

## 考虑时序约束的多智能体协同任务分配

颜 骥<sup>1</sup>, 李相民<sup>1,2</sup>, 刘 波<sup>2</sup>

(1. 海军航空工程学院 兵器科学与技术系, 山东 烟台 264001;

2. 洛阳光电设备研究所 光电控制技术重点实验室, 河南 洛阳 471023)

**摘要:** 研究多智能体系统的多目标多任务分配问题, 考虑任务之间的时序关系, 建立分布式任务分配模型. 扩展了一致性包算法(CBBA), 按优先级将目标任务归入不同层级, 各智能体在构建任务包和任务路径时, 只将分配过高阶段任务的目标添加至相应的任务包和任务路径中, 从而保证目标任务时序约束的同时, 保持了CBBA算法的特性. 与多任务分配问题经典算法的比对实验表明, 所提出的改进算法求解结果稳定可靠, 运行时间优于经典算法.

**关键词:** 一致性包算法; 分布式任务分配; 多智能体系统; 时序约束

**中图分类号:** TP273

**文献标志码:** A

## Multi-agents cooperative task allocation with precedence constrains

YAN Ji<sup>1</sup>, LI Xiang-min<sup>1,2</sup>, LIU Bo<sup>2</sup>

(1. Department of Ordnance Science and Technology, Naval Aeronautical and Astronautical University, Yantai 264001, China; 2. Science and Technology on Electro-optic Control Laboratory, Luoyang Electro-optical Equipment Research Institute, Luoyang 471023, China. Correspondent: YAN Ji, E-mail: 53072472@qq.com)

**Abstract:** The multi-target, multi-task assignment problem for multi-agents systems is studied. By considering the precedence relationships between the tasks, the model of the distributed task allocation for agents is established. The consensus-based bundle algorithm(CBBA) is extended by distributing tasks to different layers according to its precedence. Targets are only allowed to add to the bundle and path of an agent when its precedence task has been assigned, so that the precedence constrains are satisfied, and the properties of CBBA are preserved. The simulation of comparing verifies the proposed task planning methodology for complex missions.

**Keywords:** consensus-based bundle algorithm; distributed task allocation; multi-agent system; precedence constrains

## 0 引 言

多智能体间的协同任务分配是实现多智能体系统协同控制的基础, 其求解策略包括集中式和分布式两类. 集中式方法<sup>[1-2]</sup>能产生比分布式方法更优的解且智能体具有较小的计算负载, 但最优解的产生依赖于中心节点及其与各智能体间的有效通信. 为充分利用各智能体的自主能力来提高求解效率, 避免集中式求解框架存在的中央节点负荷大、系统鲁棒性差等缺陷, 分布式任务分配方法逐渐成为学术界关注的热点.

在非全连通通信网络结构下, 用分布式方法解决多智能体任务分配问题较复杂. 为获得无冲突的分配, 智能体必须在态势感知(SA)和(或)任务规划上达成一致. 文献[3-4]采取智能体在态势感知上达成一致的方式求解全局问题, 但为实现态势感知的一致, 需

要在网络上传输大量的数据<sup>[5]</sup>. 为降低网络上的信息传输量, 文献[5]通过在态势感知达成一定程度一致的基础上加入计划协同的闭环控制, 使得算法能有效应对态势感知的不确定性; 文献[6]提出了解决连续问题的通用框架; 文献[7]利用Dantzig-Wolfe分解方法将整数规划问题在多智能体之间分布. 但这些方法只是针对特定的优化问题, 并不具有普遍性. 另一种方法是采用拍卖算法, 对智能体任务规划进行协调. 文献[8]由临近任务的拍卖者协调编队的行动; 文献[9]提出满足实时性要求的基于多agent分布协同拍卖的动态目标分配算法, 能在规定的时间约束下给出较好的接近最优的分配方案; 文献[10]针对大规模的智能体任务分配问题, 改进了文献[9]的拍卖机制. 不同于上述基于协调中心或共享存储空间的冲突消

收稿日期: 2014-07-16; 修回日期: 2014-10-16.

基金项目: 航空科学基金项目(20135184008).

作者简介: 颜骥(1984-), 男, 讲师, 博士生, 从事武器装备与作战指挥一体化技术的研究; 李相民(1965-), 男, 教授, 博士生导师, 从事武器装备与作战指挥一体化技术等研究.

解方法, 文献[11]提出了一种分布式的控制框架, 各智能体通过与邻接智能体的通信, 独立地利用局部信息获得无冲突的任务分配, 从而为通信能力受限的网络化多智能体系统任务分配问题提供了新的求解策略. 在此基础上, 文献[12]将无人机执行某项任务的能力视为一个 agent, 在保证任务分配时序关系的前提下, 基于分布式拍卖算法实现了对多机协同任务分配问题的优化求解.

一致性包算法(CBBA)<sup>[13]</sup>利用拍卖算法的计算高效性和对不确定态势感知的鲁棒性, 以及一致性算法对网络拓扑的鲁棒性特征, 能为多智能体多任务分配问题提供可行、无冲突的解. 不同于文献[11-12]各智能体每次竞拍一个任务的串行拍卖方式, CBBA通过构建一个任务包, 根据任务对包的收益增量对任务投标. 由于只考虑单个任务包, 智能体的计算量大为减少; 多任务的并行分配又使得CBBA算法的收敛速度较之串行拍卖方法大幅提高. 但是, 基本CBBA算法认为各任务具有相同的优先级, 不适合求解具有优先级(时序)约束的任务分配问题, 而在很多多智能体任务分配应用中, 任务的时序约束是很常见的, 如无人机编队执行对敌防空火力压制或对地攻击任务时, 对目标需依次遂行侦察、打击和评估任务. 类似文献[12]的工作, 本文通过对CBBA算法进行扩展, 使之适应于求解具有时序约束的多任务分配问题.

## 1 问题描述

### 1.1 时序约束任务分配问题

设  $N_u$  智能体集合用于服务  $N_t$  目标, 每个目标具有  $N_m$  时序约束的任务, 如无人机编队对目标的确认、攻击和毁伤效果评估任务. 问题优化目标是对所有任务的服务时间最小, 采用文献[14]的分布式整数规划模型, 问题可描述为

$$\begin{aligned} \max \quad & \sum_{i=1}^{N_u} \left( \sum_{j=1}^{N_t} \sum_{k=1}^{N_m} c_{ij}^k(\tau_{ij}^k(\mathbf{p}_i(\mathbf{x}_i))) x_{ij}^k \right). \\ \text{s.t.} \quad & \begin{cases} \sum_{j=1}^{N_t} \sum_{k=1}^{N_m} x_{ij}^k \leq L_t, \forall i; \\ \sum_{i=1}^{N_u} x_{ij}^k \leq 1, \forall j, k; \\ \sum_{i=1}^{N_u} \sum_{j=1}^{N_t} \sum_{k=1}^{N_m} x_{ij}^k = \min(N_u L_t, N_t N_m). \end{cases} \end{aligned} \quad (1)$$

其中:  $x_{ij}^k \in \{0, 1\}$  为智能体  $i$  执行目标  $j$  的第  $k$  项任务;  $\mathbf{p}_i$  为智能体  $i$  执行所分配任务的先后顺序; 非负的收益  $c_{ij}^k(\cdot)$  为所分配的任务  $\mathbf{x}_i$  和(或)  $\mathbf{p}_i$  的函数;  $L_t$  为能指派给智能体的任务数上限;  $N_m$  为每个目标的任务数; 目标函数括号内的和式表示智能体  $i$  的局部收益. 上述问题公式化表述基于以下假设:

1) 智能体  $i$  通过执行任务  $j^k$  所获得的收益  $c_{ij}^k$  是智能体到达任务时间  $\tau_{ij}^k$  的函数;

2) 到达时间  $\tau_{ij}^k$  由智能体  $i$  任务执行路径  $\mathbf{p}_i$  唯一决定;

3) 任务执行路径  $\mathbf{p}_i$  由智能体  $i$  的任务分配向量  $\mathbf{x}_i$  唯一决定.

该问题的难点在于任务必须按一定的次序执行, 对于可行解, 除应满足式(1)中的约束外, 任务之间还应满足如下关系:

$$t_{j,k+1} > t_{j,k}, \quad \forall j, k, \quad (2)$$

即对于所有的目标和任务, 目标  $j$  的第  $k+1$  项任务必须在其第  $k$  项任务执行之后执行. 只有同时满足约束(1)和(2)的解才是问题的可行解.

### 1.2 一致性包算法

本文通过扩展一致性包算法(CBBA)求解具有时序约束的多任务分配问题. CBBA由分离的两阶段组成: 任务选择阶段和冲突消解阶段. 任务选择阶段, 每个智能体局部地构造其任务包  $\mathbf{b}_i$  和任务执行路径  $\mathbf{p}_i$ . 各智能体将新任务添加至当前任务包和任务路径中所获得的收益, 与当前获胜投标列表  $\mathbf{y}_i$  中该任务的标价值相比, 若其收益值大于当前获胜标价值, 则将该任务分配给自己. 冲突消解阶段, 各智能体与邻接智能体通信, 利用获胜投标列表  $\mathbf{y}_i$ 、获胜智能体列表  $\mathbf{z}_i$  和表示编队成员信息延迟的时间戳向量  $\mathbf{s}_i$  来解决冲突, 以在任务分配上达成一致. 冲突消解引起的获胜标价和获胜智能体更新最终将改变各智能体的任务包和任务执行路径. 如图1所示, 算法在此两阶段循环迭代, 最后能达到在可行分配上的一致.

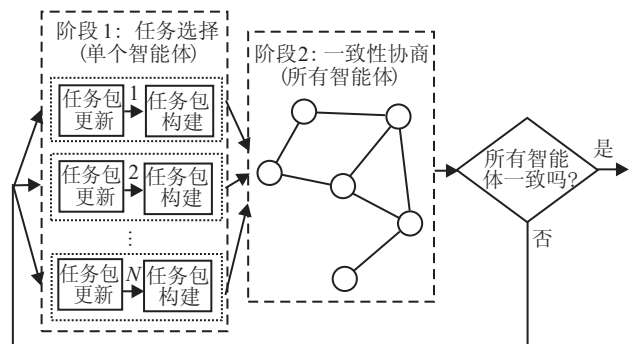


图1 CBBA算法框图

## 2 基于扩展CBBA算法的问题求解

### 2.1 信息结构

假设各智能体知道编队的规模, 且每个智能体在编队中拥有唯一的身份标识  $\mathcal{I} \triangleq \{1, 2, \dots, N_u\}$ , 每个目标标识用  $\mathcal{J} \triangleq \{1, 2, \dots, N_t\}$  表示. 智能体了解对目标需执行的任务数  $N_m$ , 其能执行的最大任务数为  $L_t$ . 在任务分配过程中, 智能体  $i$  需存储和更新如下信

息结构:

1) 任务包. 在 CBBA 中, 任务包用向量  $\mathbf{b}_i \in (\mathcal{J} \cup \{\emptyset\})^{L_t}$  表示, 任务按加入包的先后顺序排列. 为解决多任务分配的时序约束关系, 本文中任务包  $\mathbf{b}_i$  由  $N_m$  向量的合并构成, 即

$$\mathbf{b}_i = \mathbf{b}_i^1 \oplus_{\text{end}} \mathbf{b}_i^2 \oplus_{\text{end}} \cdots \oplus_{\text{end}} \mathbf{b}_i^{N_m}, |\mathbf{b}_i| \leq L_t. \quad (3)$$

其中:  $\oplus_{\text{end}}$  表示将该符号之后的向量中的分量, 不改变次序, 添加到该符号之前向量的最后一个分量之后, 形成新的向量.  $\mathbf{b}_i^k \in (\mathcal{J} \cup \{\emptyset\})^{L_t}$  表示智能体  $i$  执行的所有任务中处于第  $k$  层的任务, 任务按加入包中的先后次序排列.

2) 任务路径. 任务路径  $\mathbf{p}_i$  由  $N_m$  向量合并而成, 即

$$\mathbf{p}_i = \mathbf{p}_i^1 \oplus_{\text{end}} \mathbf{p}_i^2 \oplus_{\text{end}} \cdots \oplus_{\text{end}} \mathbf{p}_i^{N_m}, |\mathbf{p}_i| \leq L_t. \quad (4)$$

其中:  $\mathbf{p}_i^k \in (\mathcal{J} \cup \{\emptyset\})^{L_t}$  表示智能体  $i$  在任务执行路径上处于第  $k$  层的任务, 任务按执行的先后顺序排列.

3) 获胜标价列表. 获胜标价列表用矩阵  $\mathbf{y}_i$  表示, 矩阵的行表示目标任务层, 如在确认、打击和评估的任务中, 第 1 行包含智能体所有将执行的、对目标的确认任务. 矩阵的列表示编队服务的目标, 则智能体  $i$  的获胜标价列表可表示为  $\mathbf{y}_i \in \mathbf{R}_+^{N_m \times N_t}$ , 其中  $y_{ij}^k$  表示目标  $j$  第  $k$  项任务的获胜标价.

4) 获胜智能体列表. 获胜智能体列表用矩阵  $\mathbf{z}_i \in \mathcal{I}^{N_m \times N_t}$  表示, 其存储智能体  $i$  当前掌握的哪个智能体赢得哪些任务的信息,  $z_{ij}^k$  表示目标  $j$  的任务  $k$  的获胜智能体.

5) 时间戳. 时间戳  $\mathbf{s}_i \in \mathbf{R}^{N_u}$  表示智能体  $i$  从编队其他成员中获得的信息的更新时间, 该向量是冲突消除阶段的重要指标, 用于指示智能体从其他智能体获得信息的新旧程度.

## 2.2 任务包构建

任务包构建过程包含任务包构造和任务路径构造两部分. 由 2.1 节可知, 为确保问题的解满足时序约束, 智能体  $i$  任务包的构造可分解为对位于不同层的任务进行分包构造, 只有当位于高层的任务完成后, 才能进入下一层任务包的构造过程. 同理, 在任务路径构造过程中, 新加入的候选任务也只能在其所属的任务层选择能使任务收益最大的任务执行路径.

本文采用文献 [13] 中的时间折扣收益来计算智能体  $i$  执行分配任务所获得的总体收益, 即

$$S_i^{\mathbf{p}_i} = \sum \lambda_{jk}^{\tau_i^{jk}(\mathbf{p}_i)} \bar{c}_{jk}. \quad (5)$$

其中:  $\tau_i^{jk}(\mathbf{p}_i)$  为智能体  $i$  沿路径  $\mathbf{p}_i$  到达目标  $j$  执行任务  $k$  的估计 (计划) 时间,  $\bar{c}_{jk}$  为目标  $j$  任务  $k$  的静态收益,  $\lambda_{jk} < 1$  为目标  $j$  任务  $k$  的折扣因子.

利用加入新任务  $j^k$  前后智能体  $i$  的收益值, 任务

$j^k$  的边际增益可计算为

$$c_{ij}^k[\mathbf{b}_i^k] = \begin{cases} 0, j \in \mathbf{b}_i^k; \\ \max_{n^k \leq |\mathbf{p}_i^k|+1} S_i^{\mathbf{p}_i^k \oplus_{n^k} \{j\}} - S_i^{\mathbf{p}_i^k}, \text{ otherwise.} \end{cases} \quad (6)$$

其中: 标识  $n^k$  是新加入的任务在其对应任务层路径  $\mathbf{p}_i^k$  中的所有可能位置, 而不是在  $\mathbf{p}_i$  中的所有可能位置;  $\oplus_{n^k}$  表示将该符号后面的列表插入到符号前面列表的第  $n$  位, 原列表包括第  $n$  位的元素位置依次后移. 智能体包构造算法如下.

**算法 1** 智能体  $i$  任务包构建过程.

输入: 上次迭代各任务层的任务包  $\mathbf{b}_i^k$ 、任务路径  $\mathbf{p}_i^k$ 、获胜标价列表  $\mathbf{y}_i$ 、获胜智能体列表  $\mathbf{z}_i$ ;

输出: 本次迭代各任务层的任务包  $\mathbf{b}_i^k$ 、任务路径  $\mathbf{p}_i^k$ 、获胜标价列表  $\mathbf{y}_i$ 、获胜智能体列表  $\mathbf{z}_i$ .

Step 1: for  $k := 1$  to  $N_m$ , 分别按式 (3) 和 (4) 构建任务包  $\mathbf{b}_i$  和任务路径  $\mathbf{p}_i$ ;

Step 2: 按式 (5) 计算智能体  $i$  的当前任务收益  $S_i^{\mathbf{p}_i}$ ;

Step 3: 若  $|\mathbf{b}_i| < L_m$ ,  $L_m = \min(L_t, N_m N_t)$ , 则按式 (6) 计算第  $k$  层任务包  $\mathbf{b}_i^k$  所有候选任务的边际增益值  $c_{ij}^k$ , 转入 Step 4, 否则算法结束;

Step 4: 令  $h_{ij}^k = I(c_{ij}^k > y_{ij}^k)$ ,  $\forall j \in \mathcal{J}$ , 若  $\mathbf{h}_i^k \neq \mathbf{0}$ , 则转入 Step 5, 否则转入 Step 9;

Step 5: 找出第  $k$  层所有候选任务中边际增益最大的任务  $J_i^k = \arg \max_j c_{ij}^k h_{ij}^k$ ;

Step 6: 找出任务  $J_i^k$  在第  $k$  层任务路径  $\mathbf{p}_i^k$  中的最佳位置  $n_{i,J_i^k}^k = \arg \max_{n^k} S_i^{\mathbf{p}_i^k \oplus_{n^k} \{J_i^k\}}$ ;

Step 7: 更新第  $k$  层任务包和任务路径:  $\mathbf{b}_i^k = \mathbf{b}_i^k \oplus_{\text{end}} \{J_i^k\}$ ,  $\mathbf{p}_i^k = \mathbf{p}_i^k \oplus_{n_{i,J_i^k}^k} \{J_i^k\}$ ;

Step 8: 更新任务  $J_i^k$  的任务获胜标价和获胜智能体  $y_{i,J_i^k}^k = c_{i,J_i^k}^k$ ,  $z_{i,J_i^k}^k = i$ ;

Step 9:  $k = k + 1$ , 转入 Step 1.

包构建过程不断在不同层的任务间持续, 直至任务包构建完毕, 即包中的任务数达到智能体能处理的任务数上限  $L_t$ , 或者达到所有需处理的任务数  $N_m \cdot N_t$ , 由  $|\mathbf{b}_i|$  确定; 抑或没有更合适的任务可由智能体  $i$  来处理, 由  $\mathbf{h}_i \in \{0, 1\}^{N_m \times N_t}$  确定.  $I(\cdot)$  为指标函数, 若条件满足, 则  $I(\cdot) = 1$ , 否则  $I(\cdot) = 0$ .

## 2.3 冲突消解与任务包更新

任务包构建过程结束后, 智能体与邻接智能体共享 2.1 节中的信息 3)~5), 并按照一定的行动规则更新相应任务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, 异步通信机制见文献 [14].

冲突消解过程利用时间戳向量来确定最新的分配信息. 每次智能体接收到来自邻接智能体的信息, 则对该信息标记如下时间戳:

$$s_{il} = \begin{cases} \tau_r, & g_{il} = 1; \\ \max_{m: g_{im}=1} s_{ml}, & \text{otherwise.} \end{cases} \quad (7)$$

其中:  $g_{il} = 1$  表示智能体  $i$  与智能体  $l$  之间存在通信链接, 否则为 0,  $g_{ii} = 1$ ;  $\tau_r$  表示信息的接收时间.

智能体的信息更新行动规则参照文献 [13]. 智能体  $i$  接收到邻接智能体  $l$  分配的信息后, 对目标  $j$  的任务  $k$  可采取如下 3 种行动: 1) 更新  $y_{ij}^k = y_{lj}^k, z_{ij}^k = z_{lj}^k$ ; 2) 重置  $y_{ij}^k = 0, z_{ij}^k = \emptyset$ ; 3) 离开  $y_{ij}^k = y_{ij}^k, z_{ij}^k = z_{ij}^k$ .

综上可知, 当智能体  $i$  对所属任务进行更新或重置操作时, 将修改任务的获胜标价和获胜智能体信息. 因信息结构不同, 各智能体经过一致性协商后, 任务选择阶段的任务包更新过程与文献 [13] 有所区别, 各智能体从高层任务起, 依次检查各层任务包中是否有更新或重置的任务, 若有, 则将该任务及其在该层任务包中的后续任务释放, 该层任务包所有下级任务包中的所有任务也释放, 相应各层任务路径和各层任务包中对应的任务在获胜任务列表  $y_i$  和获胜智能体列表  $z_i$  中的获胜标价和获胜智能体值都应重置.

## 2.4 算法结束条件

由文献 [13] 中 CBBA 算法的收敛性分析可知, 在静态连通网络环境下, 若各 agent 所保存的分配信息保持  $\Delta (\Delta \leq N_u - 1, \text{为通信网络直径})$  通信循环不变, 则编队达到分配结果一致状态.

若时变的智能体通信拓扑在任意时刻存在  $\rho < \infty$ , 使得

$$W(\tau(t)) = G(\tau(t)) \cup G(\tau(t+1)) \cup \dots \cup G(\tau(t+\rho-1)) \quad (8)$$

是全连通的, 则 CBBA 算法至多  $\rho L_m$  次迭代后收敛. 其中:  $\tau(t)$  为各智能体第  $t$  次 CBBA 算法迭代的真实时间,  $G(\tau)$  为  $\tau$  时刻编队网络通信拓扑,

## 3 仿真实验

### 3.1 仿真设置

以多架具备侦察、打击、评估一体化能力的无人机对地面已知多个静止目标依次执行相关任务为例, 仿真分析本文所提出算法的有效性. 为便于说明问题, 假设目标和无人机在二维平面内运动, 暂不考虑目标的机动性, 采用终端航向松弛下两点间最短 Dubins 路径距离<sup>[15]</sup>近似无人机在两点间的航行距离. 若无人机对同一目标依次执行相关任务, 则无人机飞行距离只与其最小转弯半径  $r$  有关,  $r$  确定情况下, 飞行距离为常量  $2\pi r$ . 本文假设无人机在无风环境下匀速飞行, 实验中未设定敌方威胁源, 且假设所

有待执行任务均可瞬时完成或执行任务所需时间较少. 实验中, 设  $N_m = 3, L_t = 6$ , 以下数据为某次仿真实验中的目标信息 (见表 1) 和无人机信息 (见表 2).

表 1 任务目标信息表

目标标识	目标坐标	任务类型		
		侦察 C	攻击 A	评估 V
Target-1	(1915, 678)	√	√	√
Target-2	(941, 1567)	√	√	√
Target-3	(556, 1353)	√	√	√
Target-4	(1518, 1218)	√	√	√
Target-5	(99, 1410)	√	√	√

表 2 无人机信息表

无人机标识	初始位置	速度	最小转弯半径	能力类型		
				C	A	V
UAV-1	(72, 372)	20	25	√	√	√
UAV-2	(85, 240)	20	25	√	√	√
UAV-3	(235, 11)	20	25	√	√	√

### 3.2 本文算法实验结果

按照本文设计的算法, 上述想定态势下多无人机多任务分配结果如表 3 所示, 各无人机任务分配计划采用任务目标、任务类型、任务执行时间的格式进行表达. 可以看出, 改进的方法能有效地解决复杂约束条件下的多无人机多任务分配问题.

表 3 多无人机多任务分配计划结果(CBBA)

无人机	航程	任务分配计划
UAV-1	3 114	(Target-5,C,51.926)→(Target-5,A,59.926)→
		(Target-3,A,82.946)→(Target-3,V,90.946)→
		(Target-2,V,112.936)→(Target-5,V,155.722)
UAV-2	2 643	(Target-3,C,60.926)→(Target-2,C,82.457)→
		(Target-2,A,90.457)→(Target-4,A,124.195)→
		(Target-4,V,132.195)
UAV-3	2 752	(Target-4,C,88.078)→(Target-1,C,121.598)→
		(Target-1,A,129.598)→(Target-1,V,137.598)

### 3.3 对比实验结果

在同一想定态势下, 采用文献 [1] 提出的遗传算法对问题求解, 对实验采取 100 次蒙特卡洛仿真, 取其中最优结果进行比对, 实验参数设置同文献 [1]. 根据本文算法的特点, 为便于比较, 不妨设遗传算法中各无人机可完成的任务上限数为  $L_t = 6$ , 以最小化单架无人机最大航程为目标函数, 见文献 [1] 中式 (5)~(9)、(11), 仿真实验结果如表 4 所示.

表 4 多无人机多任务分配计划结果(GA)

无人机	航程	任务分配计划
UAV-1	2 689	(Target-3,C,54.708)→(Target-3,A,62.708)→
		(Target-2,C,84.698)→(Target-4,C,118.44)→
		(Target-4,A,126.44)→(Target-4,V,134.44)
UAV-2	2 551	(Target-5,C,58.53)→(Target-5,A,66.53)→
		(Target-5,V,74.53)→(Target-3,V,97.55)→
		(Target-2,A,119.54)→(Target-2,V,127.54)
UAV-3	2 128	(Target-1,C,90.385)→(Target-1,A,98.385)→
		(Target-1,V,106.385)

表 5 是不同问题规模下改进 CBBA 算法、遗传算法和基于混合整数线性规划 (MILP) 方法<sup>[2]</sup>的问题求解时间对比。

表 5 不同问题规模下 3 种算法运行时间对比

无人机数	目标数	目标任务数	计算时间/s		
			ECBBA	GA	MILP
3	2	2	0.068	1.19	0.05
		3	0.074	1.198	1.06
5	2	2	0.085	1.360	0.5
		3	0.098	1.376	504
4	3	2	0.087	1.44	6.07
		3	0.11	1.39	1620
5	3	2	0.102	1.67	20.9
		3	0.134	1.66	11 580
5	4	2	0.131	1.71	1 116
		3	0.189	1.68	未计算

由上述结果可知: 采用集中式的遗传算法, 算法能从全局角度优化分配各无人机执行相应目标任务, 从而优化结果优于各无人机采用贪婪策略的分布式 CBBA 算法, CBBA 算法目标值为 GA 算法蒙特卡洛最优值均值的 94%; 利用 CPLEX 软件求解的 MILP 方法虽然能得到精确最优解, 但随着问题规模增大, 运行时间急剧增加; CBBA 算法和 GA 算法具有较好的可延展性和较低的计算复杂度, 相对 GA 算法, CBBA 算法则具备更快的计算速度和计算稳定性。

## 4 结 论

本文扩展了分布式的一致性包算法, 使其能有效处理多无人机多任务分配中的时序约束问题。在一定的简化条件下, 由与经典算法对比实验可知, 改进方法在保证结果稳定可靠的同时, 运行时间要优于经典的集中式算法。问题的不同求解方式适应于不同的使命环境, 在通信能力受限的情况下, 本文改进算法具有更优的适应性。考虑随遇目标、编队异构性和不同目标间的特定约束关系将是进一步的研究内容。

## 参考文献(References)

- [1] Shima T, Rasmussen S R. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms[J]. *Computers & Operations Research*, 2006(33): 3252-3269.
- [2] Schumacher C, Chandler P R. Optimization of air vehicle operations using mixed-integer linear programming[R]. Air Force Research Laboratory, Wright Patterson, 2006.
- [3] McLain T W, Beard R W. Coordination variables, coordination functions, and cooperative-timing missions[J]. *J of Guidance, Control, and Dynamics*, 2005, 28(1): 150-161.
- [4] Shima T, Rasmussen S J, Chandler P. UAV team decision and control using efficient collaborative estimation[J]. *J*

of Dynamic Systems, Measurement, and Control, 2007, 129(5): 609-619.

- [5] Alighanbari M, How J P. Decentralized task assignment for unmanned aerial vehicles[C]. *Proc of the 44th IEEE Conf on Decision and Control, and the European Control Conference 2005*. Seville: IEEE, 2005.
- [6] Mathews G M, Whyte H D, Prokopenko M. Decentralized decision making in heterogeneous teams using anonymous optimization[J]. *Robotics and Autonomous Systems*, 2009, 57(3): 310-320.
- [7] Eele A, Richards A. Multi vehicle avoidance using nonlinear branch and bound optimization[C]. *Proc of the AIAA Guidance, Navigation, and Control Conf*. 2009.
- [8] Sujit P B, Beard R. Multiple MAV task allocation using distributed auctions[C]. *Proc of the AIAA Guidance, Navigation, and Control Conf*. 2007: 2007-6452.
- [9] 廖沫, 陈宗基. 基于多 Agent 分布协同拍卖的动态目标分配算法[J]. *北京航空航天大学学报*, 2007, 33(2): 180-183.  
(Liao M, Chen Z J. Dynamic target assignment method based on multi-agent decentralized cooperative auction[J]. *J of Beijing University of Aeronautics and Astronautics*, 2007, 33(2): 180-183.)
- [10] 万路军, 姚佩阳, 孙鹏. 有人-无人作战智能体分布式任务分配方法[J]. *系统工程与电子技术*, 2013, 35(2): 310-316.  
(Wan L J, Yao P Y, Sun P. Distributed task allocation method of manned/unmanned combat Agents[J]. *Systems Engineering and Electronics*, 2013, 35(2): 310-316.)
- [11] Zavlanos M M, Spesivtsev L, Pappas G J. A distributed auction algorithm for the assignment problem[C]. *Proc of the 47th IEEE Conf on Decision and Control*. Cancun, 2008.
- [12] 邸斌, 周锐, 丁全心. 多无人机分布式协同异构任务分配[J]. *控制与决策*, 2013, 28(2): 274-278.  
(Di B, Zhou R, Ding Q X. Distributed coordinated heterogeneous task allocation for unmanned aerial vehicles[J]. *Control and Decision*, 2013, 28(2): 274-278.)
- [13] Choi H L, Brunet L, How J P. Consensus-based decentralized auctions for robust task allocation[J]. *IEEE Trans on Robotics*, 2009, 25(4): 912-926.
- [14] Johnson L B, Ponda S S, Choi H L. Asynchronous decentralized task allocation for dynamic environments[C]. St. Louis: AIAA, 2011.
- [15] Rathinam S, Sengupta R, Darbha S. A resource allocation algorithm for multivehicle systems with nonholonomic constraints[J]. *IEEE Trans on Automation Science and Engineering*, 2007, 4(1): 98-104.