, GUO Jifeng. Distributed task assignment method based on local information consensus and target estimation. *Control Theory & Applications*, 2018, 35(4): 566 – 576.
(吴蔚楠, 崔乃刚, 郭继峰. 基于目标信息估计的分布式局部协调任务分配方法. 控制理论与应用, 2018, 35(4): 566 – 576.)
- [2] CAO Lei, TAN Heshun, PENG Hui, et al. Mixed dynamic task allocation for multiple UAV. *Journal of Nanjing University of Science and Technology*, 2015, 39(2): 206 – 214.
(曹雷, 谭何顺, 彭辉, 等. 一种多UAV混合动态任务分配方法. 南京理工大学学报, 2015, 39(2): 206 – 214.)
- [3] KORSAN G, STENTZ A, DIAS M. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotic Research*, 2013, 32(12): 1495 – 1512.
- [4] KORSAN G. *Exploring bounded optimal coordination for heterogeneous teams with cross-schedule dependencies*. Pittsburgh: Carnegie Mellon University, 2011.
- [5] LU Shaolei, FANG Hao. An improved distributed coalition formation algorithm in MAS. *Control and Decision*, 2017, 32(4): 632 – 636.
(卢少磊, 方浩. 一种改进的多agent分布式联盟生成算法. 控制与决策, 2017, 32(4): 632 – 636.)
- [6] YAN Ji, LI Xiangmin, LIU Bo. Multi-agents cooperative task allocation with precedence constraints. *Control and Decision*, 2015, 30(11): 2000 – 2003.
(颜骥, 李相民, 刘波. 考虑时序约束的多智能体协同任务分配. 控制与决策, 2015, 30(11): 2000 – 2003.)
- [7] CASADO E, SCARLATTI D, CAMPILO D, et al. *Network of unmanned vehicles*. U.S. Patent No.8914182. Chicago, US: BOEING CO, 2014.
- [8] JOHNSON L, CHOI H L, HOW J P. The hybrid information and plan consensus algorithm with imperfect situational awareness. *Distributed Autonomous Robotic Systems*. Tokyo: Springer, 2016, 112: 221 – 233.
- [9] GASCO P S, SHIN H, TSOURDOS A, et al. A combinatorial auction framework for decentralized task allocation. *IEEE Globecom Workshops*. Austin: IEEE, 2014: 1445 – 1450.
- [10] CHOI H L, BRUNET L, HOW J P. Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics*, 2009, 25(4): 912 – 926.
- [11] WHITTEN A K, CHOI H L, JOHNSON L B, et al. Decentralized task allocation with coupled constraints in complex missions. *Proceedings of the American Control Conference*. San Francisco: IEEE, 2011: 1642 – 1649.

- [12] CHOI H L, KIM K S, JOHNSON L B, et al. Potential game-theoretic analysis of a market-based decentralized task allocation algorithm. *Distributed Autonomous Robotic Systems*. Tokyo: Springer, 2016, 112: 207 – 220.
- [13] GALLUD X, SELVA D. Agent-based simulation framework and consensus algorithm for observing systems with adaptive modularity. *System Engineering*, 2018, 21(5): 432 – 454.
- [14] WHITTEN A. *Decentralized planning for autonomous agents cooperating in complex missions*. Cambridge: Massachusetts Institute of Technology, 2010.
- [15] BUTLER Z, HAYS J. Task allocation for reconfigurable teams. *Robotics and Autonomous Systems*, 2015, 68: 59 – 71.
- [16] JOHNSON L, PONDA S, CHOI H L, et al. Asynchronous decentralized task allocation for dynamic environments. *Proceedings of the AIAA Infotech@Aerospace Conference*. St. Louis: AIAA, 2011: 29 – 31.
- [17] YAN Fei, ZHU Xiaoping, ZHOU Zhou, et al. Real-time task allocation for a heterogeneous multi-UAV simultaneous attack. *Scientia Sinica Informations*, 2019, 49(5): 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(5): 555 – 569.)

作者简介:

唐嘉钰 博士研究生, 目前研究方向为多无人机指挥控制建模与决策, E-mail: 342118507@qq.com;

李相民 教授, 博士生导师, 目前研究方向为武器系统与运用工程、火力指挥与控制, E-mail: xiangmin_li@163.com;

代进进 讲师, 目前研究方向为火力指挥与控制, E-mail: dj8213@163.com;

薄宁 工程师, 目前研究方向为武器控制与仿真技术, E-mail: fendier@126.com.