

DOI: 10.19650/j.cnki.cjsi.J2209583

改进蚁群算法解决 UUV 集群任务规划问题^{*}

王宏健, 鄂 鑫, 张 凯, 易冬波, 牛 帅

(哈尔滨工程大学 哈尔滨 150001)

摘 要:针对水下无人航行器(UUV)集群在有限续航力和负载约束条件下求解广泛且稀疏分布区域勘察任务规划问题时常规算法存在收敛性差、解质量不高的不足,提出了一种改进的蚁群优化算法。首先,通过分析个体 UUV 平台能力和集群任务的约束条件,建立 UUV 集群任务规划的约束模型和优化模型;其次,基于任务点间距离与平均距离之差设计初始信息素浓度的非均等分配方法,提出优化模型的最佳与最差阈值对蚂蚁进行分类并对应完成信息素更新,在状态转移规则中创新加入可随迭代进程动态改变的“引力系数”来增加算法前中期次优节点被选中的概率;再次,设置对照统计实验完成算法优化项的有效性分析,依据最优解出现的次数和平均收敛值优化算法参数;最后,以经典文献案例仿真,对比分析基本蚁群算法、精英蚁群算法与提出算法,相较于前两种算法,算法在 50 次统计实验中找到近似最优解的百分比分别提升 78% 和 66%,平均在第 40 代实现收敛,表明出很好的全局寻优能力和收敛性能。通过设计具有一定规模的 UUV 集群任务规划典型案例,验证了算法求解 UUV 集群广泛且稀疏分布区域任务规划问题的快速性和有效性。

关键词: 水下无人航行器集群;任务规划;蚁群算法;约束模型;算法参数优化

中图分类号: TH701 **文献标识码:** A **国家标准学科分类代码:** 570.50

Improved ant colony algorithm to solve UUV cluster task planning problem

Wang Hongjian, E Xin, Zhang Kai, Yi Dongbo, Niu Shuai

(Harbin Engineering University, Harbin 150001, China)

Abstract: When conventional algorithms are used to solve the survey task planning problem of unmanned underwater vehicle (UUV) swarm under limited endurance and load constraints in a wide and sparsely distributed area, the poor convergence and low solution quality are common problems. In this article, an improved ant colony optimization algorithm is proposed. Firstly, by analyzing the constraints of individual UUV platform capability and swarm task, the constraint model and optimization model of UUV swarm task planning are formulated. Secondly, a method of unequal allocation of initial pheromone concentration is designed based on the difference between the average distance and the distance between task points, the optimal and worst thresholds of the optimization model are proposed to classify the ants and complete pheromone update, introduce an innovative “gravity factor” that dynamically changes according to the iteration process to the state transition rule to increase the probability of the suboptimal node being selected in the early and middle of the algorithm. Thirdly, the validity of the algorithm optimization item is analyzed by statistical experiment and the algorithm parameters are optimized according to the number of optimal solutions and average convergence value. Finally, based on the cases from classic documents, by making comparison analysis with basic ant colony algorithm and elite ant colony algorithm, the proposed algorithm in this article improves the percentage of finding approximate optimal solution by 78% and 66% in 50 statistical experiments and average convergence in the 40th generation, which shows good global optimization capability and convergence performance. Experimental results of a designed typical UUV swarm mission planning case with a certain scale show the rapidity and effectiveness of this algorithm in solving the problem of the swarm survey task planning in a wide and sparsely distributed area.

Keywords: UUV swarm; mission planning; ant colony algorithm; constraint model; algorithm parameter optimization

收稿日期: 2022-04-10 Received Date: 2022-04-10

^{*} 基金项目: GF 科技创新特区项目(21-163-05-ZT-002-005-03)、水下机器人重点实验室基金(JCKYS2022SXJQR-09)、哈尔滨工程大学“高水平科研引导专项”(3072022QBZ0403)项目资助

0 引言

水下无人航行器 (unmanned underwater vehicle, UUV) 作为一种小体积、高性能的水下智能设备,被广泛应用于军事与民用场景,如军事打击、情报侦察、资源勘探、海底维修等。现阶段对 UUV 研究的热点问题之一便是其集群任务规划问题,其中最典型也最基础的任务类型就是 UUV 集群勘察任务,该任务已知作业区域中勘察任务点的坐标与所需要的工作负载,在单体 UUV 工作负载有限、续航时间有限的情况下,对集群中的 UUV 及其对应的勘察任务点进行规划,寻找出一种最优的规划方案。

在解决 UUV 集群任务规划问题时,优秀的收敛性能和更高质量的解集一直是解决方案关注的重点。文献[1]通过 SWRL 规则对非预期事件进行检测,有效应对 UUV 集群的动态任务规划;文献[2]将拍卖的思想与动态规划算法相结合,提高了 UUV 集群任务规划方案的质量;文献[3]将分布式多 UUV 系统与多蚁群算法相结合,得到非支配解集,通过设置权重系数在解集中选择更好的规划方案;文献[4]通过改进初始信息、禁忌表以及状态转移公式,提出了一种改进的蚁群算法,提高了解决 UUV 集群任务规划问题的工作效率;文献[5]提出了一种基于一致性算法和共识算法的动态扩展算法,有效解决了在通信约束下的 UUV 集群任务规划问题并获得了良好的性能;文献[6]分别使用蚁群算法 (ant colony algorithm, ACA) 和多蚁群算法对 UUV 集群问题进行分阶段求解,所得的解集能够满足问题的约束条件;文献[7]使用层次分析法将 UUV 集群的任务进行分层,通过对子任务进行处理,在保证了解的质量的情况下保证了整体有效性;文献[8]通过单蚁群系统的单目标优化以及不同蚂蚁系统之间的相互作用,得到多目标优化问题 Pareto 解集,为 UUV 集群侦察任务规划问题提供了更优的方案。

蚁群算法作为最常用的元启发式算法之一,在求解集群任务规划问题上取得了不错的效果,但仍然存在如下缺陷:1) 蚁群算法的运行机制本质上是一种正反馈的行为,在一定程度上会导致算法陷入局部最优,且很难自主跳出;2) 其初始信息素浓度分配方式和信息素更新规则没有针对性,虽然保证了解集的多样性,但是不必要的探索会导致算法的收敛性能变差;3) 算法的随机性和收敛性之间的关系是对立的,收敛性能的提升会加大算法局部最优的概率,降低解的质量。

对蚁群算法的改进,目前的研究中大致有如下 3 个方向。1) 优化蚁群算法结构,改进状态转移规则或优化算法的逻辑框架^[9]。文献[10-11]设计了伪随机状态转

移规则,通过强化最优解的引导来提高算法的收敛性;文献[12]在启发因子中增加惩罚矩阵,用来提高蚁群的搜索效率;文献[13]通过自适应参数对状态转移实行动态规划,平衡了算法的收敛性和随机性;文献[14]改进了算法的状态转移规则,同时增加距离启发函数和平滑启发函数,提高了算法的寻优能力和收敛速度;文献[15]通过在启发因子中加入引导因子,提高了解的质量。2) 优化蚁群算法的信息素规则,优化初始分配方式或信息素更新规则。文献[16]优化了信息素的分配规则,通过降低局部信息素影响,提高了算法的收敛性;文献[17]通过增加优秀路径的信息素浓度优势,提高了算法的收敛性;文献[18]提出信息素挥发的自适应系数,提高了算法的随机性;文献[19]通过设置最大最小值来限制信息素的数值范围,提高了解的质量。3) 融合蚁群算法,将蚁群算法与其他算法的思想或流程相结合。文献[20]结合遗传算法的交叉、变异等操作,提高了算法的求解质量;文献[21]利用遗传算法为初始信息素分配提供依据,提高了算法收敛性;文献[22]通过 Dijkstra 算法规划出次优解,将次优解作为蚁群算法初始方案进行下一步规划,提高了解的质量。

综合上述国内外文献分析,目前在 UUV 集群任务规划问题的求解中,普遍存在解质量不高、随机性与收敛性互相矛盾、算法自主跳出局部最优能力不强的问题,而这些问题很大程度影响算法的全局寻优能力和收敛性能。

因此,本文结合 UUV 集群勘察任务类型的需求,改进设计了 UUV 集群任务规划蚁群算法。通过设计一种初始信息素浓度的不均等分配方法,减少算法前期的无效搜索,提高算法前期解的质量和收敛性能;提出了优化模型的最佳与最差评价价值阈值,通过设计其赋值规则,将每一代蚁群中的蚂蚁进行分类,施以不同的信息素浓度更新方法,在保证算法的解的质量向更好方向迭代的前提下提高算法的收敛性能;在状态转移规则中加入新的影响因子,通过增加蚂蚁在进行节点选择时选中次优节点的概率增强算法自主跳出局部最优的能力,提高算法的随机性和解的质量。

1 UUV 集群任务规划问题及数学模型

1.1 问题描述

本文研究的 UUV 集群勘察任务规划问题与 VRP (vehicle routing problem) 问题较为相似,均为某基地拥有 K 个 UUV,前往作业区域内已知位置以及工作负载需求量 q_i 的共计 L 个任务点进行勘察作业,每个 UUV 可以对多个任务点进行作业,完成作业任务后需返回基地。UUV 集群勘察任务规划示意图如图 1 所示。

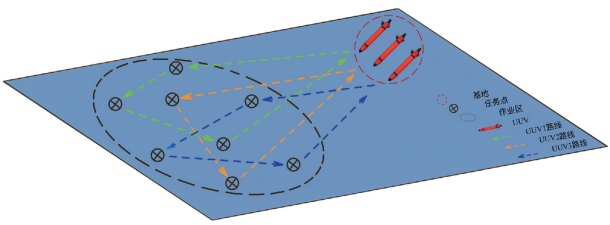


图 1 UUV 集群勘察任务规划示意图

Fig. 1 Schematic diagram of UUVs survey task planning

单体 UUV 有工作负载和续航时间(行驶距离)的限制,集群勘察任务的约束条件如下:1)所有任务点均需要进行勘察作业;2)每个任务点只能作业一次;3)每个执行勘察任务的 UUV 最多离开基地一次;4)每个 UUV 的工作负载量满足其进行勘察作业的所有任务点的总需求;5)参与勘察任务的 UUV 的续航距离满足其在不同任务点之间作业所走过的路径长度。

1.2 符号定义

本文建立的 UUV 集群勘察任务规划问题数学模型中所涉及的参数如表 1 所示。

表 1 符号定义
Table 1 Parameter symbol definition

符号	定义
L_i	基地 l_0 以及各任务点 ($i=1,2,\dots,L$)
d_{ij}	任务点 i 到任务点 j 的距离
q_i	任务点 i 的需求量
D	UUV 的最大续航距离
Q	UUV 最大工作负载量
K	可使用 UUV 的最大数量
k	参与勘察任务的 UUV 的编号 ($k=1,2,\dots,K$)
n_{car}	参与勘察任务的 UUV 总数
n_k	参与勘察任务的第 k 个 UUV 所完成的任务点总数 ($n_k \neq 0$)
r_k^i	参与勘察任务的第 k 个 UUV 所完成的第 i 个任务点
R_k	参与勘察任务的第 k 个 UUV 所完成的任务点的集合 ($R_k = \{r_k^1, r_k^2, \dots, r_k^{n_k}\}$)

1.3 约束条件

UUV 集群勘察任务规划问题的约束条件可以总结为:

$$n_{car} \leq K \tag{1}$$

$$R_{k1} \cap R_{k2} = \emptyset, k1 \neq k2 \tag{2}$$

$$\bigcup_{k=1}^{n_{car}} R_k = \{1, 2, \dots, L\} \tag{3}$$

$$\sum_{i=1}^{n_k} q_{r_k^i} \leq Q \tag{4}$$

$$d_{0,r_k^1} + \sum_{i=2}^{n_k-1} d_{r_k^{i-1},r_k^i} + d_{r_k^{n_k},0} \leq D \tag{5}$$

1.4 优化目标

UUV 集群勘察任务规划问题的优化目标为所有 UUV 的总航程最短。

$$\min Z = \sum_{k=1}^{n_{car}} (d_{0,r_k^1} + \sum_{i=2}^{n_k-1} d_{r_k^{i-1},r_k^i} + d_{r_k^{n_k},0}) \leq D \tag{6}$$

2 算法设计

2.1 基本 ACA 算法

ACA 算法最早由 Dorigo^[23] 在 1991 年提出,算法的思想来源于蚂蚁觅食行为,蚁群中的蚂蚁所带的信息素会被遗留在其所经过的路径上,累积的信息素随时间挥发,最终离食物更短的路径上会积累更多的信息素以被后续蚂蚁作为路径选择时的参考信息。

在解决 UUV 集群勘察任务规划问题时,使用蚁群模拟 UUV,当 UUV 在任务点 i 完成作业任务之后选择下一任务点 j 时,主要考虑两个因素:1)任务点 i 与任务点 j 的距离,即启发因子 η_{ij} ,2)从任务点 i 到任务点 j 的可行性,即信息素浓度 τ_{ij} 。

在 t 时刻,蚂蚁 k 从任务点 i 选择下一任务点 j 的概率如下:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{j \in j_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}, & j \in j_k \\ 0, & j \notin j_k \end{cases} \tag{7}$$

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{8}$$

信息素浓度与启发因子对状态转移概率的影响如下:在待选节点中,与当前节点间路径的信息素浓度越高的节点,被选为下一节点的概率越大,反之则越小;与当前节点间距离越短的节点,被选择为下一节点的概率越大,反之则越小。

在转移概率公式中 α 为信息素的重要程度; β 为启发因子的重要程度; j_k 为满足约束条件的下一待选任务点的集合。 j_k 中的可选任务点 j 要满足 3 个条件:1) $j \in tabu$,其中 $tabu$ 为禁忌表,是尚未完成勘察作业的任务点的集合;2) $Q(t)-q_j \geq 0$, $Q(t)$ 为当前时刻 UUV 剩余的工作装载量;3) $D(t)-d_{ij}-d_{j,0} \geq 0$, $D(t)$ 为当前时刻 UUV 剩余的续航距离。

当 $J_k = \emptyset$ 时,UUV 一定会选择返回基地;当 $J_k \neq \emptyset$ 且 UUV 当前所在任务点不是基地时,UUV 也会以一定的概率选择返回基地。如果蚂蚁选择基地作为下一节点,会在基地更新工作负载量与续航距离,进而代表新的 UUV 重新出发,并记录所代表的 UUV 的数量。信息素的更新

采用代更新的形式,当一代蚂蚁循环完毕之后,挑选出UUV数目符合限制条件的有效蚂蚁路径进行信息素的更新。更新公式如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (9)$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m_e} \Delta\tau_{ij}^k \quad (10)$$

$$\Delta\tau_{ij}^k = \frac{Q_{ant}}{L_k} \quad (11)$$

式中: ρ 为信息素的挥发系数; m_e 为符合UUV数目限制的蚂蚁数量; $\Delta\tau_{ij}$ 为符合条件的 m_e 只蚂蚁在任务点 i 与任务点 j 之间的路径上留下的信息素总和; $\Delta\tau_{ij}^k$ 为第 k 只蚂蚁在任务点 i 与任务点 j 之间的路径上留下的信息素; Q_{ant} 为一只蚂蚁的信息素总量; L_k 为第 k 只蚂蚁走过的路径总长度。

2.2 改进初始信息素分配

在基本蚁群算法中,路径上的初始信息素浓度是均等的。在算法前期,由于迭代次数少,路径间积累的信息素浓度表现不出差异,此时转移概率的计算主要取决于启发因子,即距离的倒数。但是因为节点间距离的量级相差不大且一般情况下均大于1,所以在算法前期迭代的节点选择中,仍会以一定的概率选择与当前节点相距较远的节点作为下一节点。这不利于在更短的路径上更快地积累更多的信息素,对于以路径总距离最短为优化目标的数学模型来说是一种无效的搜索。

本文受贪心算法^[24]的核心思想即“每一步选择当前的最优解则最终解一定为全局近似最优解”的启发,通过对初始信息素的浓度进行不均等分配,使得距离近的节点间与距离远的节点间的信息素浓度出现差异,引导蚂蚁在算法前期的节点选择中以更大的概率选择与当前节点距离更近的节点,降低算法前期的迭代值,为算法的快速收敛奠定良好基础。

在算法之初,求解当前任务点与其他所有任务点之间的距离均值,将其他任务点到此点的距离与均值做比较,小于均值的任务点到当前点之间路径上的信息素会进行叠加,反之进行额外挥发。信息素初始化公式如下:

$$\tau_{ij}(0) = T_0 + \Delta\tau_{ij}(0) \quad (12)$$

$$\Delta\tau_{ij}(0) = -\left(d_{ij} - \sum_{j=0}^{j=L} d_{ij} / (L-1)\right), i \neq j \quad (13)$$

式中: T_0 表示给定的初始信息素浓度; $\Delta\tau_{ij}(0)$ 表示附加的初始信息素浓度。任务点间距离小于平均距离时 $\Delta\tau_{ij}(0)$ 为正,反之为负。

文献[25]对蚁群算法的收敛性进行了数学分析

并得出若干推论,其推论3“使用最近邻策略可使收敛速度提高”与本文通过改进初始信息素浓度来增加算法前期节点选择中选择距离更近节点的思想不谋而合。

2.3 改进信息素更新规则

1996年,Dorigo等^[26]提出了一种带精英策略的蚁群算法,通过对每一次迭代中结果最好的若干只蚂蚁赋予额外信息素叠加,改进了算法的信息素更新规则。

大量的文献表明,精英蚁群算法虽然收敛性良好,但是容易陷入局部最优。原因在于单次迭代产生的优秀解偶然性太大,叠加额外的信息素可能会误导算法向不优秀的解路径偏移,陷入局部最优。

为了尽可能减小额外的信息素叠加导致局部最优的可能性,本文对精英蚁群算法的信息素更新规则进行了优化,基于之前迭代中产生的最优解对本次迭代中的优秀蚂蚁进行分类,针对不同类型的优秀解进行不同程度的额外信息素叠加。同时,为了进一步加快算法的收敛,对每一代中最差的若干个解进行分类并惩罚,施以额外的信息素挥发。

为此,本文提出了优化模型的最佳评价价值阈值 val_{best} 与最差评价价值阈值 val_{worst} ,赋值规则如下:

$$val_{best} = \begin{cases} val_{best}(0), & iter = 0 \\ val_{best}(i), & iter > 0 \text{ 且 } val_{best}(i) < val_{best} \\ val_{best}, & iter > 0 \text{ 且 } val_{best}(i) \geq val_{best} \end{cases} \quad (14)$$

$$val_{worst} = \begin{cases} val_{worst}(0), & iter = 0 \\ val_{worst}(i), & iter > 0 \text{ 且 } val_{worst}(i) < val_{worst} \\ val_{worst}, & iter > 0 \text{ 且 } val_{worst}(i) \geq val_{worst} \end{cases} \quad (15)$$

式中: $val_{best}(i)$ 表示第 i 代蚂蚁的最佳评价价值,当新一代迭代的最佳评价价值优于算法的最佳阈值时对 val_{best} 进行更新; $val_{worst}(i)$ 为第 i 次迭代的最差评价价值,值得注意的是,当新一代蚂蚁中最差蚂蚁的评价价值仍然优于最差评价价值阈值时才对 val_{worst} 进行更新。

为了更好地发挥奖励策略与惩罚策略的优势,取消了每次迭代后对所有蚂蚁的信息素叠加,只对一定比例的蚂蚁进行信息素的叠加。改进后的信息素更新规则如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau(add)_{ij} \quad (16)$$

在优秀蚂蚁集合中,评价价值优于最佳阈值 val_{best} 的蚂蚁路径进行更多的信息素叠加,评价价值差于最佳阈值 val_{best} 的蚂蚁路径仅进行少量的信息素叠加;在不优秀蚂蚁集合中,评价价值差于最差阈值 val_{worst} 的蚂蚁路径进行更多的额外信息素挥发,评价价值优于最差阈值

val_{worst} 的蚂蚁路径进行少量的额外信息素挥发,公式如下:

$$\Delta\tau(\text{add})_{ij} = \begin{cases} \sigma_1 \cdot \Delta\tau_{ij}, & \text{ant} \in \text{ant}_{\text{best}} \text{ 且 } val_{\text{ant}} = val_{\text{best}}(i) \\ & \text{且 } val_{\text{ant}} < val_{\text{best}} \\ \sigma_2 \cdot \Delta\tau_{ij}, & \text{ant} \in \text{ant}_{\text{best}} \text{ 且 } val_{\text{ant}} \neq val_{\text{best}}(i) \\ & \text{且 } val_{\text{ant}} < val_{\text{best}} \\ \sigma_3 \cdot \Delta\tau_{ij}, & \text{ant} \in \text{ant}_{\text{best}} \text{ 且 } val_{\text{ant}} \geq val_{\text{best}} \\ \sigma_4 \cdot \Delta\tau_{ij}, & \text{ant} \in \text{ant}_{\text{worst}} \text{ 且 } val_{\text{ant}} < val_{\text{worst}} \\ \sigma_5 \cdot \Delta\tau_{ij}, & \text{ant} \in \text{ant}_{\text{worst}} \text{ 且 } val_{\text{ant}} \geq val_{\text{worst}} \\ 0, & \text{其他} \end{cases} \quad (17)$$

式中: ant_{best} 为本代中优秀蚂蚁的集合,占比为本代有效蚂蚁总数的前 1/4; $\text{ant}_{\text{worst}}$ 为本代中不优秀的蚂蚁的集合,占比为后 1/4; val_{ant} 代表第 ant 只蚂蚁的评价值; $val_{\text{best}}(i)$ 代表第 i 代蚂蚁的最佳评价值;使用文献[27]对于精英蚁群算法参数选取的方法,参数取值为 $\sigma_1 = 8$, $\sigma_2 = 2$, $\sigma_3 = 1$, $\sigma_4 = -0.25$, $\sigma_5 = -0.75$ 。

与此同时,为了防止算法出现早熟和滞胀现象,引入最大最小蚁群(max-min ant system, MMAS)^[19]中的最小信息素 $\tau_{\min}$,具体公式如下:

$$\tau_{ij}(t+1) = \begin{cases} \tau_{\min}, & \tau_{ij}(t+1) < \tau_{\min} \\ \tau_{ij}(t+1), & \text{其他} \end{cases} \quad (18)$$

结合最佳阈值,根据之前迭代的最好结果对本次迭代的优秀解进行分类再处理,可以尽可能地避免精英蚁群算法因额外的信息素叠加导致的局部最优的情况。结合最差阈值,增加惩罚机制,进一步加大表现差的路径与表现好的路径间的信息素浓度差异,增加算法的收敛性能。

文献[25]对于蚁群算法收敛性分析的推论 4 印证了此方法对于提高算法收敛性的作用。

2.4 改进状态转移规则

在本文对蚁群算法的改进思路中,无论是对初始信息素浓度分配方式的优化还是对信息素更新规则的优化,都提高了算法的收敛性。虽然尽量避免了因额外的信息素叠加导致的局部最优,然而对于整个算法来说,收敛性的提高必然会引起算法随机性的降低,增加算法陷入局部最优的可能性,且因为信息素浓度对算法收敛的正反馈作用,在陷入局部最优后算法很难自主跳出。

在对局部最优问题的研究中,一种广泛使用的方法是模拟退火算法^[28]。模拟退火算法的核心思想是以一定的概率产生新解并接受次优解,增加算法的随机性,降低陷入局部最优的概率。

以模拟退火算法的思想为牵引,本文对蚁群算法的状态转移公式进行了改进。通过增加新的影响因子“引力系数 γ ”,以一定的规则主动增加节点选择过程中次优

节点被选中的概率,在相对良好的范围内提高算法的随机性,增强算法跳出局部最优的能力。

思路为在任意两个节点间都增加一种引力, $\gamma \in [0, 1]$, 引力可以减小状态转移概率计算中节点间距离的值,由式(19)可知,随着节点间引力增大,两点间距离值变小,启发因子值变大,此两个节点被蚂蚁连续选择的概率会增大。节点间引力的大小取决于蚂蚁路径中节点被选择的顺序,由式(21)、(22)可知,最初时节点间引力均为 0,两个节点间的引力会随着没有连续选择此两节点的蚂蚁数量的增加而增大,其规律符合泊松分布曲线。当此两节点被连续选中时,两点间引力会置 0 并重新开始变化。

$$\eta_{ij} = \frac{1}{d_{ij}(1 - \gamma_{ij})}^{\chi} \quad (19)$$

$$\chi = \frac{N_{\max} - N}{N_{\max}} \quad (20)$$

$$\gamma_{ij} = \sigma \frac{\lambda^{k_{ij}}}{k_{ij}!} e^{-\lambda} \quad (21)$$

$$k_{ij} = k_{ji} = \begin{cases} 0, & \text{本只蚂蚁路径中节点 } i \text{ 与 } j \text{ 连续} \\ (k_{ij} + 1) \% (2\lambda), & \text{本只中 } i \text{ 与 } j \text{ 不连续} \end{cases} \quad (22)$$

式(21)中的参数 σ 决定了两点间引力系数 γ 的峰值;参数 λ 决定了两点间引力系数 γ 达到峰值的频率,即达到峰值所需的没有连续选择此两点的蚂蚁的数量。

通过合理选用参数 σ 与参数 λ 的值,可以限制节点间引力系数的峰值和频率,使得引力系数对算法随机性提升的效果被限制在一个良好的范围内。

当算法已陷入局部最优或即将陷入局部最优(出现迭代值的平台期)时,较大比例的蚂蚁会选择同一条路径。此时的节点选择中,当前节点与离其较近但长时间未被连续选择的若干节点(次优节点)间的引力系数会增大,这些节点被选为下一节点的概率也会增大,进而为算法提供更多样的且质量可以保证的路径选择,提高算法跳出局部最优的能力。

因为本文的改进思路对初始信息素浓度的分配方式和信息素更新规则进行了优化,在算法前期和中期,相近节点间的信息素浓度与相远节点间的信息素浓度差异较为明显。在节点选择过程中,离当前节点较近的节点被选为下一节点的概率比离当前节点较远的节点被选为下一节点的概率大得多。通过合理设置参数值,引力系数的机制对次优节点被选中的概率的提升明显高于较远的节点。且因为引力系数的变化符合泊松分布曲线,较远节点与当前节点间的引力值会在增加后又重新回降。

综上所述,节点选择中的次优节点是引力系数机制的最大受益者, γ 对算法随机性的提高被限制在了一个良好的范围内(次优节点中)。

算法前期与算法中期出现局部最优的情况对算法的收敛效果影响更大,所以需要提高前期和中期的随机性,而在算法后期提高随机性会损害算法的收敛性能。为了同时保证算法后期的快速稳定收敛,根据迭代次数的增加以自适应的方式逐渐减小引力系数和对转移概率的影响,如式(20)所示,其中 N 为当前迭代数, $N_{\max}$ 为最大迭代次数。

3 UUV 集群任务规划的改进蚁群优化算法设计

将改进蚁群算法应用于 UUV 集群勘察任务规划问题时的具体实现步骤如下。

- 1) 初始化蚁群算法的参数,根据任务点间距离对信息素初始浓度进行不均等分配。
- 2) 根据参与勘察任务的 UUV 剩余的工作负载量与 UUV 剩余续航里程从禁忌表中挑选可以选择的下一任务点。
- 3) 计算可选任务点的转移概率,利用轮盘赌进行节点选择。
- 4) 更新 UUV 的工作负载与续航里程,更新蚂蚁路径,更新禁忌表,重复步骤 2)、3) 直至完成所有任务点的作业。
- 5) 根据参与勘察任务的 UUV 数量判断此只蚂蚁是否为有效蚂蚁,将符合 UUV 数量限制的蚂蚁路径以及对应的评价值保存,更新引力系数矩阵 γ 。

6) 待本代所有蚂蚁跑完,对本代中所有有效蚂蚁进行排序,先按照信息素更新规则对信息素进行更新,再更新评价值最佳阈值 val_{best} 与最差阈值 val_{worst} ,记录本代最佳蚂蚁对应的路径与评价值。

7) 重复步骤 2) ~ 6) 直至所有代蚂蚁跑完。在所有代的代最佳蚂蚁中挑选本次算法运行的最佳评价值及对应的最佳路径方案。

8) 算法结束,绘制最佳方案对应的勘察方案路径图以及本次算法的收敛曲线。

4 UUV 集群任务规划的改进蚁群优化算法有效性分析与参数选取

使用文献[20]的案例对本文改进蚁群算法的 3 个优化项进行有效性分析和相关参数的选取。

4.1 案例问题描述(案例 1)

已知某基地拥有 5 艘可执行勘察任务的 UUV,每个 UUV 的最大续航里程为 50 km,最大工作负载量为 8,作业区域中待勘察的任务点共有 20 个。基地的坐标为 (14.5 km, 13 km), 20 个任务点的坐标以及负载需求量如表 2 所示。

表 2 任务点信息表

Table 2 Mission point information sheet

ID	X/km	Y/km	q	ID	X/km	Y/km	q
1	12.8	8.50	0.1	11	6.70	16.9	0.9
2	18.4	3.40	0.4	12	14.8	2.60	1.3
3	15.4	16.6	1.2	13	1.80	8.70	1.3
4	18.9	15.2	1.5	14	17.1	11.0	1.9
5	15.5	11.6	0.8	15	7.40	1.00	1.7
6	3.90	10.6	1.3	16	0.20	2.80	1.1
7	10.6	7.60	1.7	17	11.9	19.8	1.5
8	8.60	8.40	0.6	18	13.2	15.1	1.6
9	12.5	2.10	1.2	19	6.40	5.60	1.7
10	13.8	5.20	0.4	20	9.60	14.8	1.5

4.2 算法优化项的有效性分析

针对本文提出的初始信息素分配优化项、信息素更新优化项和状态转移优化项,通过对照统计实验对其有效性进行了分析。对比方案一组为基本蚁群和加入某个优化项的基本蚁群;另一组为本文完整的改进蚁群和移除某个优化项的改进蚁群。

其中,对比统计实验的次数为 50 次,算例 1 已知的最优解为 107.84 km。

1) 对初始信息素分配优化项的有效性分析

第 1 组与第 2 组对比结果取用的皆是统计实验中的最佳次运行结果。收敛对比如图 2 所示。

(1) 对收敛质量的影响

从图 2(a) 可以看出,两种算法均找到了最优解 107.84 km,但算法最终均未收敛。这是因为,蚁群算法的收敛在本质上是路径间信息素浓度差异的收敛,而基本蚁群算法的信息素更新规则并不能明显拉开路径间信息素差异,算法很难实现收敛。

但是加入初始信息素分配优化项的基本蚁群算法在算法前期有明显更低的迭代值,且算法后期迭代值的平均值也低于基本蚁群算法。由此可见,本文对初始信息素浓度分配方式的优化可以提高算法前期的解的质量,同时也为算法最终解的质量起到积极作用。

(2) 对收敛速度的影响

从图 2(b) 可以看出,移除优化项的改进算法,算法前期的迭代值变高,解的质量变差,导致算法收敛所需要的代数增加,收敛速度慢于完整的改进蚁群算法。由此可见,初始信息素分配优化项通过降低算法前期的迭代值,加快了算法的收敛速度。

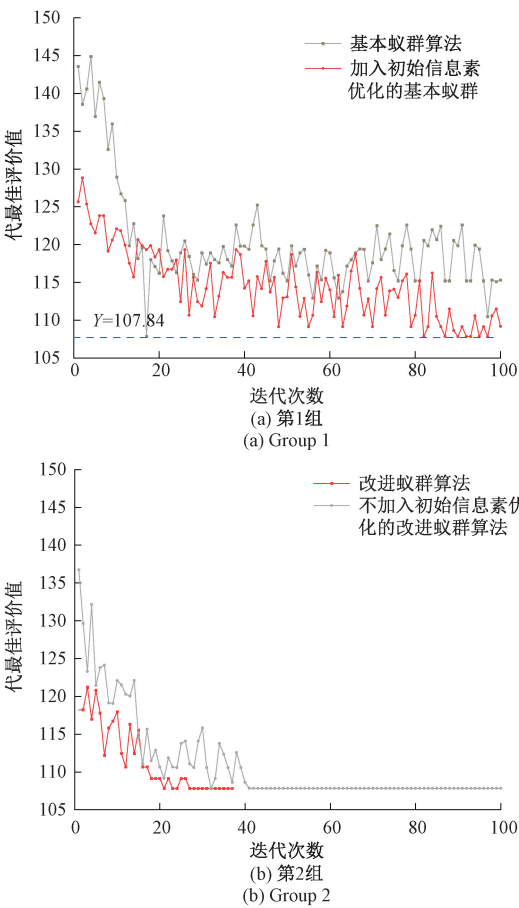


图 2 初始信息素优化项收敛对比
Fig. 2 Comparison diagram of convergence of initial pheromone optimization items

2) 对信息素更新优化项的有效性分析

本文的信息素更新优化项是基于精英蚁群算法的信息素更新规则,故而第 1 组对比实验更改为精英蚁群算法与加入信息素更新优化项的精英蚁群算法。

第 1 组与第 2 组对比结果取用的分别是统计实验中的典型次与最佳次运行结果。收敛对比如图 3 所示。

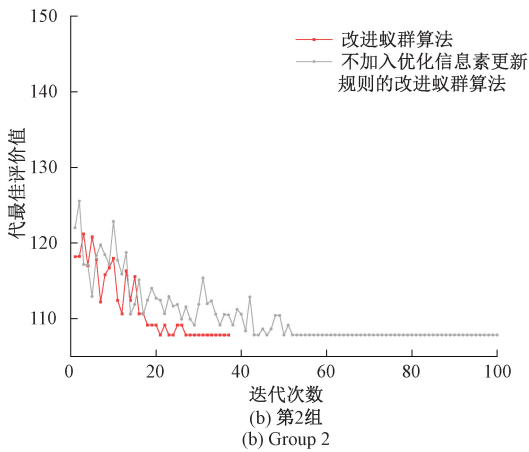
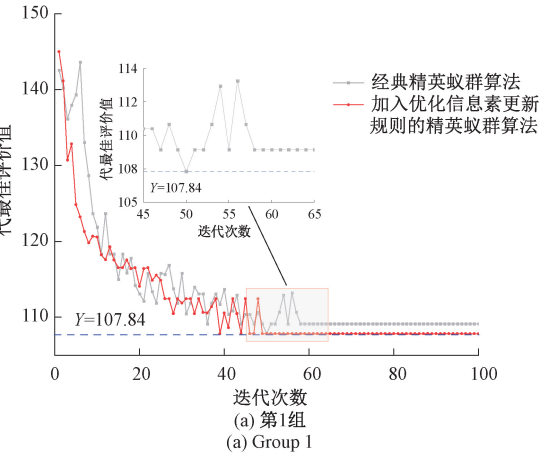


图 3 信息素更新优化项收敛对比
Fig. 3 Comparison diagram of convergence of pheromone updating optimization term

(1) 对收敛质量的影响

从图 3(a)可以看出,精英蚁群算法虽然找到最优解 107.84 km,但因为对优秀蚂蚁的额外信息素叠加没有任何限制,算法陷入局部最优,并最终收敛在 109.139 km。加入信息素更新优化项的精英蚁群算法能在找到最优解 107.84 km 之后实现稳定收敛。

本文在精英蚁群信息素更新规则基础上,增加了“通过往次最佳迭代值对本次迭代中优秀蚂蚁进行分类再施以不同程度奖励”的策略,避免因信息素的额外叠加造成算法的局部最优,提高了算法的解的质量。

(2) 对收敛速度的影响

从图 3(b)可以看出,移除信息素更新优化项的改进算法能够收敛到最优值 107.84 km,但收敛速度明显慢于完整的改进算法。这是因为在信息素更新优化项中,取消了对所有蚂蚁的信息素叠加,同时增加对不优秀蚂蚁的惩罚策略,提高了算法的收敛速度。

3) 对状态转移优化项的有效性分析

第 1 组与第 2 组对比结果取用的分别是统计实验中的最佳次与典型次运行结果。收敛对比如图 4 所示。

(1) 对收敛质量的影响

从图 4(a)可以看出,加入状态转移优化项的基本蚁群算法同样没有实现收敛,但算法后期振荡的幅度明显变小,这是由于状态转移优化项可以提升次优节点在节点选择过程中被选中的概率。

从图 4(b)可以看出,移除状态转移优化项的改进算法陷入了局部最优;完整的改进蚁群算法在跳出两次迭代值平台期后最终找到最优解 107.84 km 并实现收敛。

在完整改进算法(右图)迭代的第 2 个平台期中,算法在第 35 代与第 46 代找到了两个更低的代最优值,最

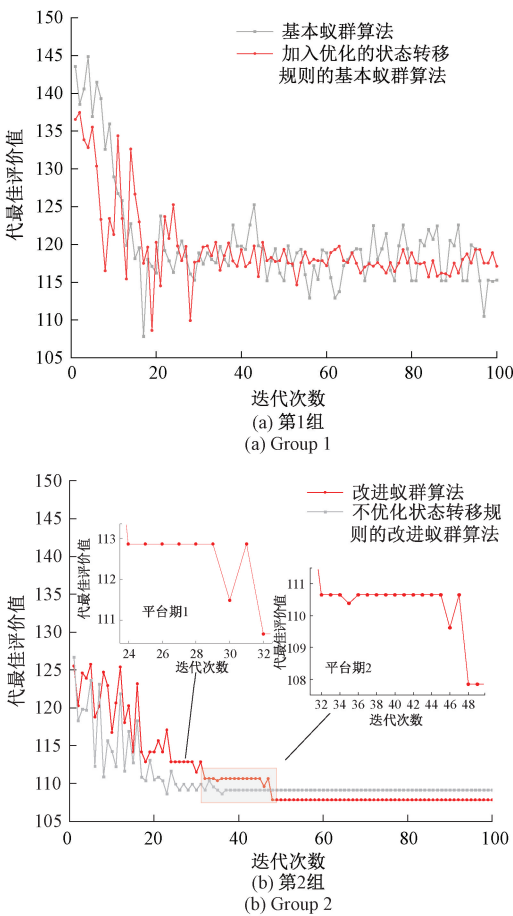


图 4 状态转移优化项收敛对比

Fig. 4 Comparison diagram of state transition optimization term convergence

终主动跳出了长达 15 代(32~47 代)的平台期,避免了陷入局部最优。这得益于如下两点:①本文对信息素更新规则的优化中,对精英蚂蚁的信息素叠加要参考之前次迭代的最优值(最佳阈值),在第 35 代时算法找到了比平台期迭代值更低的解,所以 36~45 代时对精英蚂蚁的信息素叠加按策略中的最低标准进行,平台值对应的路径上的信息素浓度没有被过度增加;②本文算法对状态转移规则的优化中,算法会在节点选择中以一定的概率选择次优节点,在良好的节点范围内增加随机性以找到新的优秀解(第 35 代和第 46 代的迭代值)。

由此可见,对状态转移规则的改进有利于算法跳出局部最优,提高算法找到最优解的概率。

(2) 对收敛速度的影响

算法随机性的提高会带来收敛性的降低,由右图可知,完整的改进蚁群算法收敛速度略慢于移除状态转移规则优化项的改进算法,但在可接受的范围内。

4) 对 3 个优化项的综合分析

为了能够更直观地表现出各优化项的效果,所以

图 3(a) 对比和图 4(b) 对比中选取的是典型次的算法运行结果。

为了更充分地分析各优化项的作用,现通过 50 次统计实验的数据进行补充说明,从收敛均值、近似最优解出现的次数、最优解出现的次数以及收敛所需的平均代数 4 个方面进行,统计结果如表 3 所示。

表 3 优化项对算法性能影响统计

Table 3 Statistical table of influence of optimization items on algorithm performance

算法名称	收敛均值 /km	近似最 优解数	最优 解数	收敛 均代
完整改进蚁群	109.656	47	15	41
移除初始信息素优化项	110.083	44	13	47
移除信息素更新优化项	111.417	28	9	56
移除状态转移优化项	110.575	43	6	39

由表 3 可知,当移除掉初始信息素分配优化项后,统计实验中算法的 4 项性能均小幅度下降,体现了此优化项对算法最终收敛质量和收敛速度的提升;移除掉信息素更新优化项之后,算法的性能明显下降,尤其是收敛均值、近似最优解出现的次数以及平均收敛代数,体现了此优化项对于提升算法平均收敛质量和收敛速度的重要作用;移除掉状态转移优化项之后,统计实验中最优解出现的次数有明显的下降,体现了状态转移规则的优化对避免局部最优的重要作用,但平均收敛代数有所减少,说明此优化项对算法收敛速度有轻微的损害。

综上所述,在 3 项优化项中,信息素更新优化项能大幅提高算法的收敛质量和收敛性能;状态转移优化项能明显增强算法跳出局部最优的能力;初始信息素分配优化项能明显降低算法前期的收敛值,在一定程度上对算法的收敛质量和收敛速度起到促进作用。

4.3 算法参数优化

本文提出的改进蚁群算法中涉及的参数主要分类 3 类,分别为算法初始参数、算法基本参数以及优化项中涉及到的参数。

本文对参数进行优化时以统计实验中算法的平均收敛值 and 最优解出现的次数为参考评价指标。

1) 算法初始参数

算法初始参数包括种群规模,算法迭代次数,每只蚂蚁的信息素总量 Q ,路径上初始信息素浓度 T_0 。

取常见的参数组合中种群规模设为 50,迭代次数设为 100, $Q = 100, T_0 = 5$ 。

2) 算法基本参数

蚁群算法的基本参数主要包括:信息素的重要程度

α ;启发因子的重要程度 β ;信息素挥发系数 ρ 。文献[29]在对蚁群算法的参数进行统计分析后得出结论,参数最优组合为 $\alpha = 1, \beta = 1 \sim 5, \rho = 0.1 \sim 0.7$ 。

因为本文的初始信息素浓度 $T_0 = 5$,取值略小于文献中的值,因此挥发系数 ρ 的值拟取 $\rho = 0.1 \sim 0.5$ 。

(1)对 β 的取值

取 $\alpha = 1, \beta$ 以 0.5 的间隔由 1~5 递增做统计实验,每种参数组合的统计实验次数为 1 000 次,启发因子对算法性能的影响曲线如图 5 所示。

在统计结果中, $\beta = 2.5, 3.0, 3.5, 4.0, 4.5$ 时统计实验的收敛均值和最优解次数均不错。其中 $\beta = 3$ 时收敛均值最低且最优解次数最多。

$\beta = 2.5$ 时,1 000 次算法运行中寻得最优解 234 次,较 $\beta = 4.0$ 时的 191 次多 43 次,但前者收敛均值反而比后者高,说明 $\beta = 2.5$ 时算法收敛结果的方差大,寻得近似最优解次数少,所以在后续过程中对 $\beta = 2.5$ 不予考虑。对 $\alpha = 1, \beta = 3.0, 3.5, 4.0, 4.5$ 做后续分析。

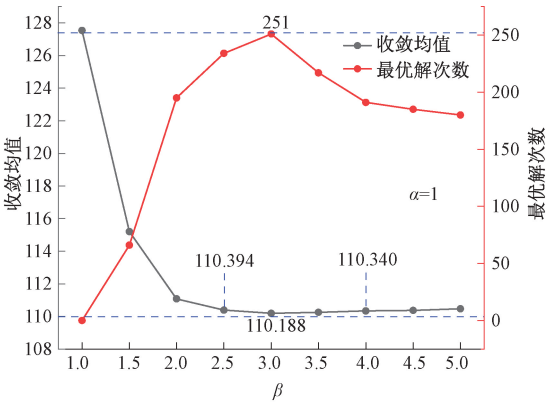


图5 启发因子对性能的影响曲线

Fig. 5 Influence curve of heuristic factor on performance

(2)对 ρ 的取值

取 $\alpha = 1$,分别取 $\beta = 3.0, 3.5, 4.0, 4.5$ 时,参数 ρ 以 0.05 的间隔由 0.1~0.5 递增做统计实验,每种参数组合的统计实验次数为 1 000 次,统计结果如图 6 所示。

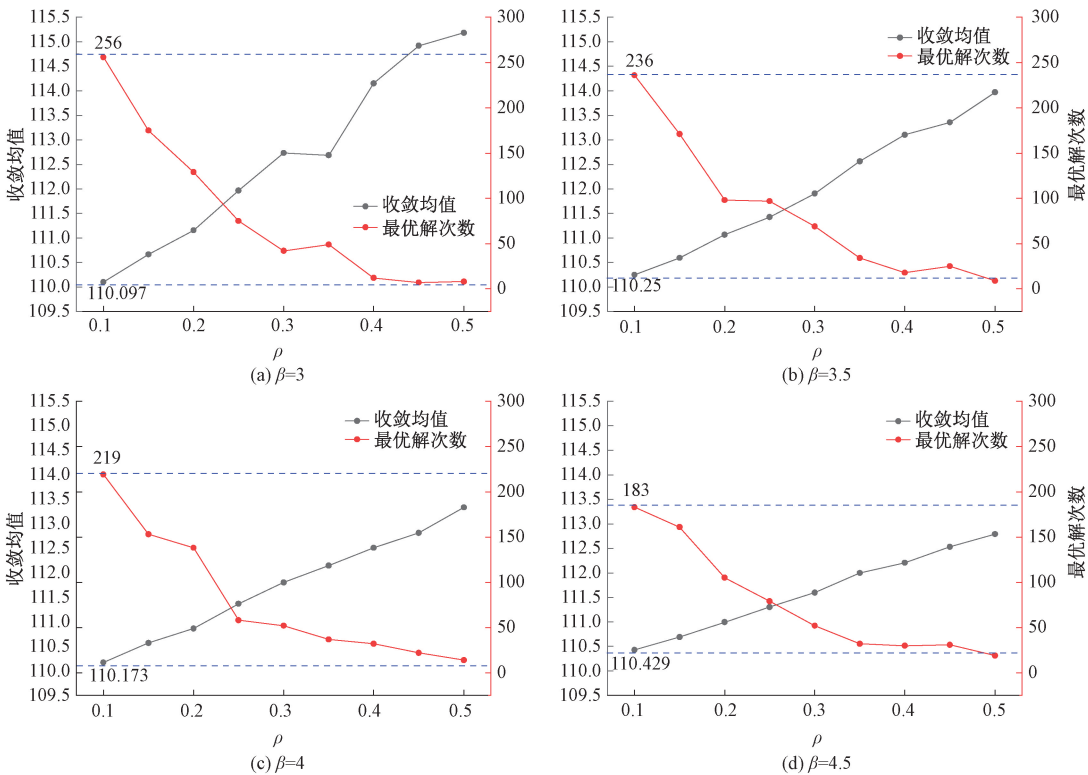


图6 挥发因子对性能的影响曲线

Fig. 6 Influence curve of volatilization factor on performance

从统计结果可以看出,无论是收敛均值还是最优解出现的次数, $\rho = 0.1$ 所在的组合表现都是最好的,所以参数 ρ 确定为 0.1。其中 $\alpha = 1, \beta = 3, \rho = 0.1$ 时的统计实验的平均收敛值最低且最优解次数最多。

综上,参数组合 $\alpha = 1, \beta = 3, \rho = 0.1$ 为当前最优的选

择,对 $\beta = 3, 3.5, 4.0, 4.5$ 的组合均作后续分析。

3)优化项参数

在对状态转移规则的优化中,引力系数 γ 的变化规律符合比松分布的特点, γ 的值涉及到两个影响算法性能的参数:泊松分布的放大倍数 δ ,泊松分布的强度系数

λ 。在对信息素更新规则的优化中,涉及到的参数为最小信息素浓度 $\tau_{\min}$ 。

(1)对 δ 的取值

δ 的取值决定了 γ 的峰值,其中 $\gamma \in [0, 1)$, γ 的取值要在合理的范围内,拟定 γ 最小取 0.2,最大取 0.8,由泊

松分布公式可得,对应的 δ 的取值为 2 ~ 9。

在 $\alpha = 1, \rho = 0.1$ 的基础上,分别取 $\beta = 3.0, 3.5, 4.0, 4.5$ 时,参数 δ 以 0.5 的间隔由 2~9 递增做统计实验,每种参数组合的统计实验次数为 1 000 次,结果如图 7 所示。

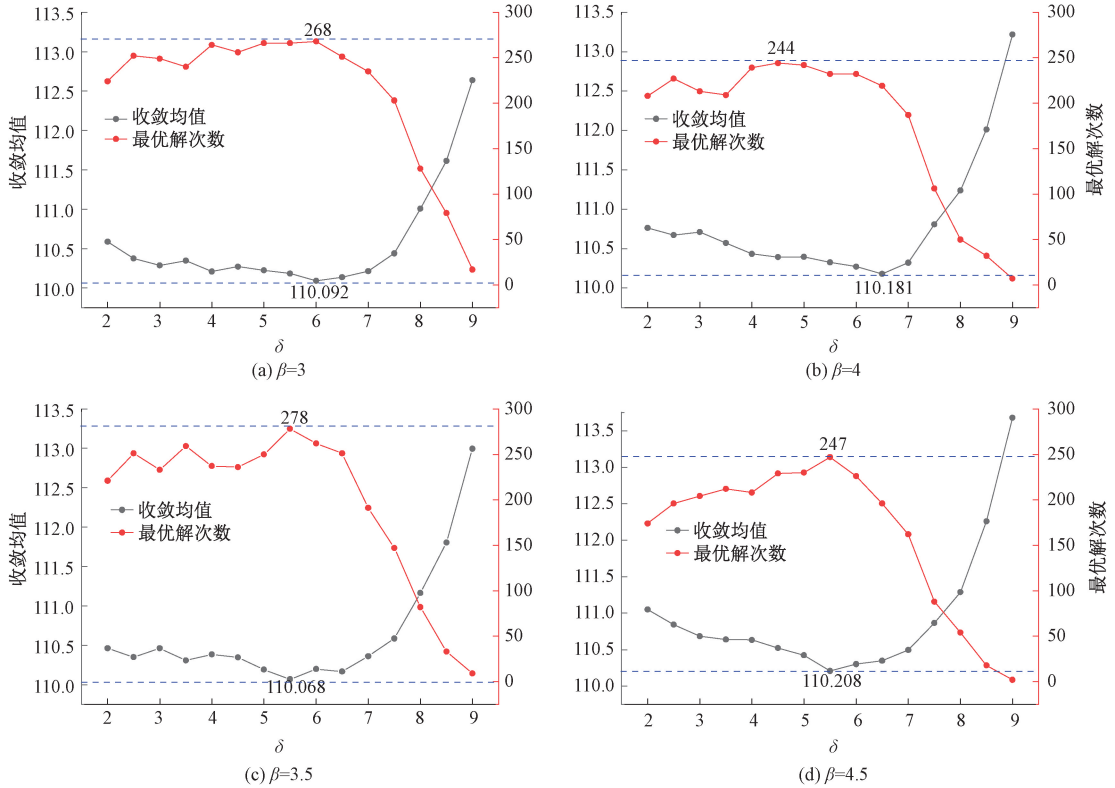


图7 参数 β 对性能的影响曲线

Fig.7 Parameter β influence curve on performance

δ 的大小会影响引力系数的峰值,影响次优节点被选择的概率,进而影响算法的质量。由曲线规律可知, δ 的取值要适中。 δ 太小,引力系数峰值小,对次优节点被选中的概率的提升不大,状态转移优化项的作用不明显; δ 过大则引力系数对节点选择的影响过大,节点选择过程中的随机性不会被限制在次优节点中,收敛均值和最优解出现的次数明显变差。

$\beta = 4.0$ 和 $\beta = 4.5$ 时统计实验中算法表现不如前两者,不做后续参考。

在 $\beta = 3$ 的前提下,取 $\delta = 6$ 时的收敛均值和最优解出现次数均为最佳;在 $\beta = 3.5$ 的前提下,取 $\delta = 5.5$ 时的性能表现也均为最佳。

在 $\alpha = 1, \rho = 0.1$ 的基础上,取参数组合 $\beta = 3, \delta = 6$ 和 $\beta = 3.5, \delta = 5.5$ 作后续分析。

(2)对 λ 的取值

λ 的大小会影响引力系数达到峰值的频率,考虑到本文中蚂蚁种群规模为 50,拟定 λ 取值为 10 ~ 40。

在 $\alpha = 1, \rho = 0.1$ 的基础上,分别取 $\beta = 3, \delta = 6$ 和 $\beta = 3.5, \delta = 5.5$ 两种组合,参数 λ 以 1 的间隔由 10~40 递增做统计实验,次数为 1 000 次,结果如图 8 所示。

λ 的大小会影响引力系数达到峰值的频率, λ 越小,引力系数达到峰值所需 $k_{\min}$ 的值越小,节点选择中次优节点被选中的频率越高。由曲线规律可知, λ 的取值也要适中。 λ 太小,次优节点被选中的频率过高,不利于算法解的质量的提升,收敛均值与最优解次数的表现明显变差; λ 太大,次优节点被选中的频率过低,不能很好的发挥引力系数的作用,算法性能变差。

由图 8 结果可得, $\beta = 3.5, \delta = 5.5$ 条件下, $\lambda = 23$ 时收敛均值和最优解次数均为最优; $\beta = 3, \delta = 6$ 条件下, $\lambda = 20$ 时最优次次数最多,收敛均值仅比最优值高 0.043。

综上所述,当前求解效果最优的两组参数为 $\beta = 3, \delta = 6, \lambda = 20$ 和 $\beta = 3.5, \delta = 5.5, \lambda = 23$ 。其中,前者的效果略好于后者,均作后续分析。

两种参数组合对应的引力系数的变化规律如图 9 所示。

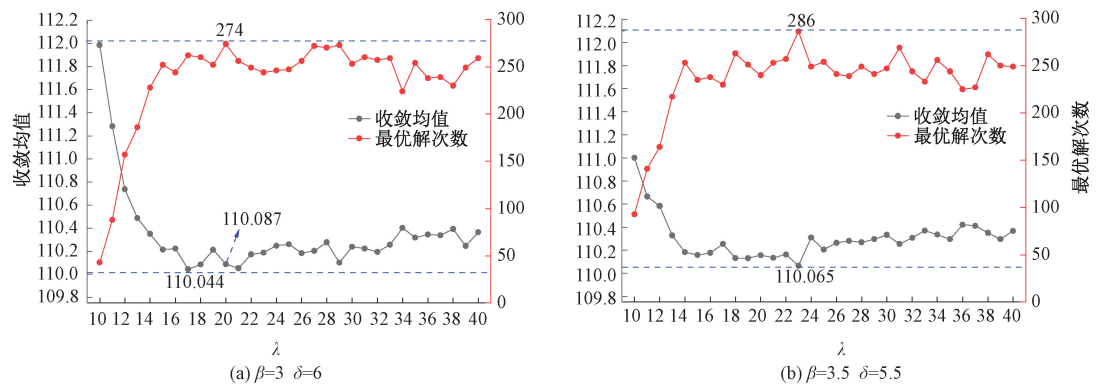


图 8 参数 λ 对性能的影响曲线
Fig. 8 Parameter λ influence curve on performance

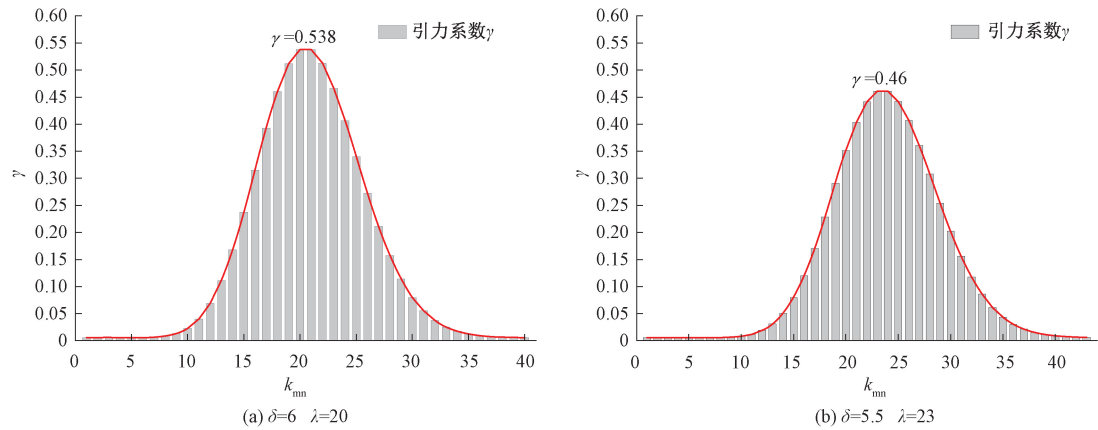


图 9 引力系数分布
Fig. 9 Gravitational coefficient distribution

(3) 对 $\tau_{\min}$ 的取值
 $\tau_{\min}$ 本身作为最小信息素浓度,其值不宜过大;同时也不宜过小,否则作用不大,考虑到初始信息素浓度 $T_0 = 5$,拟定 $\tau_{\min}$ 的取值为 0.5 ~ 1.5。

以 $\beta = 3, \delta = 6, \lambda = 20$ 和 $\beta = 3.5, \delta = 5.5, \lambda = 23$ 两种组合为基础,参数 $\tau_{\min}$ 以 0.1 的间隔由 0.5 ~ 1.5 递增做统计实验,每种参数组合的统计实验次数为 1 000 次,参数 $\tau_{\min}$ 对算法性能的影响曲线如图 10 所示。

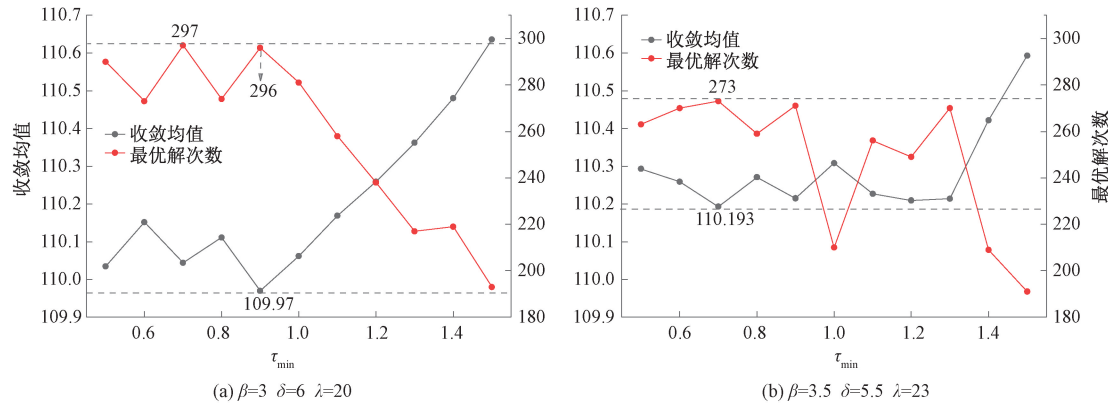


图 10 参数 $\tau_{\min}$ 对性能的影响曲线
Fig. 10 Parameter $\tau_{\min}$ influence curve on performance

由图 10 可知,在 $\beta = 3, \delta = 6, \lambda = 20$ 的基础上, $\tau_{\min} = 0.9$ 时收敛均值最低,最优解次数与最好值只差 1 次;在

$\beta = 3.5, \delta = 5.5, \lambda = 23$ 的基础上, $\tau_{\min} = 0.7$ 时两种性能参数均为最优。但前者在收敛均值和最优次次数上均明

显好于后者,所以选择前者为最终参数。

综上所述,对于本文提出的改进蚁群算法,其最佳参数组合为: $\alpha=1,\beta=3,\rho=0.1,\delta=6,\lambda=20,\tau_{\min}=0.9,Q=100,T_0=5$ 。种群规模设为 50,迭代次数设为 100。

5 UUV 集群任务规划的改进蚁群优化算法性能与功能验证

5.1 算法性能评价指标

对解决 UUV 集群勘察任务规划问题的算法的性能评价主要从收敛性能和解的质量两个方面进行。

当优化目标为总距离最短时,对于任务规划问题解的质量,公认的是规划方案中的各条路径之间无交叉无重叠即为一个近似最优解。

本文从如下 5 点来评价算法的性能:1)统计实验中的最优解(即 UUV 集群勘察任务规划总航程)及任务规划方案图;2)统计实验中的最优解次的收敛性能(算法收敛到最优值时需要的迭代次数)与各解次平均收敛性能;3)统计实验的平均结果,即 UUV 集群勘察任务规划平均总航程;4)统计实验中近似最优解出现的次数;5)统计实验算法结果中参与勘察任务的 UUV 对续航力和负载能力的约束满足情况。

5.2 与经典文献算法性能的案例对比分析与验证

1)验证案例 1

使用文献[20]的案例对基本蚁群算法、精英蚁群算法、文献中算法以及本文提出的改进蚁群算法进行对比。

(1)问题描述

同 4.1 节:案例问题描述(仿真案例 1)。

(2)案例 1 本文算法的运行结果

改进蚁群算法的 $\alpha=1,\beta=3,\rho=0.1,\delta=6,\lambda=20,\tau_{\min}=0.9,Q=100,T_0=5$ 。种群规模设为 50,迭代次数设为 100。

求得 UUV 集群勘察任务规划方案的路径最优解为 107.84 km,参与勘察的 UUV 数量为 4 艘。

集群勘察任务规划路径如图 11 所示,具体方案如下(0 代表基地):

- 0 → 18 → 0
- 0 → 20 → 11 → 17 → 3 → 4 → 0
- 0 → 8 → 19 → 15 → 16 → 13 → 6 → 0
- 0 → 1 → 7 → 10 → 9 → 12 → 2 → 14 → 5 → 0

由 UUV 集群勘察任务规划方案对负载与续航配置能力的约束满足能力如表 4 所示,最佳规划方案中参与勘察任务的 UUV 数量、每个 UUV 所使用的续航与负载符合文献[20]案例 1 中最多使用 5 艘 UUV,单体 UUV 最大续航距离为 50 km 以及最大负载为 8 的约束条件。

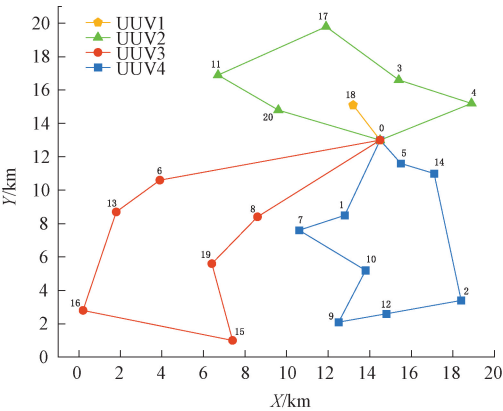


图 11 集群勘察任务规划路径

Fig. 11 Path map of cluster survey task planning

表 4 规划方案对能力约束的满足

Table 4 Table of capacity constraints satisfied by planning scheme

ID	已用续航/km	已用负载	ID	已用续航/km	已用负载
1	4.94	1.6	3	42.98	7.7
2	28.19	6.6	4	31.73	7.8

(3)算法求解质量对比

分别使用基本蚁群算法,精英蚁群算法以及本文提出的改进蚁群算法对案例 1 求解 50 次统计实验,结果如表 5 所示。

表 5 算法性能对比

Table 5 Algorithm performance comparison table

算法名称	航程均值/km	最优解次数	近似最优解次数
基本蚁群算法	113.794	2	8
精英蚁群算法	112.512	3	14
改进蚁群算法	109.656	15	47

从统计结果可知,本文提出的改进蚁群算法相较于基本蚁群算法和精英蚁群算法,50 次算法运行结果中航程平均值分别降低 4.138 和 2.856 km,找到最优解的比例分别提升 26% 和 24%,找到近似最优解的比例分别提升 78% 和 66%。50 次统计实验的结果如图 12 所示。

(4)算法收敛性能对比

分别选取 3 种算法 50 次运行中各自的最好结果进行收敛性能的对比如图 13 所示,从图 13 可知,基本蚁群算法在第 17 代时寻得最优解 107.84 km,但是因其信息素更新方式不能明显增加路径间信息素浓度的差异,算法最终并未收敛;精英蚁群算法在第 54 代收敛;本文提出的改进蚁群算法在第 27 代时实现收敛,平均 40 代实现收敛。

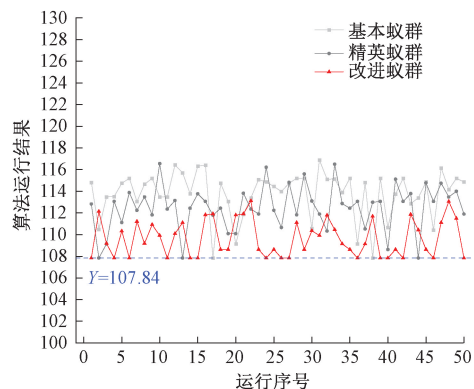


图 12 运行结果对比

Fig. 12 Operation result comparison chart

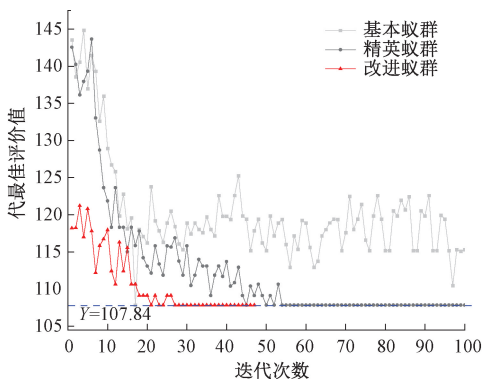


图 13 算法收敛对比

Fig. 13 Algorithm convergence comparison chart

(5) 与文献[20]算法运行结果对比

针对案例 1,使用本文提出的改进蚁群算法运行 10 次,统计结果如表 6 所示,其中 D_t 表示路径总距离。

表 6 案例 1 本文算法统计实验结果

Table 6 Statistical experimental results of the algorithm in this paper under the example 1

运行次序	D_t/km	运行次序	D_t/km
1	107.840 3	6	107.840 3
2	107.840 3	7	107.840 3
3	111.913 5	8	107.840 3
4	109.193 2	9	110.200 1
5	107.840 3	10	108.623 4

由表 6 可知,本文提出的改进算法运行 10 次平均结果为 108.691 8 km,结果包含最优解 107.840 3 km 共计 6 次,近似最优解共计 10 次。

相比于文献[20]算法运行 10 次所得的平均距离 110.308 3 km,本文提出算法的平均距离、最优解统计次数等项指标优于该文献。

2) 验证案例 2

引用文献[30]的案例并与其提出的改进蚁群算法的运行结果进行对比。

(1) 问题描述

已知某基地坐标为(0 km,0 km),作业区域中待勘察的任务点共有 19 个。UUV 的最大工作负载为 9,对参与勘察任务的 UUV 的数量和 UUV 最大续航并未做约束。任务点坐标和负载需求量如表 7 所示。

表 7 任务点信息

Table 7 Mission point information sheet

ID	X/km	Y/km	q	ID	X/km	Y/km	q
1	0	-1	1.5	11	3	4	0.2
2	0	3	1.8	12	-3	0	2.4
3	-2	-2	2.0	13	2	0	1.9
4	-3	-3	0.8	14	1	-3	2.0
5	3	-1	1.5	15	2	-1	0.7
6	-4	0	1.0	16	2	1	0.5
7	-4	-1	2.5	17	1	-4	2.2
8	1	-2	3.0	18	-3	2	3.1
9	1	-1	1.7	19	-1	-1	0.1
10	1	3	0.6				

(2) 案例 2 本文算法的运行结果

使用本文提出的改进蚁群算法求解该案例,改进蚁群算法的参数同前。

求得 UUV 集群勘察任务规划方案的路径最优解为 42.107 7 km,参与勘察的 UUV 数量为 4 艘。

集群勘察任务规划路径如图 14 所示,具体方案如下(0 代表基地):

0 → 9 → 15 → 5 → 13 → 16 → 11 → 10 → 2 → 0
0 → 12 → 6 → 7 → 4 → 3 → 19 → 0
0 → 8 → 14 → 17 → 1 → 0
0 → 18 → 0

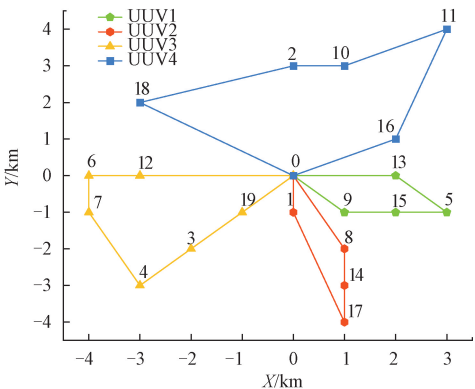


图 14 集群勘察任务规划路径

Fig. 14 Path map of cluster survey task planning

由 UUV 集群勘察任务规划方案对负载与续航配置能力的约束满足能力如表 8 所示,最佳规划方案中参与勘察任务的 UUV 数量、每个 UUV 所用的续航与负载符合文献[30]案例 2 中的约束条件。

表 8 规划方案对能力约束的满足

Table 8 Table of capacity constraints satisfied by planning scheme

ID	已用续航/km	已用负载	ID	已用续航/km	已用负载
1	6.83	5.8	3	11.48	8.8
2	8.40	8.7	4	15.40	6.2

(3)与文献[30]算法运行结果对比

针对案例 2,使用本文提出的改进蚁群算法运行 10 次,统计结果如表 9 所示,其中 D_t 表示路径总距离。

表 9 案例 2 本文算法统计实验结果

Table 9 Statistical experimental results of the algorithm in this paper under the example 2

运行次序	D_t /km	运行次序	D_t /km
1	42.314 9	6	42.314 9
2	42.181 5	7	42.246 7
3	42.314 9	8	42.107 7
4	42.107 7	9	42.107 7
5	42.314 9	10	42.246 7

使用本文提出的改进蚁群算法运行 10 次的平均结果为 42.225 8 km,较文献[30]的 10 次运行平均值 47.684 4 km 有明显的提高;且本文提出的改进蚁群算法得到的最优规划方案中参与勘察任务的 UUV 为 4 艘,最优结果为 42.107 7 km,这一结果明显优于对比文献[30]最优路径使用 5 艘 UUV 和最优值为 46.722 6 km 的结果。

5.3 UUV 集群任务规划算法的功能验证案例设计与分析(案例 3)

1)问题描述

已知作业区域中有广泛分布待勘察的任务点共计 35 个,某基地可参与勘察任务的 UUV 共有 15 艘,UUV 的最大续航为 20 km,UUV 最大工作负载为 3,基地坐标为(12.0 km,12.0 km),任务点坐标和负载需求量如表 10 所示。

2)案例 3 本文算法的运行结果改进蚁群算法的参数同前。

求得 UUV 集群勘察任务规划方案的路径最优解为 148.792 8 km,参与勘察任务的 UUV 共 10 艘。

表 10 任务点信息

Table 10 Mission point information sheet

ID	X/km	Y/km	q	ID	X/km	Y/km	q
1	14.8	9.90	1.4	19	18.5	13.7	0.3
2	16.0	7.60	0.3	20	18.0	6.70	0.3
3	10.5	7.30	0.2	21	10.3	9.00	0.8
4	3.00	11.7	0.4	22	5.70	14.2	0.7
5	6.90	15.0	0.6	23	6.00	10.0	1.0
6	14.6	9.00	1.5	24	9.00	17.3	0.3
7	13.8	16.9	0.9	25	10.8	18.1	1.2
8	19.9	12.6	0.7	26	16.7	17.1	0.9
9	18.0	12.0	0.1	27	16.6	9.90	0.5
10	16.0	11.1	0.6	28	11.9	8.10	0.4
11	12.4	9.10	1.3	29	17.2	19.1	1.6
12	4.00	12.5	0.5	30	15.0	19.0	0.6
13	12.9	19.3	0.3	31	19.3	15.4	0.8
14	9.90	15.9	1.2	32	7.60	9.50	0.4
15	8.80	9.50	1.5	33	16.2	15.0	0.8
16	7.20	13.1	1.0	34	13.9	8.00	0.6
17	8.00	17.0	0.7	35	8.10	14.6	0.6
18	18.2	9.00	0.4				

集群勘察任务规划路径如图 15 所示,具体方案如下(0 代表基地):

- 0 → 1 → 6 → 0
- 0 → 16 → 22 → 5 → 35 → 0
- 0 → 14 → 24 → 17 → 0
- 0 → 26 → 29 → 0
- 0 → 10 → 27 → 18 → 20 → 2 → 34 → 0
- 0 → 7 → 30 → 13 → 25 → 0
- 0 → 15 → 21 → 3 → 28 → 0
- 0 → 11 → 0
- 0 → 9 → 8 → 19 → 31 → 33 → 0
- 0 → 32 → 23 → 4 → 12 → 0

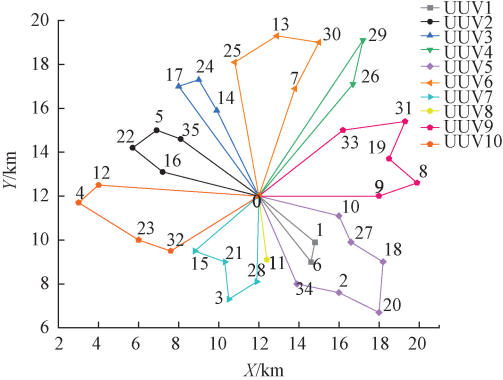


图 15 集群勘察任务规划路径

Fig. 15 Path map of cluster survey task planning

由 UUV 集群勘察任务规划方案对负载与续航配置能力的约束满足能力如表 11 所示,最佳规划方案中参与勘察任务的 UUV 数量、每个 UUV 所用的续航与负载均符合案例 3 中最多使用 15 艘 UUV,单体 UUV 最大续航里程 20 km 以及最大负载为 3 的约束条件。

表 11 规划方案对能力约束的满足

Table 11 Table of capacity constraints satisfied by planning scheme

ID	已用续航/km	已用负载	ID	已用续航/km	已用负载
1	8.39	2.9	6	18.40	2.5
2	14.18	2.7	7	12.87	2.9
3	13.54	2.2	8	5.85	1.3
4	17.80	2.3	9	19.94	2.7
5	18.35	2.7	10	19.48	2.1

运行 10 次算法做统计实验,所得统计实验结果如表 12 所示。

表 12 案例 3 本文算法统计实验结果

Table 12 Statistical experimental results of the algorithm in this paper under the example 3

运行次序	D_t /km	运行次序	D_t /km
1	148.792 8	6	148.792 8
2	149.009 9	7	149.660 9
3	148.792 8	8	149.009 9
4	149.009 9	9	149.660 9
5	149.228 1	10	148.792 8

10 次运行结果的平均值为 149.075 1 km,其中近似最优解的次数为 10 次,求得最优解 148.792 8 km 的次数为 4 次。

取统计实验中最佳次的每一代最佳评价价值,计算其收敛到最佳时需要的迭代次数,并与统计实验的平均值进行对比,收敛曲线对比如图 16 所示。

由图 16 可知,本文提出的改进蚁群算法在第 44 代时实现稳定收敛,收敛值为 148.792 8 km;取统计实验中的 10 次算法在每一代时评价值的平均值作平均收敛曲线,62 代时实现收敛,平均收敛值为 149.075 0 km。

从案例 3 的运行结果可以看出,当面对一定数量的勘察任务点数以及一定规模的 UUV 集群的勘察任务规划问题时,本文提出的改进蚁群算法仍然能在满足对个体 UUV 最大航行距离和最大负载的约束条件下保证解集的质量,统计实验中近似最优解出现的次数、最佳解的总航程、统计实验的平均航程以及对负载与续航配置能

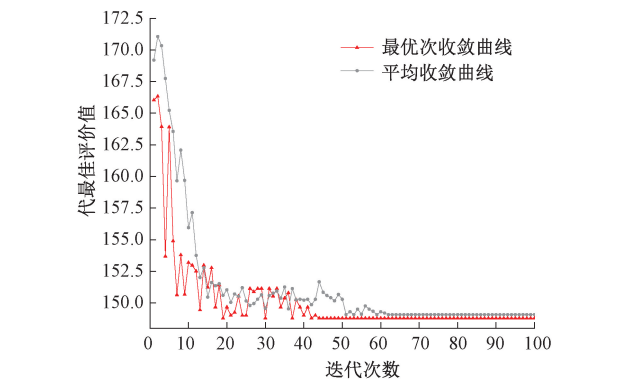


图 16 算法收敛

Fig. 16 Algorithm convergence graph

力的约束满足表可以验证这一点;同时,无论是统计实验中最佳解次实现稳定收敛时所需要的迭代次数还是统计实验实现收敛的平均代数都体现了本文算法优异的收敛性能。案例 3 验证了本文提出的算法在解决一定规模的 UUV 集群勘察任务规划问题时的有效性和快速性。

6 结 论

针对蚁群算法在求解 UUV 集群勘察任务规划问题中存在的全局寻优能力不强、收敛效率不高、易陷入局部最优的问题,本文提出了一种改进的蚁群算法。根据节点间距离设计了初始信息素浓度的不均等分配方式;提出最佳阈值与最差阈值优化了信息素更新规则;提出新的影响因子“引力系数 γ ”改进了状态转移公式。同时,对算法优化项的有效性进行了统计分析,对算法中相关参数的选取进行了细致的说明。通过与其他文献对比,验证了本文算法在收敛性能和解集质量上的优势;通过设计典型案例,验证了本文算法在解决一定规模的 UUV 集群在广泛且稀疏分布区域的集群勘察任务规划问题时的有效性和快速性。

参考文献

[1] 李航宇. 非预期事件下 UUV 容错自适应任务规划与决策方法研究[D]. 哈尔滨:哈尔滨工程大学,2016.
LI H Y. Research on UUV fault-tolerant adaptive task planning and decision-making method under unexpected events [D]. Harbin: Harbin Engineering University, 2016.

[2] 杨海智. 无人水下航行器群体协同任务规划方法研究[D]. 哈尔滨:哈尔滨工程大学,2015.
YANG H ZH. Research on group cooperative mission planning method for unmanned underwater vehicles [D]. Harbin: Harbin Engineering University, 2015.

[3] ZHANG W, ZHANG J, TENG Y, et al. The application

- of multiple ant colony algorithm to cooperative task allocation in UUVs swarm[C]. 2018 IEEE International Conference on Mechatronics and Automation (ICMA), 2018: 172-177.
- [4] KE Y S, WU R, GUO X, et al. Research on multi-objective task assignment scheme based on group isomorphism UUV [C]. 2021 IEEE 7th International Conference on Control Science and Systems Engineering (ICCSSE), 2021: 153-158.
- [5] WU X J, GAO Z Y, YUAN S, et al. A dynamic task allocation algorithm for heterogeneous UUV swarms[J]. Sensors, 2022, 22 (6): 2122.
- [6] XU Z, LI Y, FENG X. Constrained multi-objective task assignment for UUVs using multiple ant colonies system[C]. Isecs International Colloquium on Computing, Communication, Control, & Management, IEEE, 2008.
- [7] XIAO Y J, ZHANG D X. The command decision method of multiple UUV cooperative task assignment based on contract net protocol[J]. Journal of Systems Science & Information, 2016, 4(4): 379-390.
- [8] LV H L, WANG H J, LI Q, et al. Task allocation of multiple autonomous underwater vehicle system based on multi-objective optimization [C]. 2016 IEEE International Conference on Mechatronics and Automation, 2016.
- [9] 张松灿, 普杰信, 司彦娜, 等. 蚁群算法在移动机器人路径规划中的应用综述[J]. 计算机工程与应用, 2020, 56(8): 10-19.
- ZHANG S C, PU J X, SI Y N, et al. Survey on application of ant colony algorithm in path planning of mobile robot [J]. Computer Engineering and Applications, 2020, 56(8): 10-19.
- [10] DORIGO M, GAMBARDILLA L M. Ant colony system: A cooperative learning approach to the traveling salesman problem [J]. IEEE Transaction on Evolutionary Computation, 1997, 1(1): 53-66.
- [11] DORIGO M, GAMBARDILLA L M. A study of some properties of ant-Q [C]. International Conference on Parallel Problem Solving from Nature, 1996: 656-665.
- [12] LUO H, JIAN X L, LU W. Optimal test node selection based on dynamic ant colony algorithm for analog circuit[J]. Chinese Journal of Scientific Instrument, 2014, 35(10): 2231-2237.
- [13] JIANG M, WANG F, GE Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm[J]. Chinese Journal of Scientific Instrument, 2019, 40(2): 113-121.
- [14] 杨立伟, 付丽霞, 王倩, 等. 多层优化蚁群算法的移动机器人路径规划研究[J]. 电子测量与仪器学报, 2021, 35(9): 10-18.
- YANG L W, FU L X, WANG Q, et al. Multi-layer optimal ant colony algorithm for mobile robots path planning study [J]. Journal of Electronic Measurement and Instrumentation, 2021, 35(9): 10-18.
- [15] 张宏宏, 甘旭升, 李双峰, 等. 复杂低空环境下考虑区域风险评估的无人机航路规划[J]. 仪器仪表学报, 2021, 42(1): 257-266.
- ZHANG H H, GAN X F, LI SH F, et al. UAV route planning considering regional risk assessment under complex low altitude environment[J]. Chinese Journal of Scientific Instrument, 2021, 42(1): 257-266.
- [16] 李志锟, 黄宜庆, 徐玉琼. 改进变步长蚁群算法的移动机器人路径规划[J]. 电子测量与仪器学报, 2020, 34(8): 15-21.
- LI ZH K, HUANG Y Q, XU Y Q. Path planning of mobile robot based on improved variable step size ant colony algorithm[J]. Journal of Electronic Measurement and Instrumentation, 2020, 34(8): 15-21.
- [17] LUO Q, WANG H, ZHENG Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm [J]. Neural Computing and Applications, 2020, 32(6): 1555-1566.
- [18] CHEN X, YUAN Y. Novel ant colony optimization algorithm for robot path planning [J]. Systems Engineering and Electronics, 2008(5): 952-955.
- [19] STUTZLE T. Max-min ant system[J]. Future Generation Computer Systems, 2000(16): 889-914.
- [20] 张维泽, 林剑波, 吴洪森, 等. 基于改进蚁群算法的物流配送路径优化[J]. 浙江大学学报(工学版), 2008(4): 574-578, 597.
- ZHANG W Z, LIN J B, WU H S, et al. Optimizing logistic distribution routing problem based on improved ant colony algorithm[J]. Journal of Zhejiang University (Engineering Science), 2008(4): 574-578, 597.
- [21] WANG L, LUO C, LI M, et al. Trajectory planning of an autonomous mobile robot by evolving ant colony system[J]. International Journal of Robotics and Automation, 2017, 32(4): 406-413.
- [22] 吕金秋, 游晓明, 刘升. 机器人全局路径规划的混合蚁群系统算法[J]. 计算机工程与应用, 2016, 52(11): 38-43.
- LYU J Q, YOU X M, LIU SH. Hybrid ACS algorithm for robot global path planning [J]. Computer Engineering and Applications, 2016, 52(11): 38-43.
- [23] DORIGO M. The ant system: An autocatalytic optimizing

- process [C]. European Conference on Artificial Life, 1991.
- [24] 赵长鲜, 方木云. 基于贪心算法的物流配送系统的设计与实现[J]. 软件工程, 2020, 23(5): 21-23.
ZHAO CH X, FANG M Y. Design and implementation of the logistics distribution system based on greedy algorithm[J]. Software Engineering, 2020, 23(5): 21-23.
- [25] 朱庆保. 蚁群优化算法的收敛性分析[J]. 控制与决策, 2006(7): 763-766, 770.
ZHU Q B. Analysis of convergence of ant colony optimization algorithms [J]. Control and Decision, 2006(7): 763-766, 770.
- [26] DORIGO M, MANIEZZO V, COLORNI A. Ant system: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1996, 26(1): 29-41.
- [27] BULLNHEIMER B, HARTL R F, STRAUSS C. A new rank based version of the ant system—a computational study [J]. Central European Journal of Operations Research, 1999, 7(1): 25-38.
- [28] 张立峰, 张梦涵. 基于自适应模拟退火及 LM 联合反演算法的 ECT 图像重建[J]. 仪器仪表学报, 2021, 42(12): 228-235.
ZHANG L F, ZHANG M H. Image reconstruction for electrical capacitance tomography based on adaptive simulated annealing and LM joint inversion algorithm[J]. Chinese Journal of Scientific Instrument, 2021, 42(12): 228-235.
- [29] 叶志伟, 郑肇葆. 蚁群算法中参数 α 、 β 、 ρ 设置的研究——以 TSP 问题为例[J]. 武汉大学学报(信息科学版), 2004(7): 597-601.
YE ZH W, ZHENG ZH B. Configuration of parameters α , β , ρ in ant algorithm [J]. Geomatics and Information Science of Wuhan University, 2004(7): 597-601.
- [30] 王晓东, 张永强, 薛红. 基于改进蚁群算法对 VRP 线

路优化[J]. 吉林大学学报(信息科学版), 2017, 35(2): 198-203.

WANG X D, ZHANG Y Q, XUE H. Improved ant colony algorithm for VRP[J]. Journal of Jilin University (Information Science Edition), 2017, 35(2): 198-203.

作者简介



王宏健, 1993 年于黑龙江大学获得学士学位, 1998 年于哈尔滨工程大学获得硕士学位, 2004 年于哈尔滨工程大学获得博士学位, 2006 年于哈尔滨工业大学机械工程博士后流动站出站, 现为哈尔滨工程大学教授、博士生导师, 主要研究方向为无人潜航器自主智能与群体协同控制、智能优化理论和方法、机器学习、目标跟踪、同时定位与建图等。

E-mail: cctime99@163.com

Wang Hongjian received her B.Sc. degree from the Heilongjiang University in 1993, M.Sc. degree from Harbin Engineering University in 1998 and Ph.D. degree from Harbin Engineering University in 2004. From 2004 to 2006, she worked in the post-doctoral mobile station of Mechanical Engineering of Harbin Institute of Technology. She is currently a professor and a Ph.D. advisor at Harbin Engineering University. Her main research interests include autonomy control and swarm intelligence for UUV, intelligent optimization theory and methods, machine learning, target tracking, simultaneous localization and mapping (SLAM), etc.



鄂鑫(通信作者), 2020 年于合肥工业大学获得学士学位, 现为哈尔滨工程大学硕士研究生, 主要研究方向为无人水下航行器自主智能控制。

E-mail: exladida@163.com

E Xin (Corresponding author) received his B.Sc. degree from Hefei University of Technology in 2020. He is currently a M.Sc. candidate at Harbin Engineering University. His main research interests include autonomous intelligent control technology of UUV.