

# 控制与决策

Control and Decision

## 基于动态蚁群劳动分工模型的多AUV任务分配方法

杨惠珍, 王强

引用本文:

杨惠珍, 王强. 基于动态蚁群劳动分工模型的多AUV任务分配方法[J]. 控制与决策, 2021, 36(8): 1911–1919.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.1312>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 求解卫星舱布局问题的蚁群劳动分工优化算法

Ant colony labor division optimization algorithm for satellite module layout design

控制与决策. 2021, 36(7): 1637–1646 <https://doi.org/10.13195/j.kzyjc.2019.1764>

### 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>

### USV&AUV水下目标协同搜索与打击航迹规划

Coordinated path planning of USV&AUV for an underwater target

控制与决策. 2021, 36(4): 825–834 <https://doi.org/10.13195/j.kzyjc.2019.1038>

### 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>

### 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>

# 基于动态蚁群劳动分工模型的多 AUV 任务分配方法

杨惠珍<sup>†</sup>, 王 强

(西北工业大学 航海学院, 西安 710072)

**摘要:** 多水下自主航行器 (autonomous underwater vehicle, AUV) 的动态任务分配问题具有高度非线性、动态不确定性以及多模态的特征, 对多 AUV 任务分配方法的自组织性、鲁棒性以及快速性提出了更高的要求. 动态蚁群劳动分工 (dynamic ant colony's labor division, DACLD) 模型是一种采用分布式框架的群智能算法, 众多行为简单的个体相互作用过程中涌现产生的整体智能行为能很好地适应复杂多变的环境, 在解决任务分配问题上具有很好的柔性. 引入动态蚁群劳动分工中的刺激-响应原理, 建立动态蚁群劳动分工与多 AUV 任务分配问题之间的映射关系, 将任务的状态预测纳入响应阈值, 研究基于动态蚁群劳动分工模型的多 AUV 任务分配方法. 同时, 针对任务分配过程中可能出现的任务冲突现象, 提出新的循环竞争方案以实现最大限度地利用 AUV 资源. 仿真结果表明, 所提出的方法能高效地完成分配过程, 具有很好的自组织性、鲁棒性及快速性.

**关键词:** 任务分配; 动态蚁群劳动分工; 分布式; 自组织性; 鲁棒性; 快速性

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.1312

开放科学(资源服务)标识码(OSID):



**引用格式:** 杨惠珍, 王强. 基于动态蚁群劳动分工模型的多 AUV 任务分配方法 [J]. 控制与决策, 2021, 36(8): 1911-1919.

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

YANG Hui-zhen<sup>†</sup>, WANG Qiang

(School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract:** The problem of dynamic task allocation for autonomous underwater vehicles (AUVs) has high nonlinearity, uncertainty and multi-modal characteristics, which requires that the task allocation method has self-organization, robustness and rapidity. The dynamic ant colony's labor division (DACLD) mechanism is a group intelligence algorithm applying a distributed framework. The overall intelligent behavior that emerges from many simple interactions can be well adapted to complex and variable environment, which is very flexible in solving the problem of task allocation. This paper presents a multi-AUV task allocation method based on DACLD, in which the stimulus-response principle of ant colony division of labor is introduced and the mapping relationship between ant colony division of labor and multi-AUV task assignment is established, and the task state estimation is incorporated into the response threshold. And also, a new cyclic competition scheme is proposed to maximize the utilization of AUV resources for the task conflict phenomenon. The simulation results show that the proposed method can efficiently complete the task allocation process and has good self-organization, robustness and rapidity.

**Keywords:** task allocation; dynamic ant colony's labor division (DACLD); distributed framework; self-organizing; robustness; quickness

## 0 引言

异构多 AUV 协作完成多个任务是当前海洋工程领域的研究热点之一<sup>[1]</sup>. 任务分配通过构建科学健壮的数学模型, 设计优化算法完成任务配置, 使个体的资源得到充分利用, 高效地完成分配任务, 体现了多 AUV 系统的高层组织形式与运行机制. 本文针对异构多

AUV 携带资源有限、任务状态动态变化以及在水下复杂环境中随时面临突发事件的情况, 开展动态任务分配方法研究, 对及时响应未知事件、提高任务完成率以及增强系统鲁棒性具有重要意义.

蚁群算法 (ant colony algorithm, ACO) 作为经典的启发式优化算法<sup>[2]</sup>, 已经成功应用于旅行商问

收稿日期: 2019-09-18; 修回日期: 2020-10-23.

基金项目: 国家自然科学基金联合基金重点支持项目 (U1813225).

<sup>†</sup>通讯作者. E-mail: rainsun\_ly@npu.edu.cn.

题<sup>[3]</sup>、车间调度问题<sup>[4-5]</sup>、多处理器资源分配问题<sup>[6]</sup>、车辆路径问题<sup>[7]</sup>等. 通过对蚁群算法进行改进和拓展, 克服了其易陷入局部最优的劣势, 基于蚁群算法的多机器人任务分配方法研究取得了大量研究成果. 文献[8]引入局部优化变异算子和改进模拟退火算法来解决多机器人任务分配问题; 文献[9]通过非初始任务点选择方式中引入回程优化策略, 以提高算法的搜索效率、加快收敛速度; 文献[10]为使无人机打击任务更有针对性, 基于传统蚁群算法对启发函数进行重新定义, 将战术目标价值与启发信息相结合, 引导蚂蚁向战术价值高的点进行转移, 提高了任务分配的可行性和有效性. 其他启发式搜索算法, 如粒子群算法(PSO)、遗传算法(GA)、狼群算法(WPA)等也在任务分配中得到广泛应用<sup>[11-12]</sup>. 上述启发式算法都是集中式算法. 分布式算法有面向协商的熟人网<sup>[13]</sup>和合同网<sup>[14]</sup>分配方法, 对策论联盟形成策略<sup>[15]</sup>, 依据价格的市场拍卖<sup>[16-17]</sup>, 面向结构的自组织方法<sup>[18-19]</sup>等.

对动态任务分配而言, 无论是集中式算法还是分布式算法, 都需要算法具有一定的快速性, 从而能够对动态变化的任务态势进行及时有效的响应. 上述方法受限于中心节点有限的计算能力或频繁的信息交换, 在动态任务分配中难以保证快速性和有效性<sup>[20]</sup>. 蚁群劳动分工模型是一种群智能进化算法, 它模拟了蚁群自适应的劳动分工行为<sup>[21]</sup>, 在无需知晓任何关于群体需求全局信息的情况下, 自下而上地通过个体的本地意识和响应来产生全局自组织行为, 其结果是每个蚂蚁都在忙碌地工作, 而且任务的分配又恰好符合群体对各项任务的要求<sup>[22-24]</sup>. 文献[25]针对无人机群任务分配的多目标多约束问题, 基于经典的固定响应阈值模型, 提出了动态蚁群的分工模型(DACL D), 用于解决无人机群在针对地面固定目标进行攻击过程中, 新的目标突现时的任务重分配问题. 本文受此启发, 针对异构多AUV动态任务分配问题, 研究基于动态蚁群劳动分工模型的多AUV动态任务分配方法. 本文的主要贡献如下:

1) 建立了异构多AUV动态任务分配问题的数学模型, 用一个多输入、多输出、多约束以及多目标的优化问题来描述该问题. 与文献[25]的不同之处在于, 本文研究的任务是动态变化的, 即任务的位置随时间变化.

2) 根据动态蚁群劳动分工中的刺激-响应原理, 将动态蚁群劳动分工中的响应阈值模型、环境刺激模型及转移概率模型在多AUV任务分配问题上进行一一映射.

3) 任务的动态变化带来了任务分配的复杂性, 为了避免在解决任务冲突问题时AUV资源被闲置, 本文提出一种循环竞争方案, 以实现最大限度地利用AUV资源, 确保每一分配结果的任务完成效率最高.

## 1 异构多AUV任务分配数学模型

### 1.1 问题描述

异构多AUV系统动态任务分配问题是指在一定区域内, 有 $N$ 个异构AUV需要执行 $M$ 个不同特性的任务, 且在执行过程中, 任务的数量和状态(如位置和所需资源等)会发生变化. 为保证任务执行的效率, 需要异构多AUV系统实时根据AUV和任务的态势变化进行动态任务分配.

**定义1** 不同类型的AUV组成多AUV系统, 即  $\mathbf{AUV} = \{AUV_1, AUV_2, \dots, AUV_N\}$ ,  $N$ 表示AUV的个数. 为了更好地表征个体AUV的能力, 将各参数使用一个集合进行描述, 即

$\{\text{State Position Speed Resource Ability Scope}\}$ .

**AUV\_State**——AUV的状态, 指空闲或执行哪个任务;

**AUV\_Position**——AUV所处的位置;

**AUV\_Speed**——AUV载体的航行速度;

**AUV\_Resource**——AUV携带的资源总量;

**AUV\_Ability**——AUV执行任务的能力;

**AUV\_Scope**——AUV执行任务的范围.

**定义2** 需求不同、价值不同的任务组成任务集合, 即  $\mathbf{TASK} = \{\text{Task}_1, \text{Task}_2, \dots, \text{Task}_M\}$ ,  $M$ 表示任务的个数. 为了更好地表征任务的特征, 将各参数使用一个集合进行描述, 即

$\{\text{State Position Resource Value}\}$ .

**TASK\_State**——任务的状态, 指是否被完成;

**TASK\_Position**——任务所处的位置;

**TASK\_Resource**——任务的需求资源数量;

**TASK\_Value**——任务的价值.

**定义3** 任务分配问题可以用集合  $\{\mathbf{TASK}, \mathbf{AUV}, U_{\text{total}}\}$  表示, 即输入一组任务给多AUV系统, 通过合理的任务分配使得系统整体效能最大.

**TASK**——任务集合;

**AUV**——AUV群体集合;

$U_{\text{total}}$ ——系统的效能函数, 即目标函数.

对于传统的任务分配方法(如蚁群算法),  $U_{\text{total}}$ 用于迭代过程中任务分配结果的评定(如蚁群算法中信息素的叠加), 本文 $U_{\text{total}}$ 体现在冲突解决方案的

评价指标(响应阈值)中,进而影响每只蚂蚁(AUV)对任务的竞争.

异构多AUV系统的任务分配是一个多输入、多输出、多约束以及多目标的优化问题. 输入包括各AUV参数、各任务参数,输出包括任务与AUV的分配关系、面向任务的AUV之间资源分配关系. 本文建立如下所述的约束条件.

## 1.2 约束条件

定义决策变量 $x_{(i,j)}$ ,表示AUV $i$ 是否执行Task $j$ ,取值为1表示AUV $i$ 执行Task $j$ ,否则表示不执行,即

$$x_{(i,j)} = \begin{cases} 1, & \text{AUV}_i \rightarrow \text{Task}_j; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

### 1.2.1 状态约束

1) 同一时刻,一个任务可由多个AUV执行,即

$$\sum_{i=1}^N x_{(i,j)} \geq 1, \quad j = 1, 2, \dots, M. \quad (2)$$

2) 同一时刻,任意一个AUV只能执行一个任务,即

$$\sum_{j=1}^M x_{(i,j)} \leq 1, \quad i = 1, 2, \dots, N. \quad (3)$$

3) 同一AUV对于同一任务只能执行一次,即

$$\sum_{c=1}^C x_{(i,j)} \leq 1, \quad (4)$$

其中 $c$ 表示一个AUV执行同一任务的次数.

### 1.2.2 资源约束

AUV个体或AUV个体间自发形成的联盟执行任务时,其携带的资源总量必须满足任务所需的资源总量才能完成该任务. 设联盟中AUV的个数为 $\lambda$ ( $\lambda \geq 1$ ),即

$$\text{for } \forall t, \sum_{i=1}^{\lambda} r_i \geq \text{TASK\_Resource}(t), \quad (5)$$

其中 $r_i$ 表示联盟中第 $i$ 个AUV的剩余资源数量.

综合上述约束条件,建立异构多AUV系统任务分配数学模型,即

$$\begin{cases} \sum_{i=1}^N x_{(i,j)} \geq 1, \quad j = 1, 2, \dots, M; \\ \sum_{j=1}^M x_{(i,j)} \leq 1, \quad i = 1, 2, \dots, N; \\ \sum_{c=1}^C x_{(i,j)} \leq 1; \\ \text{for } \forall t, \sum_{i=1}^{\lambda} r_i \geq \text{TASK\_Resource}(t). \end{cases} \quad (6)$$

## 2 基于动态蚁群劳动分工模型的多AUV动态任务分配方法

### 2.1 定制化模型描述

#### 2.1.1 动态环境刺激值

在动态复杂的水下环境中,任务集合 $\text{TASK} = \{\text{Task}_1, \text{Task}_2, \dots, \text{Task}_M\}$ 中每个任务对应一个环境刺激值 $s_j$ ( $j = 1, 2, \dots, M$ ),组成集合 $s = (s_1, s_2, \dots, s_M)$ ,其大小取决于任务的价值以及是否有AUV执行它,且随着时间变化而变化. 在初始时刻,任务的环境刺激值 $s_j(0)$ 大小仅取决于执行的任务类型和任务价值,即

$$s_j(0) = \frac{V(\text{Task}_j)}{\sum_{j=1}^M V(\text{Task}_j)}, \quad (7)$$

其中 $V(\text{Task}_j)$ 表示Task $j$ 的价值.

任务环境刺激值大小反映任务的紧迫程度,刺激值越大,任务越紧迫,则越容易吸引蚂蚁(AUV)执行该项任务. 各个蚂蚁结合任务的环境刺激值与自身对任务的响应阈值决定是否执行该项任务,如果该项任务没有被执行,则该项任务的环境刺激值随时间变化而变化,其变化规则如下:

$$s_j(t+1) = (1 - \eta) \times (s_j(t) + \delta_j). \quad (8)$$

其中: $\delta_j$ 表示Task $j$ 单位时间内环境刺激值的增量; $\eta$ 表示任务的完成度,

$$\eta = \frac{\text{RT}_j(t+1)}{\text{TASK\_Resource}(j)}. \quad (9)$$

$\text{TASK\_Resource}(j)$ 表示初始状态下完成Task $j$ 所需要的资源总量, $\text{RT}_j(t+1)$ 表示AUV在执行Task $j$ 后Task $j$ 仍需要的资源. AUV在每次执行完Task $j$ 后,任务所需资源的更新规则如下:

$$\text{RT}_j(t+1) = \text{RT}_j(t) - R_{\text{AUVCost}}(t), \quad (10)$$

其中 $R_{\text{AUVCost}}(t)$ 表示在 $t$ 时刻为了执行该任务,AUV所耗费的资源. 当AUV选择执行某些任务后,即AUV的状态从 $\text{state}_\alpha$ 转移到状态 $\text{state}_\beta$ ,则需计算相对环境刺激值<sup>[25]</sup>

$$\bar{s}_{(\text{state}_\alpha \rightarrow \text{state}_\beta)} = \frac{s_{\text{state}_\beta}(t+1)}{s_{\text{state}_\alpha}(t)}. \quad (11)$$

当AUV $i$ 处于空闲状态时,假定环境中存在一个“虚任务”Task $0$ 与之对应且不对其状态进行更新,该虚任务的环境刺激值<sup>[25]</sup>满足以下公式:

$$s_0 = \min(s_1, s_2, \dots, s_M). \quad (12)$$

#### 2.1.2 基于任务预测的动态响应阈值

随着任务态势的变化,AUV执行了一定数量的任务,剩余资源发生变化,响应阈值也随之改变,即

$$\xi_{ij}(t) = \phi_i \cdot \frac{\omega_1 D_{ij}^- + \omega_2 T_{ij}^-}{\varphi_i(t) \cdot \psi_i(t) \cdot \text{AUV\_Ability}(i)}. \quad (13)$$

其中

$$D_{ij}^- = \frac{\hat{D}_{ij}(t+1)}{\sum_{i=1}^n \sum_{j=1}^m \hat{D}_{ij}(t+1)}$$

和

$$T_{ij}^- = \frac{\hat{T}_{ij}(t+1)}{\sum_{i=1}^n \sum_{j=1}^m \hat{T}_{ij}(t+1)}$$

分别表示归一化后无量纲的距离和达到时间的预测值。 $\omega_1$ 和 $\omega_2$ 表示权重。 $n$ 表示剩余资源数量不为零的个数, $m$ 表示任务所需资源不为零的任务个数。 $\phi_i$ 表示学习因子,反映AUV $i$ 的学习能力,其范围为 $\phi_i < 1^{[25]}$ ,且有

$$\phi_i = \begin{cases} \frac{1}{N(t-1)} \cdot \text{Stu}, & N_k(t-1) \neq 0; \\ \text{Stu}, & N_k(t-1) = 0. \end{cases} \quad (14)$$

式中 $N_k(t-1)$ 是智能体 $i$ 在时刻 $t$ 之前执行任务的次数, $\text{Stu} \in (0, 1)$ 是初始学习因子。 $\varphi_i(t)$ 表示 $t$ 时刻AUV $i$ 的资源消耗因子, $\varphi_i(t)$ 的大小与AUV $i$ 最初携带的资源 $\text{RC}_i(0)$ 及 $t$ 时刻自身所拥有的资源 $\text{RC}_i(t)$ 有关,计算方式<sup>[25]</sup>为

$$\varphi_i(t) = \frac{\text{RC}_i(t)}{\text{RC}_i(0)}. \quad (15)$$

当 $\text{RC}_i(t) \neq 0$ 时,资源消耗因子 $\varphi_i(t) \neq 0$ ;当 $\text{RC}_i(t) = 0$ 时,资源消耗因子 $\varphi_i(t) = 0$ ,表示AUV $i$ 的响应阈值随着资源消耗增大而增大,从而达到不执行任务的目的。 $\psi_i(t)$ 表示 $t$ 时刻AUV $i$ 的剩余资源占有率,计算方式为

$$\psi_i(t) = \frac{\text{RC}_i(t)}{\sum_{i=1}^N \text{RC}_i(t)}. \quad (16)$$

当AUV $i$ 在 $t$ 时刻的状态 $\text{state}_\alpha$ 在 $t+1$ 时刻转移到状态 $\text{state}_\beta$ 时,需计算相对响应阈值 $\bar{\xi}_i^{[25]}$ 为

$$\bar{\xi}_i^{(\text{state}_\alpha \rightarrow \text{state}_\beta)} = \frac{\xi_{i(\text{state}_\beta)}(t+1)}{\xi_{i(\text{state}_\alpha)}(t)}. \quad (17)$$

AUV $i$ 对虚任务 $\text{Task}_0$ 的响应阈值<sup>[25]</sup>为

$$\xi_{i0}(t) = \max(\xi_{i1}, \xi_{i2}, \dots, \xi_{iM}). \quad (18)$$

### 2.1.3 转移概率

转移概率指AUV $i$ 从当前状态转移至其他状态的概率,通过比较转移概率的大小,选取概率最大的状态预设为下一步的状态<sup>[22-24]</sup>。 $p_{ij}$ 表示下一时刻AUV $i$ 转去执行任务 $j$ 的转移概率,有

$$p_{ij} = p[x_{(i,j^*)}(t) = 1 \rightarrow x_{(i,j)}(t+1) = 1] =$$

$$\frac{[\bar{s}_{(j^* \rightarrow j)}]^2}{[\bar{s}_{(j \rightarrow j)}]^2 + [\bar{\xi}_{i(j^* \rightarrow j)}]^2}, \quad (19)$$

其中 $j^*$ 表示当前时刻AUV $i$ 所执行的任务编号。

## 2.2 循环竞争的冲突解决方案

一般情况下,AUV选择转移概率最大的任务作为下一步要执行的任务.当有多个AUV同时选择同一任务(假设为第 $e$ 个任务)时,则出现了任务冲突.冲突解决方案涉及以下问题:

1) 如何在冲突的AUV中选择某一个或某几个AUV来执行第 $e$ 个任务?评价标准是什么?

2) 在对第 $e$ 个任务的竞争中落选的AUV是否有可能比在其余任务(假设为第 $f$ 个任务)的竞争中获胜的AUV更适合执行第 $f$ 个任务?如果“是”,则在对第 $e$ 个任务的竞争中落选的AUV与对第 $f$ 个任务的竞争中获胜的AUV再一次构成了一次竞争关系,那么在第2次对第 $f$ 个任务竞争中落选的AUV便又会依 $p_{ij}$ 与 $\xi_{ij}$ 进行第3次的竞争,以此类推,此即为循环竞争.

3) 循环竞争到什么地步便会停止?依据是什么?

以往的冲突解决方案大多只进行一次竞争,被淘汰的智能体便不再参与此次任务分配,处于空闲状态.这种分配方案会造成AUV资源的浪费,且任务执行效率不高.因此,针对以上问题提出如下冲突解决方案:

当多个AUV均依转移概率选择某个任务时,优先选用AUV资源满足任务所需资源的AUV而不采取合作方案.此外,由于AUV的响应阈值 $\xi_{ij}$ 综合了AUV到任务的距离、AUV执行任务所需的时间、AUV的资源消耗因子、AUV的剩余资源占有率及AUV执行任务的能力,故采用 $\xi_{ij}$ 作为冲突解决指标. $\xi_{ij}$ 越小代表AUV执行该任务越优,故原则上尽可能地为每个任务分配较小 $\xi_{ij}$ 的AUV.

当任务执行过程中剩余待执行的任务个数小于仍存活的AUV个数时,可能出现有的AUV在对所有任务的竞争中均落败的现象,此时该AUV被分配为虚任务.

当所有AUV均被分配任务(包含虚任务)时,一次分配过程结束,结束循环竞争方案.这样做既保证了AUV资源的合理利用,又保证了AUV群执行任务的高效性.

冲突解决方案的伪代码表示如下.

**算法1** 冲突解决方案.

1: 计算转移概率 $p_{ij}$ ;

2: while 有AUV未分配任务 do

```

3: 所有 AUV 依  $p_{ij}$  选择最意愿执行的 TASK;
4: if 某一 TASK 被多个 AUV 选择 then
5:   if 该 TASK 在上次循环中已被分配 AUV
6:     将已分配该 TASK 的 AUV 提取出来, 与此
    次循环选择该 TASK 的 AUV 一起竞争该 TASK;
7:   end
8:   if  $RT_j < \min(RC_i)$  then
9:     分配  $\xi_{ij}(t)$  最小的 AUV 执行 Taskj;
10:  else if  $\min(RC_i) < RT_j < \max(RC_i)$  then
11:   在满足  $RC_i \geq RT_j$  的 AUV 中选  $\xi_{ij}(t)$  最
    小的 AUV 分配执行 Taskj;
12:  else if  $\max(RC_i) < RT_j < \sum(RC_i)$  then
13:   将参与竞争的 AUV 依  $\xi_{ij}(t)$  从小到大排
    序;
14:   依次选择 AUV 执行 Taskj 直到
       $\sum(RC_i) \geq RT_j$ ;
15:  else
16:   所有 AUV 分配执行 Taskj;
17:  end
18: 将竞争中淘汰的 AUV 对应的  $p_{ij}$  设为 0;
19:  if  $p_{ij} == 0 (j = 1, \dots, m)$  then
20:   为 AUVi 分配虚任务;
21:  end
22: end
23: end
    
```

**2.3 算法流程**

综合以上内容, 本文提出的基于 DACLD 的多 AUV 任务分配算法的伪代码表示如下.

**算法2** 基于 DACLD 的多 AUV 任务分配算法.

```

1: 初始化 {AUV1, AUV2, ..., AUVN}, {Task1,
    Task2, ..., TaskM},  $t = 0, \phi_i, \varphi_i$ , AUV 及 TASK 禁忌
    表;
2: while 有 TASK 未加入禁忌表 do
3:   计算  $D_{ij}(t), T_{ij}(t), \hat{D}_{ij}(t), \hat{T}_{ij}(t)$ ;
4:   if  $D_{ij}(t) < \text{AUV\_Scope}(i)$  then
5:     AUVi 执行 Taskj 并更新 AUVi 与 Taskj 的
    禁忌表;
6:   end
7:   由式(8)和(13)计算  $s_j(t)$  及  $\xi_{ij}(t)$ ;
8:   由式(11)和(17)计算  $\bar{s}_{(\text{state}_\alpha \rightarrow \text{state}_\beta)}$  以及
     $\bar{\xi}_{i(\text{state}_\alpha \rightarrow \text{state}_\beta)}$ ;
9:   由式(19)计算  $p_{ij}$ ;
10:  依据  $p_{ij}$  及冲突解决方案为每个 AUV 分配
    
```

```

    任务;
11: 更新未加入禁忌表的 AUV 及 TASK 位置;
12: end
    
```

**3 仿真实验**

假设初始时刻有 3 个异构 AUV 和 5 个动态任务, 每个 AUV 根据动态任务分配结果向所分配的任务移动, 任务进入 AUV 的执行范围便会被执行, 直到所有任务被完成, 仿真结束.

这里设置两种场景:

场景 1: 任务的位置改变但数量不变;

场景 2: 任务的位置和数量都变化, 即有突发任务出现.

**3.1 参数设置**

AUV 和任务的信息分别如表 1 和表 2 所示.

**表 1** AUV 参数设置

序号	位置	速度	能力	资源数量
1	[0, 0]	0.35	0.76	10
2	[0, 0]	0.36	0.83	8
3	[0, 0]	0.38	0.65	7

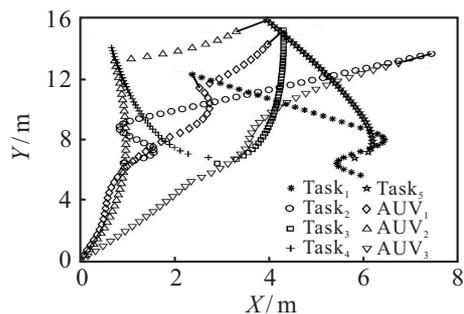
**表 2** TASK 参数设置

序号	初始位置	资源需求	价值
1	[5.89, 5.66]	5	7
2	[0.91, 6.48]	6	9
3	[2.88, 6.44]	4	8
4	[2.69, 6.83]	3	10
5	[5.76, 6.79]	4	8

**3.2 仿真结果**

**3.2.1 场景 1**

通过与基于 ACO、PSO 及 GA 的任务分配方法的结果进行对比, 分析本文所提出的基于 DACLD 的任务分配方法的自组织性、鲁棒性和快速性. ACO、PSO 及 GA 中的个体数目  $k$  为 30 只, 每得到一次任务分配结果的迭代次数  $N_c$  为 100 次. 权重  $\omega_1 = 0.7, \omega_2 = 0.3$ . 仿真结果如图 1 ~ 图 9 所示.



**图 1** DACLD 相对运动结果(场景 1)

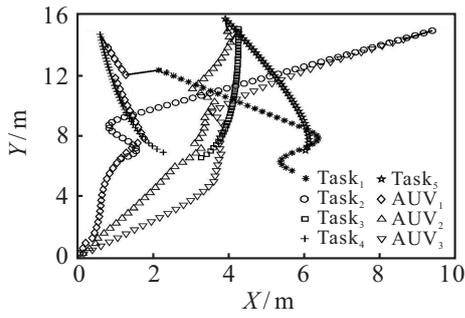


图2 ACO相对运动结果(场景1)

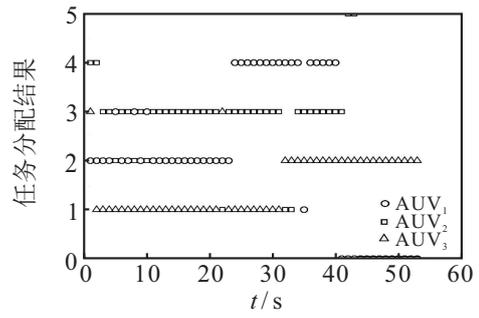


图7 PSO动态任务分配结果(场景1)

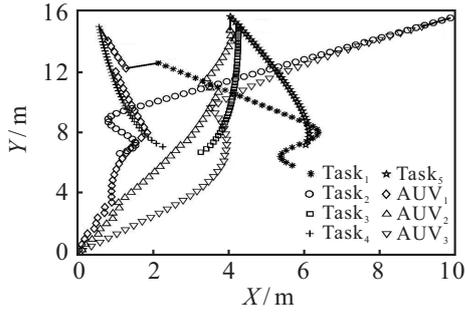


图3 PSO相对运动结果(场景1)

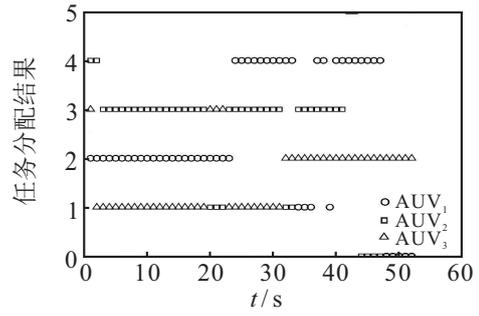


图8 GA动态任务分配结果(场景1)

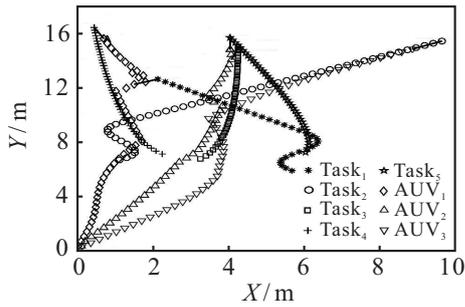


图4 GA相对运动结果(场景1)

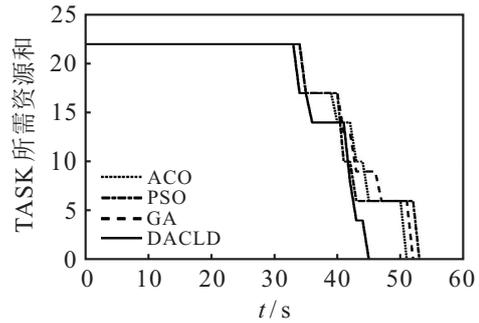


图9 TASK所需资源变化(场景1)

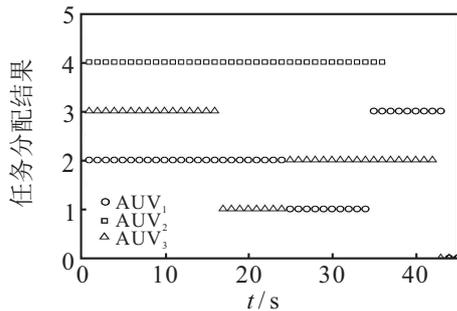


图5 DACLD动态任务分配结果(场景1)

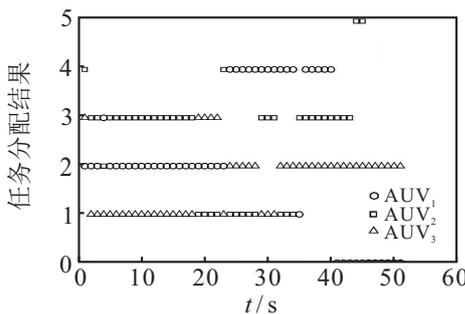


图6 ACO动态任务分配结果(场景1)

初始时刻,对各AUV而言,Task<sub>2</sub>距离最近,由式(13),各AUV对Task<sub>2</sub>的响应阈值比其余Task小,即各AUV对Task<sub>2</sub>的相对响应阈值最小.与此同时,Task<sub>2</sub>的价值为9,相比其余Task较大,由式(11),各AUV对Task<sub>2</sub>的相对环境刺激值较大,由式(19),各AUV对Task<sub>2</sub>的相对转移概率最大,即各AUV均意愿执行Task<sub>2</sub>.由冲突解决方案,在资源约束满足的情况下,选择相应阈值最小的AUV来执行此任务.由于AUV<sub>1</sub>具有较高的能力,且资源占有率最大,故由式(13),AUV<sub>1</sub>具有最低的响应阈值( $\xi_{12}(0) = 0.2038$ ),故初始时刻选择AUV<sub>1</sub>来执行Task<sub>2</sub>.

由图1,  $t = 24s$ 时AUV<sub>1</sub>原本跟踪Task<sub>2</sub>,AUV<sub>3</sub>原本跟踪Task<sub>1</sub>,此时AUV<sub>1</sub>转而更愿意跟踪Task<sub>1</sub>,且根据冲突解决方案在与AUV<sub>3</sub>对Task<sub>1</sub>的竞争中获胜( $\xi_{11}(24) = 0.1887, \xi_{31}(24) = 0.2186$ ),故AUV<sub>1</sub>转而执行Task<sub>1</sub>,而AUV<sub>3</sub>转而执行转移概率较大的Task<sub>2</sub>.从图1可以看到,由于两个任务的位置发生了明显的变化,从而引起响应阈值的变化,交换任务后

的执行效率明显更高。

将图1与图2、图3、图4对比,可以看出:

1) 基于 ACO、PSO 及 GA 的任务分配结果表明 AUV 间未能及时交换任务而导致执行效率不高. 所提出的基于 DACLD 的任务分配方法对动态变化的任务状态有自适应分配能力,具有良好的自组织性。

2) 采用 DACLD 的任务分配方法, AUV 的运动轨迹更加平滑,表明所提出的冲突解决方案具有很好的鲁棒性。

将图5与图6、图7、图8对比,可以看出:

1) DACLD 的任务分配结果相比 ACO、PSO 及 GA 的任务分配结果变化次数少,分配结果更稳定。

2) ACO、PSO 及 GA 的任务分配方法执行完成所有任务用时分别为 51 s、53 s、52 s,而本文的 DACLD 方法用时 45 s,表明所提出的 DACLD 任务分配方法具有更高的执行效率。

表3对比了4种算法的仿真时间. 仿真环境为 Matlab 2018a, 计算机配置为 2.2 GHz Intel(R) Core (TM) i5-5200U 处理器, 12 G RAM, Win10.

算法名称	总用时	单次最大用时	平均每次用时
ACO	18.987	0.526	0.372
PSO	4.375	0.219	0.083
GA	2.984	0.109	0.057
DACL D	0.203	0.019	0.005

由于 ACO、PSO 及 GA 为了保证算法结果收敛,需完成  $N_c \times k$  次任务分配计算才得到一次最终的任务分配结果,而 DACLD 只需要完成一次任务分配计算即可,因而算法复杂度要远小于 ACO、PSO 及 GA. 由表3可知,本文所提方法的算法时间效率要远大于 ACO、PSO 及 GA 的算法时间效率。

### 3.2.2 场景2

假设在  $t = 40$  s 时环境中突然出现两个新任务 Task<sub>6</sub> 与 Task<sub>7</sub>, 任务 Task<sub>6</sub> 位置为 (3.55, 7.28), 价值为 8, 资源需求数量为 4; 任务 Task<sub>7</sub> 位置为 (7.54, 6.32), 价值为 4, 资源需求数量为 3. 仿真结果如图 10~图 16 所示。

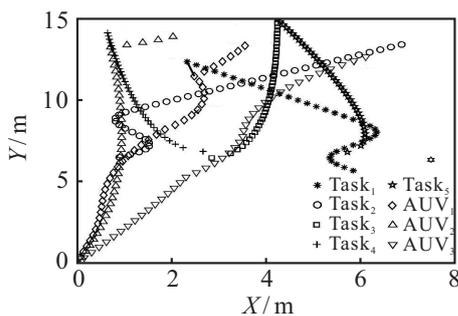


图 10  $t = 40$  s 时的分布图(场景2)

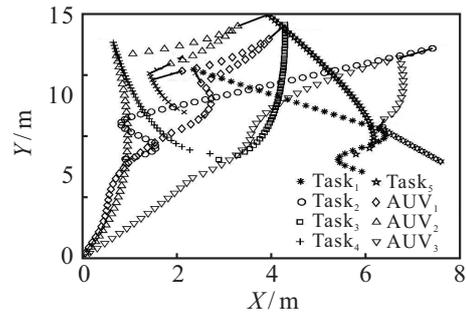


图 11 DACLD 相对运动结果(场景2)

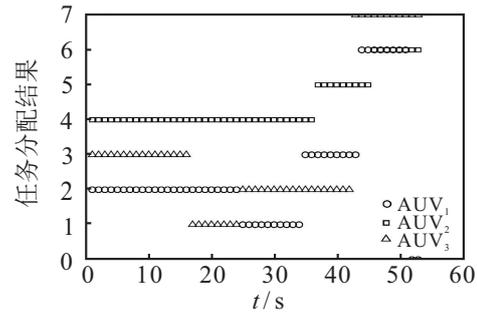


图 12 DACLD 动态任务分配结果(场景2)

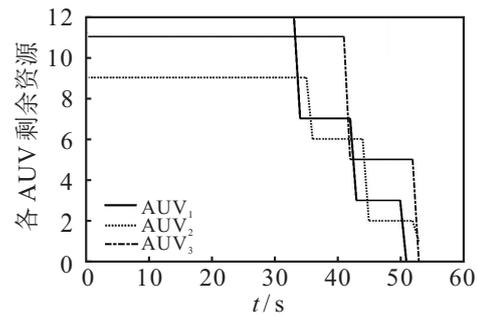


图 13 AUV 的剩余资源变化(场景2)

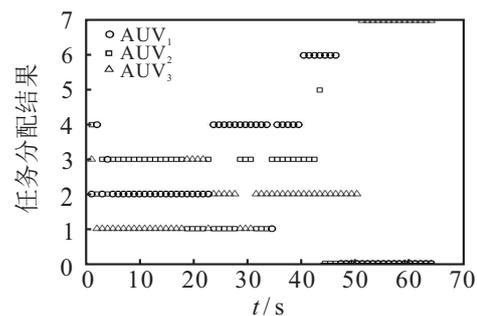


图 14 ACO 动态任务分配结果(场景2)

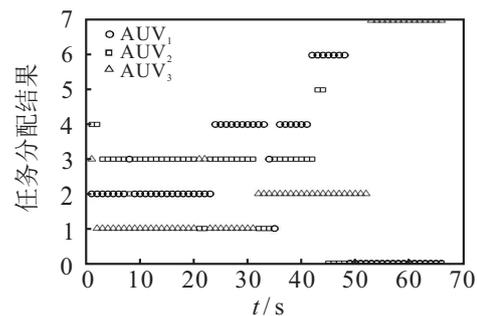


图 15 PSO 动态任务分配结果(场景2)

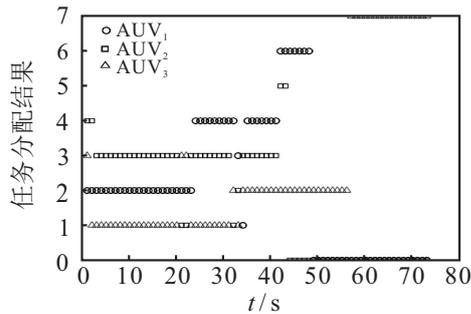


图16 GA动态任务分配结果(场景2)

由图10可知,在 $t = 40\text{ s}$ 时,Task<sub>3</sub>与Task<sub>5</sub>的位置很接近,此时AUV<sub>2</sub>依转移概率 $p_{ij}$ 选择最愿意执行的任务为Task<sub>3</sub>,与AUV<sub>1</sub>发生冲突,此时AUV<sub>1</sub>与Task<sub>3</sub>的距离较近.结合图13可知,AUV<sub>1</sub>的剩余资源占有率较高,故由式(13)可知,AUV<sub>1</sub>具有较低的响应阈值,由冲突解决方案,AUV<sub>1</sub>在对Task<sub>3</sub>的竞争中获胜,而AUV<sub>1</sub>转而执行 $p_{ij}$ 较大的Task<sub>5</sub>.

由图11可知,新出现的Task<sub>6</sub>由AUV<sub>1</sub>与AUV<sub>2</sub>协作完成,由图12可知,这是由于在 $t = 45\text{ s}$ 时AUV<sub>2</sub>执行完Task<sub>5</sub>后,AUV<sub>2</sub>依转移概率选择最愿意执行的任务为Task<sub>6</sub>.结合图13可知,由于新出现的Task<sub>6</sub>所需资源为4,大于AUV<sub>1</sub>、AUV<sub>2</sub>所剩余的资源但是小于两者的剩余资源之和,由冲突解决方案可知,两者均被分配执行Task<sub>6</sub>,从而实现了AUV间的合作,并于 $t = 53\text{ s}$ 时,协作完成Task<sub>6</sub>.

由图14、图15及图16可知,3种任务分配方法均于50s后才对Task<sub>7</sub>做出响应并于65s后完成全部的任务.相比图13可知,基于DACLD的任务分配方法中Task<sub>7</sub>在 $t = 43\text{ s}$ 就被AUV<sub>3</sub>响应,并于 $t = 53\text{ s}$ 时执行完成Task<sub>7</sub>,完成所有任务.这体现了所提出的DACLD任务分配方法能快速地响应新出现的任务,具有更高的任务执行效率.

## 4 结论

本文基于动态蚁群劳动分工模型研究了异构多AUV系统的动态任务分配问题,针对任务的位置随时间动态变化带来的冲突难题,提出了一种新的循环竞争方案,提高了任务的完成效率.仿真结果表明,所提方法具有良好的自组织性、鲁棒性以及快速性.下一步将引入通信拓扑图来对通信代价进行衡量,同时对多AUV系统引入更为复杂的约束(如通信范围、通信角度以及最小转弯半径等),以更好地满足实际系统的需求.

## 参考文献(References)

[1] 李建军. 基于群体智能的多AUV协同任务分配方法研究[D]. 哈尔滨: 哈尔滨工程大学, 2018.  
(Li J J. Research on multi-AUV collaborative task

allocation method based on swarm intelligence[D]. Harbin: Harbin Engineering University, 2018.)

- [2] 施龙. 基于蚁群算法的分布式多智能体任务分配研究[D]. 杭州: 浙江工商大学, 2011.  
(Shi L. Research on multi-agent task allocation based on ant colony optimization[D]. Hangzhou: Zhejiang Gongshang University, 2011.)
- [3] 包子阳. 智能优化算法及其Matlab实例[M]. 北京: 电子工业出版社, 2016: 99-104.  
(Bao Z Y. Intelligent optimization algorithm and its MATLAB example[M]. Beijing: Publishing House of Electronics industry, 2016: 99-104.)
- [4] 王娜. 基于改进蚁群算法的多AGV作业调度研究[D]. 西安: 陕西科技大学, 2017.  
(Wang N. Research on multi-AGV job scheduling based on improved ant colony algorithm[D]. Xi'an: Shaanxi University of Science & Technology, 2017.)
- [5] 禹晓蕾. 基于约束满足的改进蚁群算法在车间调度中的应用研究[D]. 武汉: 武汉理工大学, 2010.  
(Yu X L. Improved ant colony algorithm research based on constraint satisfaction in job-shop scheduling problem[D]. Wuhan: Wuhan University of Technology, 2010.)
- [6] Behrouzian A R B, Goswami D, Basten T, et al. Multi-constraint multi-processor resource allocation[C]. 2015 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS). Samos: IEEE, 2015: 338-346.
- [7] Gan Y, Qu F T, Sun F J, et al. Research on path planning for mobile robot based on ACO[C]. The 29th Chinese Control and Decision Conference (CCDC). Chongqing: IEEE, 2017: 6738-6743.
- [8] 秦新立, 宗群, 李晓瑜, 等. 基于改进蚁群算法的多机器人任务分配[J]. 空间控制技术与应用, 2018, 44(5): 55-59.  
(Qin X L, Zong Q, Li X Y, et al. Task allocation of multi-robot based on improved ant colony algorithm[J]. Aerospace Control and Application, 2018, 44(5): 55-59.)
- [9] 葛斯楠. 基于改进蚁群算法的多机器人任务分配[D]. 沈阳: 沈阳工业大学, 2016.  
(Ge S N. Multi-robot task allocation based on modified ant colony algorithm[D]. Shenyang: Shenyang University of Technology, 2016.)
- [10] 陈丹妮. 基于蚁群算法的无人机作战规划[D]. 广州: 华南理工大学, 2018.  
(Chen D N. Combat planning of unmanned aerial vehicles based on ant colony algorithm[D]. Guangzhou: South China University of Technology, 2018.)
- [11] 曾文权, 余爱民. 基于改进PSO算法的任务分配研究[J]. 计算机工程与应用, 2013, 49(13): 51-55.

- (Zeng W Q, Yu A M. Research on improved PSO algorithm based task allocation[J]. Computer Engineering and Applications, 2013, 49(13): 51-55.)
- [12] Roberge V, Tarbouchi M, Labonte G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning[J]. IEEE Transactions on Industrial Informatics, 2013, 9(1): 132-141.
- [13] 李立森, 梁志伟, 任彦达. 基于改进熟人模型的多智能体协作[J]. 南京邮电大学学报: 自然科学版, 2012, 32(3): 76-80.  
(Li L S, Liang Z W, Ren Y D. Multi-agent cooperation based on improved-acquaintance model[J]. Journal of Nanjing University of Posts and Telecommunications: Natural Science, 2012, 32(3): 76-80.)
- [14] Smith R G. The contract net protocol: High-level communication and control in a distributed problem solver[J]. IEEE Transactions on Computers, 1980, 29(12): 1104-1113.
- [15] Piccione M, Razin R. Coalition formation under power relations[J]. Theoretical Economics, 2009, 4(1): 1-15.
- [16] 陶雪丽, 李双群. 基于拍卖的动态任务分配算法[J]. 计算机与现代化, 2012(7): 51-54.  
(Tao X L, Li S Q. Dynamic task allocation algorithm based on auction[J]. Computer and Modernization, 2012(7): 51-54.)
- [17] Cheng Q, Yin D, Yang J, et al. An auction-based multiple constraints task allocation algorithm for multi-UAV system[C]. 2016 International Conference on Cybernetics, Robotics and Control (CRC). Hong Kong: IEEE, 2016: 1-5.
- [18] 朱大奇, 李欣, 颜明重. 多自治水下机器人多任务分配的自组织算法[J]. 控制与决策, 2012, 27(8): 1201-1205.  
(Zhu D Q, Li X, Yan M C. Task assignment algorithm of multi-AUV based on self-organizing map[J]. Control and Decision, 2012, 27(8): 1201-1205.)
- [19] Lee D H, Zaheer S A, Kim J H. A resource-oriented, decentralized auction algorithm for multirobot task allocation[J]. IEEE Transactions on Automation Science and Engineering, 2015, 12(4): 1469-1481.
- [20] 赵明. 多无人机系统的协同目标分配和航迹规划方法研究[D]. 哈尔滨: 哈尔滨工业大学, 2016.  
(Zhao M. Research on cooperative target assignment and path planning for multi-unmanned aircraft system[D]. Harbin: Harbin Institute of Technology, 2016.)
- [21] 肖人彬, 王英聪. 面向群体利益分配的蚁群劳动分工建模与仿真[J]. 管理科学学报, 2016, 19(10): 1-15.  
(Xiao R B, Wang Y C. Modeling and simulation of ant colony's labor division for interest allocation of social groups[J]. Journal of Management Sciences in China, 2016, 19(10): 1-15.)
- [22] 王英聪, 肖人彬. 基于蚁群劳动分工的空间分配方法求解带平衡约束的圆形装填问题[J]. 计算机集成制造系统, 2019, 25(2): 359-372.  
(Wang Y C, Xiao R B. Space allocation method based on ant colony's labor division for circular packing problem with equilibrium constraints[J]. Computer Integrated Manufacturing Systems, 2019, 25(2): 359-372.)
- [23] 据春华, 陈庭贵. 基于能力评价与利益驱动的扩展蚁群劳动分工模型及在动态任务分配中的应用[J]. 系统工程理论与实践, 2014, 34(1): 84-93.  
(Ju C H, Chen T G. Extended labor division model of ant colony based on ability-evaluation and interest-driven and its applications in dynamic task allocations[J]. Systems Engineering — Theory & Practice, 2014, 34(1): 84-93.)
- [24] 肖人彬, 王英聪. 群智能自组织劳动分工研究进展[J]. 信息与控制, 2019, 48(2): 129-139.  
(Xiao R B, Wang Y C. Research progress of self-organized labor division in swarm intelligence[J]. Information and Control, 2019, 48(2): 129-139.)
- [25] Wu H S, Li H, Xiao R B, et al. Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm[J]. Physica A: Statistical Mechanics and Its Applications, 2018, 491: 127-141.

### 作者简介

杨惠珍(1974—), 女, 副教授, 博士, 从事自主水下飞行器控制、多AUV系统等研究, E-mail: rainsun\_ly@npu.edu.cn;

王强(1997—), 男, 硕士生, 从事多AUV任务分配方法的研究, E-mail: 863025154@qq.com.

(责任编辑: 齐 霁)