

基于能量约束的多 AUV 任务规划算法

赵旭浩^{1,2,3,4}, 王轶群^{1,2,3,4*}, 刘健^{1,2,3}, 徐春晖^{1,2,3}

(1. 中国科学院沈阳自动化研究所, 沈阳 110016; 2. 机器人学国家重点实验室(中国科学院沈阳自动化研究所), 沈阳 110016;

3. 中国科学院机器人与智能制造创新研究院, 沈阳 110016; 4. 中国科学院大学, 北京 100049)

(*通信作者电子邮箱 wangyiqun@sia.cn)

摘要: 多水下自主航行器(Autonomous Underwater Vehicle, AUV)任务规划是影响集群智能水平的关键技术, 针对现有任务规划模型只考虑同构 AUV 集群和单潜次任务规划的问题, 本文提出了适用于 AUV 异构集群的多潜次任务规划模型。首先该模型考虑了 AUV 的能量约束、AUV 多次往返母船充电的工程代价、异构集群个体间的效能差异、任务多样性等关键因素; 然后为提高问题模型的求解效率, 提出了一种基于离散粒子群的优化算法, 该算法引入用于描述粒子速度、位置的矩阵编码和用于评估粒子质量的任务损耗模型, 改进粒子更新过程, 实现了高效的目标寻优。仿真实验表明, 该算法不但解决了异构 AUV 集群的多潜次任务规划问题, 而且与采用遗传算法的任务规划模型相比较, 任务损耗降低了 11%。

关键词: 自主水下航行器(Autonomous Underwater Vehicle, AUV); 多 AUV 集群; 任务规划; 离散粒子群算法; 多样性任务

中图分类号: TP242.6

文献标志码: A

Task planning algorithm of multi-AUV based Energy constraint

Zhao Xuhao^{1,2,3,4}, Wang Yiqun^{1,2,3,4*}, Liu Jian^{1,2,3}, Xu Chunhui^{1,2,3}

(1. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China;

2. The State Key Laboratory of Robotics(Shenyang Institute of Automation, Chinese Academy of Science), Shenyang 110016, China;

3. Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China;

4. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Autonomous Underwater Vehicle (AUV) mission planning is the key technology that affects the level of cluster intelligence. For the existing mission planning model, only the problem of homogeneous AUV cluster and single dive mission planning is considered. So a multi-dive mission planning model for AUV heterogeneous clusters is proposed. Firstly the model considered the energy constraints of AUV, the engineering cost of AUV multiple round-trip mother ship charging, the efficiency difference between heterogeneous cluster individuals, and the diversity of tasks. Then in order to improve the efficiency of solving the problem model, an optimization algorithm based on discrete particle swarm optimization was proposed. The algorithm introduced matrix coding for describing particle velocity and position, and the task loss model for evaluating particle quality, and improves the particle update process, achieving efficient target optimization. Simulation experiments show that the algorithm not only solves the multi-dive mission planning problem of heterogeneous AUV clusters, but also reduces the task loss by 11% compared with the mission planning model using genetic algorithm.

Keywords: AUV; multi-AUV cluster; task planning; discrete particle swarm optimization; multiple task

0 引言

自主水下航行器(Autonomous Underwater Vehicle, AUV) 是进行海洋探索和海洋科学研究的有效工具。AUV 以其智能化、灵活性、快速性成为各国争相发展的水下利器^{[1][2]}。随着

收稿日期: 2019-04-03; 修回日期: 2019-04-30; 录用日期: 2019-05-05。

基金项目: 国家重点研发计划(2017YFC0306800)

作者简介: 赵旭浩(1994—), 男, 山东济宁人, 硕士研究生, 要研究方向: 多 AUV 任务规划; 王轶群(1985—), 男, 辽宁沈阳人, 副研究员, 硕士, 主要研究方向: AUV 导航及路径规划; 刘健(1962—), 男, 辽宁沈阳人, 研究员, 硕士, 主要研究方向: 水下机器人导航与控制; 徐春晖(1982—), 男, 辽宁沈阳人, 副研究员, 硕士, 主要研究方向: 水下机器人控制

近年来机器人技术的快速发展,多 AUV 系统受到广泛关注。和单体 AUV 相比,多 AUV 系统的作业范围更广,作业效率更高,而且能够完成单体 AUV 不能完成的复杂海洋探测任务,在海洋矿产资源调查、海洋现象观测、水下目标搜索、海洋信息网络组建^[3]等诸多方面有广阔的应用前景。而多 AUV 任务规划是多 AUV 系统协作的一个基本问题。多 AUV 系统的工作方式如图 1 所示,多个不同类型的 AUV 在水下通信网络和母船的支持下协同完成探测任务,由于 AUV 的布放和回收都比较困难,提高能源利用率对 AUV 意义重大,合理的任务规划能够提高能源利用率、提高系统的工作效率。

目前多 AUV 任务规划的研究已经取得了很大的进展^[4]。针对多 AUV 的任务规划问题,文献[5]建立了以最大收益为目标的任務规划模型,然后用混沌粒子群算法求解,但是所建立的模型只考虑了执行任务的收益和时间,没有考虑 AUV 和任务之间的距离信息。文献[6]根据任务的位置和能耗建立了多 AUV 任务规划模型,并且用蚁群算法进行求解,但是模型中缺乏传感器匹配机制和任务均衡策略。文献[7]构建了以每个 AUV 的能量耗费与能耗均衡为约束条件的水下三维空间下的多旅行商问题模型,并且设计了考虑巡航总路径及访问目标数的适应度函数,但是这种方法没有考虑任务的规模。文献[8]用自组织神经网络求解 AUV 的任务规划和路径规划问题,但是没有考虑 AUV 的能量约束和传感器约束。文献[9]以使用最少的 AUV 为目标,建立了只考虑传感器匹配和时间限制的整数规划模型。文献[10]首先建立了每个 AUV 和每个任务之间的匹配矩阵,然后以多目标元启发式方法求解满足约束的最优解,实现了 AUV 和任务之间的合理匹配。文献[11]用基于市场的分布式方法解决 AUV 的任务分配问题,这种方法能够合理的处理各类约束,但是对通信的稳定性要求比较高。文献[12]研究了 AUV 和无人机的任务协同,并且提出了考虑时间窗口的任务分配模型,取得了较好的效果。

上述文献在一定程度上能够解决多 AUV 的任务规划,但是所建立的模型不能够反映多 AUV 集群的实际工作过程。首先,作者都假设在一次任务规划过程中所有的 AUV 有能力执行完所有的任务,或者只分配 AUV 能够承担的任务,或者直接不考虑 AUV 的数量。但实际情况是,AUV 一次下潜所能探测的区域远远小于所要探测的海洋区域,AUV 不可能一次作业下潜就能够执行完所有任务,往往需要多次回母船补充能源,多次下潜才能够完成任务。再者,AUV 执行任务主要依靠 AUV 自身所携带的传感器和探测设备,而不同的任务也是根据需要的传感器和设备来区分的,因此,考虑 AUV 的异构和任务的多样性至关重要。

因此,本文在上述文献的基础上建立了新的多 AUV 任务规划模型,考虑了 AUV 回母船充电的过程,对 AUV 在各个任务点的能量变化进行了精细建模,在满足传感器约束的条件下能够为集群内的每条 AUV 规划多次作业下潜。新模型以 AUV

的下潜次数和任务时长为优化目标,能够在下潜次数和任务时长之间达到平衡。改进了一种新型的离散粒子群算法对异构多 AUV 任务规划模型求解,设计了矩阵编码的方式表示粒子位置和粒子速度,将约束的处理加入到了粒子的编码和更新过程中。

1 问题描述

多个 AUV 组成编队协同执行海洋探测任务时,海洋探测任务需求不同,需要不同的 AUV 完成不同的任务,因此本文建立了 AUV 传感器向量和任务传感器向量,用来完成 AUV 和任务之间的匹配。另外,有时候任务的工作量较大,所有的 AUV 一次下潜并不能完成所有任务,必须将 AUV 回母船或者回水下基站充电的过程在任务规划模型中体现出来。针对上述问题,本文在模型中用多个虚拟充电点来模拟 AUV 多次回母船充电的过程,AUV 经过一个虚拟充电时,表示 AUV 返回了母船一次。

执行海洋探测任务时,AUV 的操控者和海洋科学家所关心的目标主要有两点:第一是执行任务的时间,第二是 AUV 的下潜次数。在多 AUV 的场景下,这两个目标是存在矛盾的,多个小型的 AUV 同时下潜能够执行完的任务对于大型 AUV 来说一次下潜就能够完成,如何平衡这两种目标是关键。

问题的假设条件如下:AUV 的电量有限,只存在一个母船,AUV 都从母船出发,母船的位置固定不变,AUV 从母船出发时都是满电状态,AUV 都能够顺利的执行完任务。

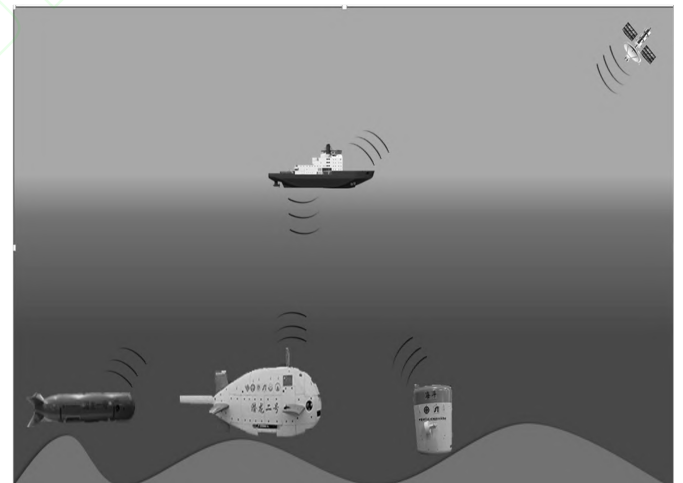


图 1 多 AUV 工作场景

Fig 1 Scene of multi-AUV working

2 任务分配建模

多 AUV 任务分配模型描述如下:多样性任务点的集合为 T ,异构 AUV 的集合为 R ,一条母船。AUV 要经过多次回母船充电的过程然后执行完这些任务。模型参数如表一所示

表 1 模型参数

Tab 1 Model parameter

参数	描述
R	AUV 集合
T	任务点集合
O	所有 AUV 的起点, 即为母船
O'	所有 AUV 的终点, 可以为母船, 也可能是其他点
M	虚拟充电点集合, 实际位置为母船
V	$T \cup O \cup M$
V'	$T \cup M$
V''	$T \cup O' \cup M$
C	AUV 的充电成本
d_{ij}	任务点 i 和任务点 j 之间的距离 (Km)
E_{jr}^1	AUV r 到达 j 点时的剩余电量($kW \cdot h$)
E_{jr}^2	AUV r 离开 j 点时的剩余电量($kW \cdot h$)
W_i	AUV 执行任务 i 所需的电量($kW \cdot h$)
λ^r	AUV r 行驶单位距离所需要的电量($kW \cdot h$)
E_{max}^r	AUV r 充满电时的电量($kW \cdot h$)
$sensor^r$	AUV r 的传感器向量
$speed^r$	AUV r 的速度 (m/s)
$sensor^j$	任务 j 所需要的传感器向量
x_{ij}^r	0-1 决策变量, AUV r 是否从任务点 i 行驶到任务点 j

2.1 建立模型

2.1.1 目标函数

目标函数(1)中的 Z_1 是所有 AUV 的充电成本, 再加上 AUV 在执行任务的过程中行驶的总路程。从目前来看, 首先 AUV 的回收和布放都是带有危险性的作业, 所以应尽量减少 AUV 回母船充电的次数。其次, 即使将来采用了水下基站的方式进行充电, AUV 和基站的对接过程依旧是耗时、耗力的工作, 最后, 现阶段 AUV 所使用特殊电池的使用寿命和陆地电池还有差距。因此, 应该保证以最少的充电次数就能够完成任务。同时, 采用多 AUV 编队的目的就是要减少任务的时长, 因此 AUV 执行任务的时长也应该在目标函数中有所体现, 多 AUV 执行任务是一个并行系统, 其任务时长应该用运行时间最长的那台 AUV 的运行时间来表示。综上, 可以得到 Z_1 和 Z_2 两个目标函数, Z_1 是充电成本和 AUV 在任务点之间的行驶距离相加, Z_2 是所有 AUV 中最大的那个任务时长, 并且使用 ω_1 和 ω_2 两个参数调整他们的权重, 可以根据重视程度来调整, 这样既能够保证机器人的能源得到最大的利用, 又能够优化机器人的运行时间。

$$\text{Min } Z = \omega_1 Z_1 + \omega_2 Z_2 \quad (1)$$

$$Z_1 = \sum_{r \in R} \sum_{i \in V'} \sum_{m \in M} Cx_{im}^r + \sum_{r \in R} \sum_{i \in V'} \sum_{j \in V''} d_{ij} x_{ij}^r \quad (2)$$

$$Z_2 = \max_{r \in R} \sum_{i \in V'} \sum_{j \in V''} x_{ij}^r (d_{ij} + W_i) / speed^r \quad (3)$$

$$\omega_1 + \omega_2 = 1; \quad (4)$$

2.1.2 约束条件

(1) 每个任务只能被执行一次。

$$\sum_{r \in R} \sum_{i \in V'} x_{ij}^r = 1 \quad \forall j \in T \quad (5)$$

(2) 虚拟充电点有可能被使用, 也有可能不被使用。

$$\sum_{r \in R} \sum_{i \in V'} x_{im}^r \leq 1 \quad \forall m \in M \quad (6)$$

(3) 所有 AUV 都必须从起点出发

$$\sum_{j \in V'} x_{Oj}^r = 1 \quad \forall r \in R \quad (7)$$

(4) 所有 AUV 都不能再回到起点

$$\sum_{j \in V'} x_{jO}^r = 0 \quad \forall r \in R \quad (8)$$

(5) 所有 AUV 最终都必须回到终点

$$\sum_{j \in V'} x_{jO'}^r = 1 \quad \forall r \in R \quad (9)$$

(6) 所有 AUV 都不能从终点出发

$$\sum_{j \in V'} x_{O'j}^r = 0 \quad \forall r \in R \quad (10)$$

(7) 必须遵循流量守恒原则, AUV 不能在任务点之间跳跃。

$$\sum_{i \in V'} x_{ij}^r = \sum_{i \in V'} x_{ji}^r \quad \forall j \in V', \forall r \in R \quad (11)$$

(8) 能量约束, AUV 在一个任务点的剩余能量必须保证能到达下一个任务点, E_{max}^r 是松弛条件, 保证没有连接关系的任务点也能满足约束。

$$E_{jr}^1 \leq E_{ir}^2 - d_{ij} \lambda^r x_{ij}^r + E_{max}^r (1 - x_{ij}^r) \quad \forall i, j \in V', r \in R \quad (12)$$

(9) AUV 在离开起点时是满电量状态。

$$E_{Or}^2 = E_{max}^r \quad \forall r \in R \quad (13)$$

(10) AUV 在离开虚拟充电点时是满电量状态

$$E_{ir}^2 = E_{max}^r \sum_{j \in V'} x_{ij}^r \quad \forall r \in R, i \in M \quad (14)$$

(11) 离开任务点的能量为到达任务点的能量减去执行任务消耗的能量

$$E_{ir}^2 = E_{ir}^1 - W_i \sum_{j \in V'} x_{ij}^r \quad \forall r \in R, i \in T \quad (15)$$

(12) 要保证在每个任务点执行完任务都能回到终点

$$E_{ir}^2 \geq \lambda^r d_{iO'} \quad \forall r \in R, i \in T \quad (16)$$

(13) 传感器匹配约束, 对于不同的 AUV 来说, 最大的区别就是传感器, 不同的任务之间的区别也是是否需要某种传感器, 因此本文引入了 AUV 传感器向量和任务传感器向量。 $sensor^r$ 和 $sensor^j$ 都是内容为 0 和 1 的向量, 维度相同, $sensor^r$ 的每一位都要大于 $sensor^j$ 。

$$\sum_{r \in R} \sum_{i \in T} x_{ij}^r sensor^r \geq sensor^j \quad \forall r \in R, j \in T \quad (17)$$

3 改进基于集合的粒子群算法

基于集合的粒子群算法 (Set-based Particle Swarm Optimization, S-PSO) [13] 是一种特殊的离散粒子群算法, 相比于常见的离散粒子群算法 [14], 他的特点在于用集合的概念来表示粒子的位置和速度, 粒子的更新过程用集合运算来代替, 最大的优点在于每次粒子更新都能够得到满足约束的解。S-PSO 算法常常用来解决离散领域的组合优化问题 [15]。S-PSO 算法的搜索空间用一个全集 S 表示, 每个维度代表 S 的一个子集 S^i ($i=1,2,\dots,D$), 即

$$S = S^1 \cup S^2 \cup \dots \cup S^D \quad (18)$$

每个粒子位置 X 都是全集 S 的一个子集, $X \subseteq S$, X 可以表示为:

$$X = X^1 \cup X^2 \cup \dots \cup X^D, X^i \subseteq S^i (i=1,2,\dots,D) \quad (19)$$

如果 X 能够满足所有约束, X 就是一个可行解。在基于集合的粒子群算法中, 速度使用一个可能性集合来表示 [13]:

$$V = \{e / p(e) | e \in E\} \quad (20)$$

3.1 粒子编码

按照上述思想, 设计多 AUV 任务分配求解问题的编码方式如下

$$X_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad VEL_i = \begin{bmatrix} 0 & 0.5 & 0.8 \\ 0.8 & 0 & 0.63 \\ 0.39 & 0.22 & 0 \end{bmatrix}$$

X_i 是一个 0,1 矩阵, 表示粒子的位置编码, VEL_i 的每一个元素值都在 [0,1] 之间, 表示粒子的速度编码, 其中矩阵的维度也是粒子的维度, 在本文的问题中, 粒子的维度就代表任务的数量。

3.2 速度更新公式

S-PSO 算法主要利用综合学习公式, 这样能避免粒子早熟

$$VEL_{id} = \omega \times VEL_{id} + c \times r_d \times (P_{id} - X_{id}) \quad (21)$$

但是 S-PSO 算法重新定义了公式(21)的各个运算符, 将上述的运算符操作改为了概率之间的运算和集合之间的运算。公式(22)-(25)中的 i 代表粒子编号, d 代表维度, 在本文

的问题中, 第 d 维元素就是所有和任务 d 有连接关系的任务。 $\langle u,v \rangle$ 就代表着这种连接关系, $p(u,v)$ 表示这种连接关系对应的概率。

$$c \times VEL_{id} = \{ \langle u,v \rangle / p'(u,v) | \langle u,v \rangle \in A^d \}$$

$$p'(u,v) = \begin{cases} 1, & \text{if } c \times p(u,v) > 1 \\ c \times p(u,v), & \text{otherwise} \end{cases} \quad (22)$$

$$VEL_{id} + VEL_{jd} = \{ \langle u,v \rangle / \max(p_i(u,v), p_j(u,v)) | \langle u,v \rangle \in A^d \} \quad (23)$$

$$X_{id} - X_{jd} = U^d = \{ \langle u,v \rangle | \langle u,v \rangle \in X_{id}, \langle u,v \rangle \notin X_{jd} \} \quad (24)$$

$$c \times U^d = \{ \langle u,v \rangle / p'(u,v) | \langle u,v \rangle \in A^d \}$$

$$p'(u,v) = \begin{cases} 1 & \text{if } \langle u,v \rangle \in U^d \text{ and } c \geq 1 \\ c & \text{if } \langle u,v \rangle \in U^d \text{ and } 0 \leq c \leq 1 \\ 0 & \text{if } \langle u,v \rangle \notin U^d \end{cases} \quad (25)$$

假设当前粒子位置如 X_i 所示, 粒子速度如 VEL_i 所示, 而目前的最优粒子为

$$P_i = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

如果要计算维度 2 的粒子速度更新结果, 那么

$$X_{i2} = \{ \langle 1,2 \rangle, \langle 2,3 \rangle \}$$

$$P_{i2} = \{ \langle 3,2 \rangle, \langle 2,1 \rangle \}$$

$$U^2 = \{ \langle 3,2 \rangle, \langle 2,1 \rangle \}$$

假设 $c = 0.5$

$$c \times U^d = \{ \langle 3,2 \rangle / 0.5, \langle 2,1 \rangle / 0.5 \}$$

$$c \times VEL_{i2} = \{ \langle 1,2 \rangle / 0.25, \langle 2,1 \rangle / 0.4, \langle 2,3 \rangle / 0.315, \langle 3,2 \rangle / 0.11 \}$$

那么

$$c \times VEL_{i2} + c \times U^d = \{ \langle 1,2 \rangle / 0.25, \langle 2,1 \rangle / 0.5, \langle 2,3 \rangle / 0.315, \langle 3,2 \rangle / 0.5 \}$$

就得到了更新之后的速度值, 同时可以看到有关维度 2 的概率值得到了更新, 并且概率值得到了提升, 表示从最优粒子中学到了知识。

3.3 位置更新

原始的粒子群算法的位置更新公式为:

$$X_{i-new} = X_i + V_i \quad (26)$$

而 S-PSO 算法将公式(26)中的加法运算符改成了如算法 1 所示的集合运算操作。在文献 [15] 中, 作者将 S-PSO 应用到了多车辆路径问题上, 但是本文的问题和多车辆路径问题

有一定差异。因为多车辆问题不考虑车辆的异构性，不考虑车辆的数量，而 AUV 的异构性是本文的主要关注点，因此，本文在原有的粒子位置更新算法中加入了随机选择 AUV 的过程和传感器匹配的过程，改进后的粒子更新过程如算法 1 所示：

算法 1. 粒子位置更新过程

输入： 速度，原有粒子

输出： 新粒子

P_v :速度编码矩阵; P_x :当前位置矩阵; P_E :所有位置都为 1 的矩阵。 P'_x :保存新粒子, i :当前已分配的任务; j :下一个任务

- 1) Convert(P_v);//将小于 0.5 的概率值置 0
- 2) While(还有任务没有被分配)
- 3) If 存在 $P_v(i,j)>0$ 且满足约束
- 4) 以从满足条件的集合中以距离最短的标准从 P_v 中选择 j
- 5) end if
- 6) If 在 P_v 中没有找到满足约束的任务
- 7) 从 P_x 中寻找满足条件的 j
- 8) end if
- 9) If 在 P_x, P_v 中都没有找到
- 10) 从 P_E 中寻找满足约束的任务
- 11) end if
- 12) If 从上述三个集合中都没找到合适的任务
- 13) 说明当前机器人不合适，以随机选择的方式更换 AUV 并且记录当前的 i 和 AUV,让当前 AUV 回到终点,即 $P'_x(i,1)=1$
- 14) end if
- 15) $i=j$;将 j 加入到已分配任务序列中
- 16) endwhile
- 17) return P'_x

上述算法模拟的过程是：按照 AUV 的编号随机选择一个 AUV，给 AUV 分配满足约束的任务，直到 AUV 达到电池容量约束或者再没有合适的任务可以执行，然后让该 AUV 回到母船充电，再次随机选择新的 AUV，直到所有任务分配完毕。该算法在更新粒子的过程中能够保证上述模型中的约束都能够得到满足。因此，一次循环之后得到的一定是一个可行解。

3.4 粒子评估

粒子评估采用上文中建立的多目标函数，目标函数的值代表着执行任务总的损耗，这个损耗既有电池能量上的损耗也包括时间上的损耗，任务损耗的值越小，表明粒子的适应度越好。

3.5 粒子解码

假设有一个起点（终点和起点相同）和 3 个待分配任务的问题，那么这个粒子位置的一种可能性如下。

$$P = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

从这个矩阵中可以得到 $P(1,2)=1, P(2,1)=1, P(1,4)=1, P(4,3)=1, P(3,1)=1$ 。那么这三个任务就是依靠 AUV 的两次下潜来完成的，执行任务的顺序分别是 1-2-1 和 1-4-3-1。虽然得到了两次下潜执行任务的顺序，但是不知道由那个 AUV 执行的任务。好在在粒子更新的过程中记录了每个粒子在更新 AUV 时，前一个 AUV 执行的最后一个任务编号以及 AUV 型号。这样就能到解码得到每个 AUV 所要执行的任务。

3.6 S-PSO 算法总流程

S-PSO 算法的工作流程和原始粒子群算法的流程没有区别，流程如下：

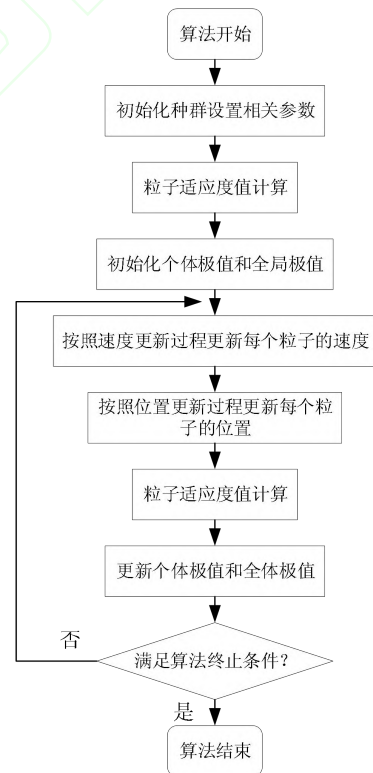


图 2 S-PSO 算法流程

Fig 2 Algorithm of S-PSO

4 仿真分析

本文的实验环境为 I5-6300HQ, 8GB RAM, 实验软件为 MATLAB R2016a。仿真实验的基本参数为：粒子种群数量

为 20, 惯性参数为 2, 加速度系数为 0.7, 目标函数中的 $\omega_1 = 0.5$, $\omega_2 = 0.5$, 总的任务数量为 30 个。表 2 展示的是 30 个任务的信息, 任务信息中分别有四个数字, 这四个数字的含义分别是 (任务横坐标/Km, 任务纵坐标/Km, 任务代价/ $kW \cdot h$, 任务所需传感器向量)。编号 1 属于母船, 母船的位置是 (35, 35)。

表 2 任务信息
Tab 2 Information of task

编号	任务信息	编号	任务信息
2	(41,49,10,1100)	17	(10,20,19,1110)
3	(35,17,7,1100)	18	(5,30,2,1110)
4	(55,45,13,1100)	19	(20,40,12,1110)
5	(55,20,19,1100)	20	(15,60,17,1110)
6	(15,30,26,1100)	21	(45,65,9,1110)
7	(25,30,3,1100)	22	(45,20,11,1010)
8	(20,50,5,1110)	23	(45,10,18,1010)
9	(10,43,9,1110)	24	(55,5,29,1010)
10	(55,60,16,1110)	25	(65,35,3,1010)
11	(30,60,16,1110)	26	(65,20,6,1010)
12	(20,65,12,1110)	27	(45,30,17,1010)
13	(50,35,19,1110)	28	(35,40,16,1010)
14	(30,25,23,1110)	29	(41,37,16,1010)
15	(15,10,20,1110)	30	(64,42,9,1010)
16	(30,5,8,1110)	31	(40,60,21,1010)

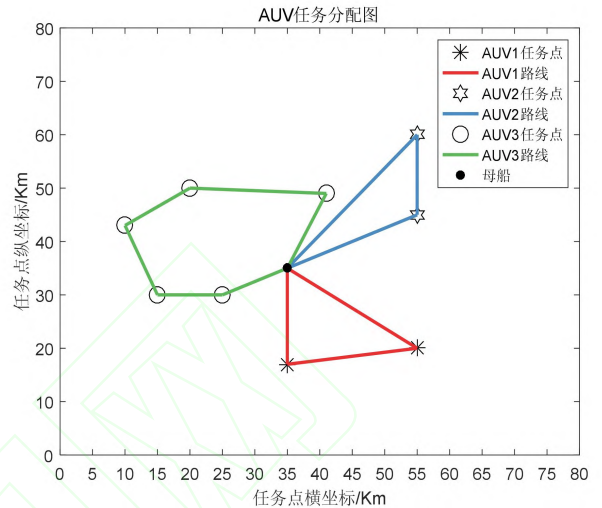
仿真实验设定有四条 AUV, 每条 AUV 的参数设置如表 3 所示, 其中电池表示 AUV 的电池容量, 可以看到 4 条 AUV 所拥有的传感器都不一样。其中 AUV2 和 AUV3 能够执行执行所有的任务, 而 AUV1 和 AUV4 只能够执行部分任务。这样的参数设置能够最大程度的保证 AUV 之间的异构性, 也能够有效的测试算法的有效性。AUV 行驶单位距离所用的电量都设为 $\lambda' = 1$ 。充电成本设为 100。

表 3 AUV 列表
Tab 3 List of AUV

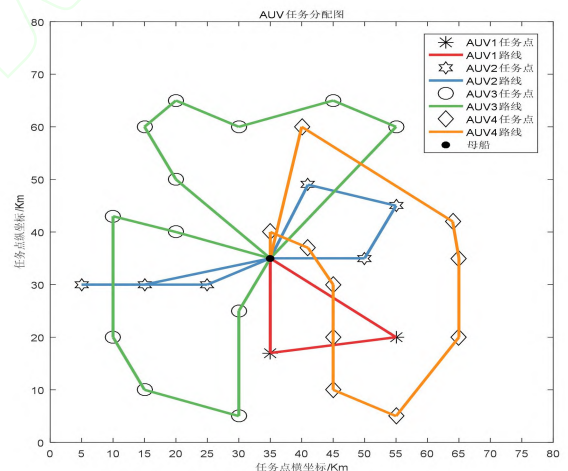
编号	速度 /m/s	电池 / $kW \cdot h$	S1	S2	S3	S4
AUV1	1.5	100	1	1	0	0
AUV2	1	100	1	1	1	0
AUV3	1	200	1	1	1	0
AUV4	1	300	1	0	1	0

首先, 对待分配的任务量小于多 AUV 系统一次下潜的工作量的情况进行仿真。从上述 30 个任务中抽取前 10 个任务, 分配给 4 个 AUV。粒子群算法迭代 200 次, 结果如图 3 (a) 所示, 可以看到 AUV1-AUV3 分配得到了任务, 而 AUV4 没有分配到任务。因为前 10 个任务都需要传感器 S2, AUV4 没有这个传感器, 所以 AUV4 没有分配到任务; 再者, 能够

执行任务的 AUV 都被分配到了任务, 没有单个 AUV 负载过重的情况, 而且 AUV3 的电池容量最大, 它承担的任务也最多, 带来了最少的下潜次数, 任务分配的结果完全符合目标函数的设定。



(a) 10 个任务分配给 4 个 AUV 的结果



(b) 30 个任务分配给 4 个 AUV 的结果

图 3 任务规划结果

Fig 3 Result of mission planning

图 3(b) 和表 4 展示的是把 30 个任务分配给 4 个 AUV 的结果, 粒子群算法迭代 1000 次。30 个任务的工作量明显超过了多 AUV 系统一次下潜的工作量, 因此 AUV2 和 AUV3 都下潜了两次, 而 AUV4 单次下潜执行的任务最多, 这种情况非常符合所设计的目标函数。首先, 目标函数中的 Z_1 是最小化下潜次数, AUV4 的电池容量最大, 它基本上执行了它所能执行的所有任务, AUV2 和 AUV3 的能力最强, 能执行所有任务, 所以这两个 AUV 执行的任务也多, 符合目标函数

数的设定。如果只看 Z_1 ，最优的情况应该是 AUV3 和 AUV4 执行完所有的任务，只有这样分配任务才能得到最少的下潜次数，但是这样会造成任务执行时间过长，失去了部署多 AUV 执行任务的意义。所以，本文在目标函数中加入了 Z_2 ，同时实验结果也体现了目标函数一和目标函数二之间的均衡，AUV1 号和 AUV2 分担了一部分任务，将任务时长降低，如果在 AUV3 执行任务期间多次派遣 AUV1 执行任务，确实可以进一步将任务执行时间减少，但是这样又会造成下潜次数增加，给保障工作带来负担。

通过上述的仿真实验可以看出，相对于其他算法只能实现 AUV 的单次下潜任务规划，本文设计的方法再完成传感器匹配的前提下能够实现多 AUV 的多次作业下潜任务规划，并且能够平衡下潜次数和任务的执行时间。

表 4 任务规划结果

Tab 4 Result of mission planning

AUV 编号	任务执行次序
AUV1	1-5-3-1
AUV2	1-6-18-7-1-2-4-13-1
AUV3	1-10-21-11-12-20-8-1-14-16-15-17-9-19-1
AUV4	1-28-29-27-22-23-24-26-25-30-31-1

为了说明基于 S-PSO 算法的有效性，本文也改进了传统的遗传算法(Genetic Algorithm,GA)用来求解本文提出的模型，采用正整数编码的方式，在对染色体适应度评估的过程中对染色体任务序列加入虚拟充电点，采用和粒子群算法同样的多目标函数，并且对不符合约束的个体添加惩罚函数。遗传算法的参数如下：种群大小为 100，交叉概率为 0.9，变异概率为 0.05，代沟为 0.9。在任务数量为 30 个，迭代次数为 300 次的情况下，对比两种算法所求出的任务损耗最优解，结果如图 5 所示。可以看到遗传算法在迭代初期求出的任务损耗很高，这是因为遗传算法处理约束的能力不强，求出的解有很多不满足约束，所以初期的任务损耗很高。而 S-PSO 算法则恰恰相反，在迭代初期求出的任务损耗就比遗传算法低，这是因为基于集合的粒子群算法求出的每一个解都是满足约束的解，惩罚函数根本就没有发挥作用。同时，在图中可以看到，S-PSO 算法在第 60 次迭代的时候就趋于稳定，而遗传算法在 110 代的时候才趋于稳定；最后 S-PSO 算法所求出的任务损耗只有 1160，而遗传算法所求出的损耗为 1310，S-PSO 算法比遗传算法降低了 11%。因此，相比于遗传算法，S-PSO 算法收敛速度更快，求出的最优解更好。

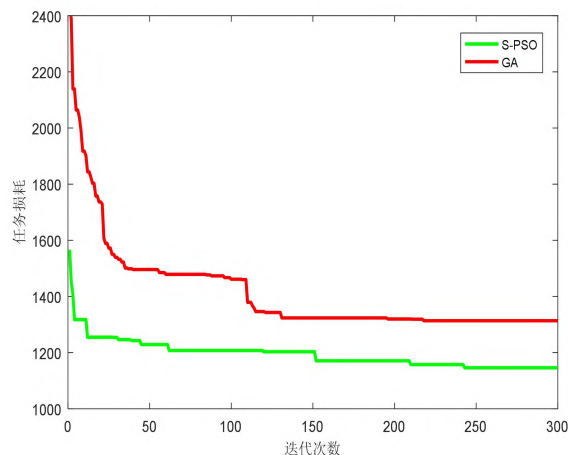


图 4 遗传算法和 S-PSO 算法对比

Fig 4 Comparison of GA and S-PSO

5 结语

本文从实际应用角度出发，对多 AUV 集群的任务规划问题进行了深入的研究，提出了多 AUV 的多次作业下潜任务规划概念，并且对提出的问题进行了数学建模，在模型中加入了 AUV 回母船充电这个过程。同时为模型的求解改进了 S-PSO 算法。仿真结果表明这个模型具有很好的通用性，能够将任意数量的任务分配给合适的 AUV，并且能够合理的平衡 AUV 的下潜次数和任务执行的总时长。同时，仿真结果也表明 S-PSO 算法在多约束问题上所得到的结果优于其他算法。后续工作将考虑有关多 AUV 集群的动态任务分配问题。

参考文献

- [1] 朱心科, 金翔龙, 陶春辉, 等. 海洋探测技术与装备发展探讨[J]. 机器人, 2013, 35(3): 376-384
- [2] 李硕, 刘健, 徐会希, 赵宏宇, 王轶群. 我国深海自主水下机器人的研究现状[J]. 中国科学: 信息科学, 2018, 48(09): 1152-1164
- [3] 许真珍, 封锡盛. 多 UUV 协作系统的研究现状与发展[J]. 机器人, 2007(02): 186-192.
- [4] THOMPSON F, GUIHEN D. Review of mission planning for autonomous marine vehicle fleets[J]. J Field Robotics. 2018:1-22.
- [5] 李建军, 张汝波, 杨玉. 基于混沌 QPSO 算法的多 AUVs 任务分配[J]. 华中科技大学学报(自然科学版), 2015, 43(S1): 424-427.
- [6] 刘瑞轩, 张永林. 基于改进蚁群算法的多自主式水下机器人任务分配[J]. 中国舰船研究, 2018, 13(06): 107-112.
- [7] 张美燕, 蔡文郁. 基于多 AUV 间任务协作的水下多目标探测路径规划[J]. 传感技术学报, 2018, 31(07): 1101-1107.
- [8] 朱大奇, 曹翔. 多个水下机器人动态任务分配和路径规划的信度自组织算法[J]. 控制理论与应用, 2015, 32(06): 762-769.
- [9] J. WILDE, D. DIBIASO AND M. NERVEGNA. Team Planning for Unmanned Vehicles in the Risk-Aware Mixed-Initiative Dynamic Replanning System[C]//OCEANS 2007. Vancouver, BC. IEEE, 2007: 1-8.

- [10] LANDA-TORRES I, MANJARRES D, BILBAO S, et al. Underwater robot task planning using multi-objective meta-heuristics[J]. *Sensors*, 2017, 17(4): 762.
- [11] DENG Y, BEAUJEAN P P J, AN E, et al. Task allocation and path planning for collaborative autonomous underwater vehicles operating through an underwater acoustic network[J]. *Journal of Robotics*, 2013.
- [12] WILDE J, DIBIASO D, NERVEGNA M. Team planning for unmanned vehicles in the risk-aware mixed-initiative dynamic replanning system[C]//OCEANS 2007. IEEE, 2007: 1-8.
- [13] CHEN W N, ZHANG J, CHUNG H S H, et al. A novel set-based particle swarm optimization method for discrete optimization problems[J]. *IEEE Transactions on evolutionary computation*, 2010, 14(2): 278-300.
- [14] 程毕芸,鲁海燕,徐向平,沈莞蕾.求解旅行商问题的改进局部搜索混沌离散粒子群优化算法[J].*计算机应用*,2016,36(01):138-142+149.
- [15] GONG Y J, ZHANG J, LIU O, et al. Optimizing the vehicle routing problem with time windows: a discrete particle swarm optimization approach[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2012, 42(2): 254-267.

This work is supported by

National Key R&D Program of China (2017YFC0306800)

ZHAO Xuhao, born in 1994, M. S. candidate. His research interests include task planning of multi-AUV

WANG Yiqun, born in 1985, associate professor. His research interests include navigation methods and path planning algorithm of deep-sea AUVs.

LIU Jian born in 1962, professor. His research interests include navigation methods and control algorithm of deep-sea AUVs.

XU Chunhui born in 1982, associate professor. His research interests include control algorithm of deep-sea AUVs.

1.