

# 基于改进粒子群算法的海上遇险目标搜寻方法

孔祥凤, 王海红, 李盛威, 黄伟

(青岛科技大学 信息科学技术学院, 山东 青岛 266061)

**摘要:** 针对海上遇险目标搜寻范围动态化、影响因素众多导致搜救成功率较低的问题, 提出了一种基于改进粒子群算法的海上遇险目标搜寻方法, 旨在寻找最佳搜寻路径, 提高海上遇险目标的搜救成功率; 该方法基于遇险目标的位置信息和搜寻资源参数, 构建海上遇险目标搜寻模型, 并采用余弦曲线自适应方法改进算法的惯性权重系数, 增强粒子群算法的初期全局搜索和后期局部搜索能力; 采用自适应策略调整加速度, 并保持其总和不变, 以避免搜索效率下降或不稳定; 引入扰动粒子更新机制来保持种群的多样性, 避免陷入局部最优; 将改进算法应用于实际搜寻问题, 验证了算法的有效性, 将改进算法与传统的粒子群算法和遗传算法进行对比, 结果表明, 改进算法较传统粒子群算法和遗传算法具有更高的搜救成功率。

**关键词:** 改进粒子群算法; 海上搜寻; 自适应; 全局最优; 惯性权重; 扰动粒子

## Searching Method for Maritime Distress Targets Based on Improved Particle Swarm Algorithm

KONG Xiangfeng, WANG Haihong, LI Shengwei, HUANG Wei

(College of Information Science and Technology, Qingdao University of Science and Technology,  
Qingdao 266061, China)

**Abstract:** In response to the problems of dynamic search ranges and numerous influencing factors, which leads to lower success rates in search and rescue operations for maritime distress targets, an improved method based on improved particle swarm algorithm is proposed. The method aims to identify the optimal search path to enhance the success rate of rescuing maritime distress targets. Based on the location information of distress targets and parameters of search resources, a searching maritime distress target model is constructed. A cosine curve adaptive method is used to improve the inertia weight coefficient, and enhance the initial global search and later local search capabilities of the particle swarm algorithm. Moreover, to avoid a decrease in search efficiency or instability, an adaptive strategy is employed to adjust acceleration while maintaining the total unchanged. The perturbation particle update mechanism is introduced to maintain population diversity and avoid local optimal solution. The improved algorithm is applied in practical search scenarios, which verifies the effectiveness of the proposed algorithm. The results show that the improved algorithm has a higher success rate in search and rescue operations compared to traditional particle swarm and genetic algorithms.

**Keywords:** improved particle swarm algorithm; maritime search; self adaption; global optimum; inertia weight; disturbance particle

## 0 引言

我国航海业的快速发展不仅带来了经济的繁荣, 同时也对海上遇险目标搜寻能力提出了严峻考验。为保障

公众的生命及财产安全, 制定合理的搜救计划、及时处理已发生事故、确保群众的生命及财产安全成为目前亟需解决的现实问题。由于时间紧迫、搜救范围广阔、海洋环境复杂, 海上遇险目标的搜寻被认为是搜救任务中

收稿日期: 2023-12-24; 修回日期: 2024-02-18。

基金项目: 国家自然科学基金项目(61104004, 61170258); 山东省自然科学基金重大基础研究项目(ZR2021ZD12)。

作者简介: 孔祥凤(1998-), 女, 硕士研究生。

通讯作者: 王海红(1972-), 女, 博士。

引用格式: 孔祥凤, 王海红, 李盛威, 等. 基于改进粒子群算法的海上遇险目标搜寻方法[J]. 计算机测量与控制, 2025, 33(3): 183

- 189.

最关键的环节。搜救成功率受到多种因素的影响,包括事发位置、事发海区的海洋环境、遇险目标类型以及搜救资源的航速等限制。传统的搜救方式已无法满足搜救任务所需的快速高效要求。因此,采用先进的科技手段和创新性的搜救方法,成为确保海上搜寻能力不断提升的关键所在。

为了满足快速高效的搜救要求,卫星监测、无人机搜救、自动化图像识别等现代技术方法得到了广泛应用。目前海上遇险目标搜寻的主要方法是通过各种漂移模型<sup>[1]</sup>预测出物体可能的漂移轨迹,根据具体的搜救任务、环境条件和可用资源,选择适合的搜索方式或者结合多种方式进行搜索,典型的搜寻方式有平行线搜寻<sup>[1]</sup>、横移线搜寻、扇形搜寻<sup>[2]</sup>等。然而,由于遇险目标在待搜寻区域中并非等概率分布,可能存在一些特定的潜在位置或更高的概率分布,这种情况下传统的搜寻方式成功概率较低。为了解决这个问题,通用的做法是收集和分析可用的信息来源,优化漂移模型<sup>[3]</sup>,采用数据驱动建模<sup>[4]</sup>等方法加以解决。从而更加有效地提高搜救任务的成功率,确保在复杂多变的海上环境中迅速、准确地找到遇险目标。

关于海上遇险目标搜救方法,国外的研究起步较早,早在 20 世纪初,德国、法国、英国等国家就开始基于现代化方法进行海上搜救体系的研究。一些先进的海上搜救系统被广泛采用和借鉴。其中,美国海上搜救最优规划系统(SAROPS)中的 PLANNER 模块在获取目标位置信息基础上,采用启发式方法为搜寻单位制定搜寻计划并确定搜寻方向。欧盟名为“共同交通监控和信息系统”(SafeSeaNet)的海上搜救规划系统,提供了全面的搜救信息管理和共享平台。加拿大搜救系统(CANSAR)采用 min/max 方法生成搜寻区域,提高了搜救成功率。挪威海岸警卫队使用的搜救任务管理系统 SARSYS<sup>[5]</sup>,整合了卫星监测数据、船舶自动识别系统(AIS)、雷达和通信设备等,提供实时的目标定位、路径规划和资源调度等功能。中国海事局主导开发的“海上应急处置指挥平台”(MERC),整合了国家海洋局的环境预报数据,提供漂移预测以及搜救辅助决策功能。此外,包括中国在内的多个国家参与了 INMARSAT Cospas-Sarsat 国际合作项目,利用卫星通信技术进行全球范围的搜救任务,可提供遇险目标的紧急信号监测、定位和跟踪、协助组织搜救行动。

随人工智能的发展,智能技术<sup>[6-10]</sup>在海上遇险目标搜寻中得到了广泛的应用。

其中,启发式优化算法是一种根据最少的相关信息在尽可能快的时间内提供解决方案的方法,在单目标优化问题上被广泛的应用。在这些算法中,粒子群算法具有优秀的竞争力,被认为是解决许多实际应用问题的优

秀候选算法,主要原因在于归因于其易于实现和快速的收敛速度。然而,在解决一些实际问题的过程中,粒子群算法通常面临从局部最优陷阱中脱离的困难。针对这个问题,文献[6]提出了随机漂移粒子群优化算法,其设计思路来自于金属导体中自由电子的运动模型,增强算法中粒子摆脱局部最优的能力,因此有着优异的搜索性能。维度学习粒子群算法<sup>[7]</sup>在粒子群算法中加入了维度学习机制,该机制在每次迭代后都让每个粒子最佳位置的每个维度向当前全局粒子最佳位置的对应维度学习,从而避免了粒子群算法存在的“振荡”和“两步前进、一步后退”的现象。遗传-粒子群算法<sup>[8]</sup>则直接将遗传算法的粒子更新方式和粒子群算法的粒子更新方式混合使用,适应值更好的粒子会采用遗传算法的粒子更新方式,而适应值较差的粒子会采用粒子群算法的粒子更新方式,以此保持种群的多样性。

综上,很多研究者致力于改进粒子群算法,避免陷入局部最优,并取得了成效<sup>[6-8]</sup>。但是仍然面对复杂动态环境和高维度解空间的挑战。在海上遇险目标搜索中,环境条件可能会经常变化,通常涉及高维度的解空间,传统粒子群算法在高维空间下的搜索效率和有效性有待提高。因此,本文的目的是设计一种改进的粒子群算法,增强粒子群算法全局探索能力、加强算法的动态适应性,并采用该算法寻找海上遇险目标最佳搜寻路径,算法引入变动的惯性权重,设计加速度系数为常数,调整粒子的收敛速度和搜索方向,提高粒子的寻优能力,加快算法的收敛速度。旨在加快海上搜寻的速度,提高海上搜救的成功率,为海上遇险目标搜寻提供策略支持。

## 1 海上遇险目标搜寻模型构建

采用海上搜救的成功率最大化作为优化目标,构建海上遇险目标搜寻模型,设计输入变量、决策变量、中间变量、目标函数以及约束条件。

### 1.1 输入变量设计

综合考虑待搜寻海域的环境状况(风、浪)以及遇险人员的实际情况(失事位置、人数、事故等级)结合搜寻资源自身的情况(位置、航速、扫海宽度、航线间距),提取到的有效信息如表 1 所示。

表 1 事故相关信息表

序号	信息
1	事故等级/参与组织
2	失事位置/人数
3	事故发生地风力等级/风向/海浪
4	搜救船舶的位置/航速/扫海宽度/航线间距
5	漂移预测模型生成的赛搜寻海域

根据搜寻优化的信息需求,设计采用以下输入变

量: 搜寻区域的子单元位置、搜寻区域子单元的包含概率、搜救船舶的个数、搜救船舶的最大航速、搜救船舶的扫海宽度。具体设计如表 2 所示。

表 2 输入变量符号及含义

类型	符号	符号含义
输入变量	$x_j$	搜寻子单元位置
	$p_{oj}$	子单元包含概率
	$B_i$	搜救船舶
	$V_i$	第 $i$ 艘搜救船舶的最大航速
	$W_i$	第 $i$ 艘搜救船舶的扫海宽度
	$R_i$	第 $i$ 艘船舶的航线间距

其中, 搜寻的有限区域位置集合, 即遇险目标可能存在的位置集合为  $X = \{x_j, 1 \leq j \leq N\}$ 。

### 1.2 决策变量设计

模型中求取的最终结果是搜救成功率最大的搜寻单元选取顺序, 因此选取模型的决策变量为搜寻单元  $x_j (j = 1, 2, 3, \dots, N)$ , 其中  $x_j$  表示船舶选择的搜寻单元, 设计如下:

$$x_j = \begin{cases} 1, & \text{如果船舶选择搜寻该单元} \\ 0, & \text{如果船舶不选择搜寻该单元} \end{cases} \quad (1)$$

### 1.3 中间变量设计

根据对海上目标搜寻已知信息的分析, 提前计算一些中间变量, 包括船舶的发现概率、子单元的搜救成功率以及船舶的累计搜救成功率, 从而计算最终的搜救成功率。

1) 第  $i$  艘船的探测到遇险目标的概率  $p_{di}$ :

随机探测模型中发现概率随覆盖率的增加而增大, 并且结合了目标动态漂移的特点, 更适应海上复杂的搜寻环境, 因此本文采取随机探测模型<sup>[9]</sup>。船舶探测到遇险目标的能力由下式表示:

$$p_{di} = 1 - e^{-W_i/R_i} \quad (2)$$

其中:  $p_{di}$  表示第  $i$  艘船的探测概率,  $W_i$  表示第  $i$  艘搜救船舶的扫海宽度,  $R_i$  表示第  $i$  艘船的航线间距。

2) 子单元  $j$  的搜救成功率  $Pos$  为此单元  $j$  包含概率  $p_{di}$  与搜寻该单元船舶  $i$  探测概率  $p_{oj}$  的乘积:

$$Pos = p_{di} \cdot p_{oj} \quad (3)$$

其中:  $p_{di}$  表示第  $i$  艘船的探测概率,  $p_{oj}$  表示第  $j$  个子单元的包含概率。

3) 子单元  $j$  更新的包含概率  $p'_{oj}$ : 船舶在子单元  $j$  未搜寻到遇险目标时, 包含概率更新为初始的包含概率  $p_{oj}$  与探测失败概率的乘积:

$$p'_{oj} = p_{oj} \cdot (1 - p_{di}) \quad (4)$$

其中:  $p_{di}$  表示船舶  $i$  的探测概率,  $p_{oj}$  表示子单元  $j$  的包含概率。

4) 当在子单元  $j$  搜寻到遇险目标时, 搜救成功率

$Pos_i$  由式 (5) 计算。当未搜寻到时, 子单元  $j$  的包含概率更新为  $p'_{oj}$ , 因此第二次搜寻的搜救成功率为第一次搜寻失败的概率与子单元更新的包含概率的乘积, 即式 (6):

$$Pos_i = p_{di} \cdot p_{oj} \quad (5)$$

$$Pos'_i = p'_{oj} \cdot (1 - Pos_i) \quad (6)$$

其中:  $p_{di}$  表示艘船  $i$  的探测概率,  $p_{oj}$  表示子单元  $j$  的包含概率,  $Pos'_i$  表示第二次搜寻的搜救成功率,  $p'_{oj}$  表示子单元  $j$  更新的包含概率,  $Pos_i$  表示船舶  $i$  的搜救成功率。

### 1.4 目标函数设计

以最大化搜救成功率  $Pos_c$  作为最终设计目标, 设计如下:

$$Pos_c = \sum_i^M Pos_i \quad (7)$$

### 1.5 约束条件设计

在建立海上遇险目标搜寻优化模型时, 必须考虑实际环境情况和可用的搜救资源情况。其算法设计需要满足一定的约束条件, 以确保算法在实际应用中的可行性和有效性, 具体约束如下。

1) 搜救资源的数量小于等于能够在相应搜寻时间内完成搜寻任务的能力最弱搜救资源的数量:

$$\sum_{i=1}^M B_i \leq n \quad (8)$$

其中:  $B_i$  表示第  $i$  个搜救资源,  $n$  表示完成搜寻任务全部采用搜寻能力最弱资源的数量。

2) 搜救资源到达待搜寻区域的时间要小于总搜寻时间:

$$T_i^0 \leq T_{\max} \quad (9)$$

其中:  $T_i^0$  表示第  $i$  艘船到达搜寻区域的时间,  $T_{\max}$  表示预估的总搜寻时间。

根据上述模型, 本设计的目的是在满足合理的搜救资源数量式 (8), 尽可能少地搜寻时间式 (9) 条件下实现累计搜救成功率最大化式 (7), 旨在提高海上搜救成功率。为了优化海上搜寻的成功率及时间, 引入改进粒子群算法, 实现优化目标。

## 2 粒子群算法的改进

传统粒子群算法在一些问题上表现优异, 但仍存在早熟收敛、参数敏感性高、高维问题及多模态限制等挑战。为了增强算法在初期的全局搜索和后期的局部搜索、收敛能力, 惯性权重系数采用余弦曲线自适应的策略, 通过动态调整惯性权重来平衡全局搜索和局部搜索的权衡。加速度系数采用随着迭代次数的增加总和为 1 的调整策略, 以逐渐增加局部搜索的影响力。同时, 引入扰动粒子更新机制, 通过扩大粒子的搜索范围来改善算法的勘探能力, 使其更好地探索可能的解空间。

## 2.1 惯性权重系数更新设计

在传统的粒子群算法中，惯性权重系数更新采用线性自适应方式进行控制。然而这种方式存在一定的不足，具体表现在惯性权重系数初期下降较快，导致了前期勘探不足，无法搜索到全局最优；惯性权重系数在后期更新速度太慢，会导致算法后期不够发达，以致算法无法在局部区域准确搜索。

为了克服上述问题，采用了余弦曲线自适应策略替代线性自适应策略。这两种策略的不同之处是惯性权重系数的递减趋势不同。线性自适应是以相同的速度减少，余弦曲线自适应策略在下降的初始阶段和结束阶段速度较慢，有助于增强粒子在开始阶段的全局搜索能力和后期的局部搜索能力。而且，在早期余弦自适应策略中惯性权重系数的值大于线性自适应的值，可以扩大搜索范围，增强粒子的探测能力；后期余弦自适应策略中惯性权重的值小于线性自适应的值，可以缩小粒子搜索范围，提高粒子的探测能力。具体设计方法如下。

惯性权重系数表示为：

$$\omega = \left[ \cos\left(\pi \times \frac{K}{K_{\max}}\right) + \omega_{\max} \right] \times \frac{\omega_{\max} + \omega_{\min}}{2} \quad (10)$$

其中： $K$  代表当前迭代步数； $K_{\max}$  是最大迭代次数； $\omega$  是惯性权重系数； $\omega_{\max}$  是最大惯性权重系数； $\omega_{\min}$  是最小惯性权重系数。

## 2.2 加速度系数更新设计

在粒子群算法的应用中，加速度系数  $c_1$  和  $c_2$  对调整收敛速度和搜索方向起到了至关重要的作用，能够灵活地控制粒子的最大学习步长。由于传统的粒子群算法搜索过程中的随机性较高，精确分析加速度系数之间的定量关系变得非常困难<sup>[10]</sup>。因此，本文将应用于海上遇险目标的粒子群算法中的加速度系数的和设置为一个固定的常数。并且，为提升算法的性能，重新设计加速度系数的更新机制。在算法的前半阶段，搜索过程占主导地位，粒子积极地探索搜寻空间，获取信息；而在算法的后半阶段，收敛过程占主导地位，粒子会逐渐聚集在全局最优解附近。采用式 (11) (12) 完成自适应加速度系数更新：

$$c'_1 = c_{\max} - \frac{(c_{\max} - c_{\min})K}{K_{\max}} \quad (11)$$

$$c'_2 = c_{\min} + \frac{(c_{\max} - c_{\min})K}{K_{\max}} \quad (12)$$

其中： $c'_1$  随时间推移是逐渐减小， $c'_2$  随时间推移逐渐增大，它们的总和保持不变，这意味着粒子群体的探索和收敛能力是固定的。在算法的搜索阶段  $c'_1 > c'_2$ ，增强了个体粒子对路径节点的探索能力，从而有效地避免了算法陷入局部最优的困境；而在算法的收敛阶段  $c'_1 < c'_2$ ，粒子群体的收敛效果得到提升，从而使粒子

群体快速达到全局最优。

## 2.3 扰动粒子更新机制

对传统粒子群算法的速度更新公式进行了改进，如式 (13)，引入了扰动项  $c_3 r_3 (G_{mm}^k - X_{mm}^k)$ ，这样粒子在飞行方向上的可能性变得更大，从而扩大了粒子的搜索范围。通过选择与当前粒子存在非支配关系甚至支配当前粒子的个体作为扰动粒子，提升粒子群的探索能力，使其能够在更广泛的范围内进行搜索：

$$V_{mm}^{k+1} = \omega V_{mm}^k + c_1 r_1 (P_{mm}^k - X_{mm}^k) + c_2 r_2 (G_{mm}^k - X_{mm}^k) + c_3 r_3 (G_{mm}^k - X_{mm}^k) \quad (13)$$

其中： $V_{mm}^k$  是在第  $K$  代中的第  $m$  ( $m = 1, 2, \dots, M$ ) 个粒子的第  $n$  ( $n = 1, 2, \dots, N$ ) 个维度的速度； $X_{mm}^k$  是在第  $K$  代中的第  $m$  ( $m = 1, 2, \dots, M$ ) 个粒子的第  $n$  ( $n = 1, 2, \dots, N$ ) 个维度的位置； $P_{mm}^k$  是个体认识最佳解； $G_{mm}^k$  是全局认识最佳解； $\omega$  是惯性权重系数； $c_1$ 、 $c_2$  为加速度系数，用以调节粒子学习的最大步长； $c_3$  为扰动因子； $r_1$  和  $r_2$  是  $[0, 1]$  内的随机值，以增加搜索随机性； $r_3$  为  $[-1, 1]$  区间内均匀分布的随机数。

改进后，在粒子群算法的搜索过程中，引入了更为多样性和具有竞争优势的粒子，从而增强了群体对问题空间的覆盖。通过选择非支配关系的个体作为扰动粒子，确保了扰动粒子不仅具有较高的适应性，同时具备了对当前粒子的一定竞争力。这种引入的多样性有助于克服传统粒子群算法容易陷入局部最优的弊端，使得算法在搜索过程中更具有全局探索的能力。

## 2.4 改进粒子群算法的海上遇险目标搜寻优化中的应用

利用改进的粒子群算法对海上遇险目标搜寻优化模型进行求解，得到搜寻子单元的顺序列表，进而得出海上遇险目标搜寻优化的具体方案。

具体的算法求解流程如下所述：

1) 初始化海上遇险目标搜寻优化模型，根据遇险信息，初始化  $x_j$  全部为 0， $p_j$  按照划分子区域依次输入数值，根据搜救船舶的具体参数更新  $B_i$ 、 $V_i$ 、 $W_i$ 、 $R_i$ 。详细见第 1 章。设置改进粒子群算法的粒子数量为  $[50, 500]$ ，设计粒子子单元的区域编号  $x_j$ 。

2) 采用式 (5) 作为改进粒子群算法的适应度函数。将初始  $Pos_i$  作为当前粒子的局部最优值、位置作为当前的局部最优所在的位置。将所有粒子中最高的初始  $Pos_i$  作为当前全局最优值，并将其作为当前的全局最优位置。

3) 设计粒子运动为自适应更新策略。 $\omega_{\max}$  设置为 0.9， $\omega_{\min}$  设置为 0.4， $c_{\max}$  设置为 1.5， $c_{\min}$  设置为 0.5。采用公式 (10) 更新惯性权重系数，采用式 (11) (12) 更新加速度系数，采用式 (13) 更新粒子速度，粒子位置更新表达式为：

表 3 搜寻方案对比

时间段	算法	搜寻方案	累计 POS/%
1:00~3:00	GA	A <sub>14</sub> A <sub>24</sub> A <sub>34</sub> A <sub>35</sub> A <sub>25</sub> A <sub>15</sub> A <sub>16</sub> A <sub>26</sub> A <sub>36</sub> A <sub>46</sub> A <sub>45</sub> A <sub>44</sub> A <sub>43</sub> A <sub>33</sub> A <sub>32</sub> A <sub>22</sub> A <sub>23</sub> A <sub>13</sub> A <sub>12</sub> A <sub>11</sub> A <sub>21</sub> A <sub>31</sub> A <sub>41</sub> A <sub>42</sub> A <sub>52</sub> A <sub>53</sub> A <sub>54</sub> A <sub>55</sub> A <sub>56</sub> A <sub>51</sub>	86.18
	PSO	A <sub>14</sub> A <sub>24</sub> A <sub>34</sub> A <sub>44</sub> A <sub>15</sub> A <sub>35</sub> A <sub>25</sub> A <sub>15</sub> A <sub>16</sub> A <sub>26</sub> A <sub>36</sub> A <sub>46</sub> A <sub>56</sub> A <sub>55</sub> A <sub>54</sub> A <sub>53</sub> A <sub>43</sub> A <sub>33</sub> A <sub>23</sub> A <sub>13</sub> A <sub>12</sub> A <sub>22</sub> A <sub>32</sub> A <sub>42</sub> A <sub>52</sub> A <sub>51</sub> A <sub>41</sub> A <sub>31</sub> A <sub>21</sub> A <sub>11</sub>	86.32
	改进 PSO	A <sub>14</sub> A <sub>24</sub> A <sub>34</sub> A <sub>33</sub> A <sub>32</sub> A <sub>22</sub> A <sub>23</sub> A <sub>13</sub> A <sub>12</sub> A <sub>11</sub> A <sub>21</sub> A <sub>31</sub> A <sub>41</sub> A <sub>42</sub> A <sub>43</sub> A <sub>44</sub> A <sub>45</sub> A <sub>35</sub> A <sub>25</sub> A <sub>15</sub> A <sub>16</sub> A <sub>26</sub> A <sub>36</sub> A <sub>46</sub> A <sub>56</sub> A <sub>55</sub> A <sub>54</sub> A <sub>53</sub> A <sub>52</sub> A <sub>51</sub>	87.68
3:00~5:00	GA	A <sub>36</sub> A <sub>35</sub> A <sub>34</sub> A <sub>33</sub> A <sub>32</sub> A <sub>22</sub> A <sub>23</sub> A <sub>24</sub> A <sub>25</sub> A <sub>26</sub> A <sub>16</sub> A <sub>15</sub> A <sub>14</sub> A <sub>13</sub> A <sub>12</sub> A <sub>11</sub> A <sub>21</sub> A <sub>31</sub> A <sub>41</sub> A <sub>42</sub> A <sub>43</sub> A <sub>44</sub> A <sub>45</sub> A <sub>46</sub> A <sub>56</sub> A <sub>55</sub> A <sub>54</sub> A <sub>53</sub> A <sub>52</sub> A <sub>51</sub>	85.29
	PSO	A <sub>14</sub> A <sub>24</sub> A <sub>34</sub> A <sub>33</sub> A <sub>32</sub> A <sub>22</sub> A <sub>23</sub> A <sub>13</sub> A <sub>12</sub> A <sub>11</sub> A <sub>21</sub> A <sub>31</sub> A <sub>41</sub> A <sub>42</sub> A <sub>43</sub> A <sub>44</sub> A <sub>45</sub> A <sub>35</sub> A <sub>25</sub> A <sub>15</sub> A <sub>16</sub> A <sub>26</sub> A <sub>36</sub> A <sub>46</sub> A <sub>56</sub> A <sub>55</sub> A <sub>54</sub> A <sub>53</sub> A <sub>52</sub> A <sub>51</sub>	85.33
	改进 PSO	A <sub>14</sub> A <sub>24</sub> A <sub>34</sub> A <sub>44</sub> A <sub>15</sub> A <sub>35</sub> A <sub>25</sub> A <sub>15</sub> A <sub>16</sub> A <sub>26</sub> A <sub>36</sub> A <sub>46</sub> A <sub>56</sub> A <sub>55</sub> A <sub>54</sub> A <sub>53</sub> A <sub>43</sub> A <sub>33</sub> A <sub>23</sub> A <sub>22</sub> A <sub>32</sub> A <sub>42</sub> A <sub>52</sub> A <sub>51</sub> A <sub>41</sub> A <sub>31</sub> A <sub>21</sub> A <sub>11</sub> A <sub>12</sub> A <sub>13</sub>	86.58
5:00~7:00	GA	A <sub>53</sub> A <sub>43</sub> A <sub>33</sub> A <sub>32</sub> A <sub>22</sub> A <sub>23</sub> A <sub>24</sub> A <sub>25</sub> A <sub>35</sub> A <sub>34</sub> A <sub>44</sub> A <sub>45</sub> A <sub>46</sub> A <sub>36</sub> A <sub>26</sub> A <sub>16</sub> A <sub>15</sub> A <sub>14</sub> A <sub>13</sub> A <sub>12</sub> A <sub>11</sub> A <sub>21</sub> A <sub>31</sub> A <sub>41</sub> A <sub>42</sub> A <sub>52</sub> A <sub>54</sub> A <sub>55</sub> A <sub>56</sub> A <sub>51</sub>	85.12
	PSO	A <sub>41</sub> A <sub>42</sub> A <sub>43</sub> A <sub>44</sub> A <sub>34</sub> A <sub>33</sub> A <sub>32</sub> A <sub>22</sub> A <sub>23</sub> A <sub>24</sub> A <sub>25</sub> A <sub>35</sub> A <sub>45</sub> A <sub>46</sub> A <sub>36</sub> A <sub>26</sub> A <sub>16</sub> A <sub>15</sub> A <sub>14</sub> A <sub>13</sub> A <sub>12</sub> A <sub>11</sub> A <sub>21</sub> A <sub>31</sub> A <sub>51</sub> A <sub>52</sub> A <sub>53</sub> A <sub>54</sub> A <sub>55</sub> A <sub>56</sub>	84.33
	改进 PSO	A <sub>41</sub> A <sub>42</sub> A <sub>43</sub> A <sub>33</sub> A <sub>34</sub> A <sub>35</sub> A <sub>25</sub> A <sub>24</sub> A <sub>23</sub> A <sub>22</sub> A <sub>32</sub> A <sub>31</sub> A <sub>21</sub> A <sub>11</sub> A <sub>12</sub> A <sub>13</sub> A <sub>14</sub> A <sub>15</sub> A <sub>16</sub> A <sub>26</sub> A <sub>36</sub> A <sub>46</sub> A <sub>45</sub> A <sub>44</sub> A <sub>54</sub> A <sub>55</sub> A <sub>56</sub> A <sub>52</sub> A <sub>51</sub> A <sub>53</sub>	84.44

$$X_{mm}^{k+1} = X_{mm}^k + V_{mm}^{k+1} \quad (14)$$

4) 评估所有粒子的适应度值, 与目标函数 (7) 进行对比, 如果适应度值小于平均适应度值, 表示粒子距离最大值较远, 需扩大搜索范围来寻找最值, 此时位置更新变化会更大, 根据式 (14) 进行位置的更新。如果适应度值大于平均适应度值, 表示粒子距离最大值较近, 需缩小搜索范围进行局部精确搜索, 需要通过加快速度实现, 即式 (13) 进行更新。然后更新  $P_{mm}$  和  $G_{mm}$ 。

5) 对整个算法空间进行迭代搜索, 直达到最大迭代次数, 得到搜救成功率最大的子单元搜寻顺序  $\{x_a, x_b, x_c, \dots\}$ , 该搜寻方案为最终解。

6) 将算法空间映射回问题空间, 获取船舶搜寻任务区域编号的顺序列表, 如表 3 所示。

算法具体流程如图 1 所示。

### 3 系仿真及结果分析

以一次青岛市海上搜救中心接到的报警事件为例, 对本文提出的算法进行示例研究。2022 年 12 月 24 日, 杂货船“浙玉 XXX”轮因不明原因船舱进水, 船上共 6 人, 位置 123°53'E/35°46'N, 请求救助。根据以上信息, 进行海上遇险目标的搜寻方案的生成。为了方便对比算法的有效性, 假设指定某搜救船舶。船舶性能参数如表 4 所示。

表 4 船舶性能参数表

船舶编号	船舶当前位置	最大航速	扫海宽度	航线间距
1	123°29'E/36°22'N	8.6 节	20 节	0.2 nmile

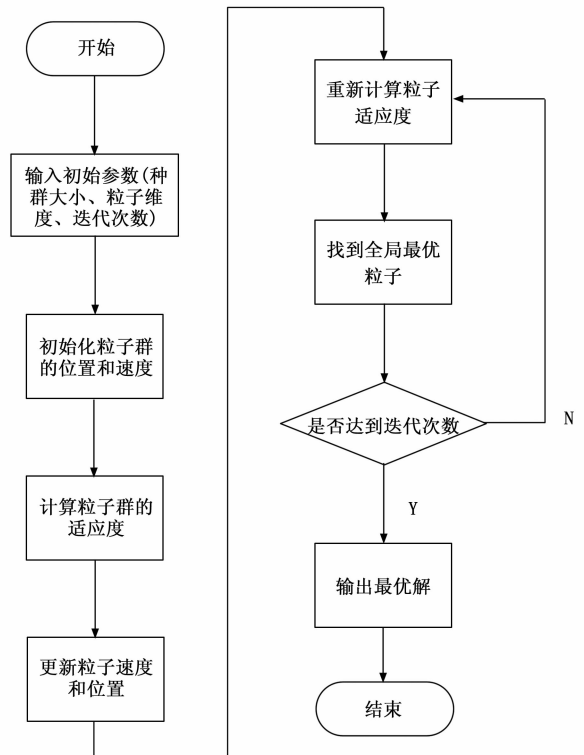


图 1 改进粒子群算法流程图

采用基于蒙特卡洛随机粒子法<sup>[1]</sup>漂流模拟得到的海上遇险目标的区域概率分布图, 即先验信息已知, 每个栅格具有一定的初始包含概率值。本文假定待搜寻海域 100% 包含遇险目标。待搜寻区域的总体面积为 1 200 平方千米。

仿真基于 Windows10 操作系统，软件平台为 MatlabR2019a。采用如图 2 所示的搜救区域概率图方法进行实验，对改进的粒子群算法进行仿真分析。

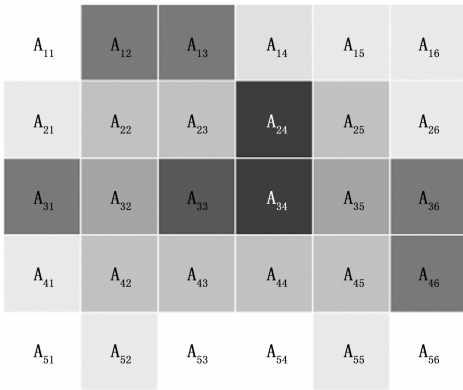


图 2 搜救区域概率分布图

实验的输入数据为概率图分布数据表 2 所示和搜救船舶的属性表 4 所示，输出为任务区域编号  $\{x_a, x_b, x_c, \dots\}$  和相应的累计成功率  $Pos_c$ 。实验中改进粒子群算法的参数设置如表 5 所示，粒子群算法参数设置如表 6 所示，遗传算法参数设置如表 7 所示，设最大迭代次数为  $M$ 。

表 5 改进粒子群算法参数设置

$M$	最大加速度系数 $c_{max}$	最小加速度系数 $c_{min}$	最大惯性权重系数 $\omega_{max}$	最小惯性权重系数 $\omega_{min}$	扰动因子 $c_3$
500	1.5	0.5	0.9	0.4	1.5

表 6 粒子群算法(PSO)参数设置

最大迭代次数	加速度系数	惯性权重系数 $c_1$	惯性权重系数 $c_2$
500	0.9	1.6	1.8

表 7 遗传算法(GA)参数设置

最大迭代次数	初始种群	交叉概率	变异概率
500	200	0.9	0.1

为了更好的进行对比实验，将概率分布图的信息、船舶性能数据代入构建的海上遇险目标搜救模型，在 3 个不同时间段内根据表 4、表 5、表 6 设定的参数，分别运行改进的粒子群算法、PSO 算法和 GA 算法 10 次，记录下每种算法下的最优值。具体搜寻方案对比数据如表 3 所示，各时间段算法迭代曲线分别如图 3~5 所示。

由表 3 可知，在 1:00 到 3:00 时间段，改进 PSO 算法的搜救为 87.68%，比 GA 算法的搜救成功率 86.18% 提高了 0.983%，比 PSO 算法的搜救成功率 86.32% 提高了 0.984%，说明了改进 PSO 算法在海上目标搜寻方面优于 GA 算法和 PSO 算法。时间段 3:00

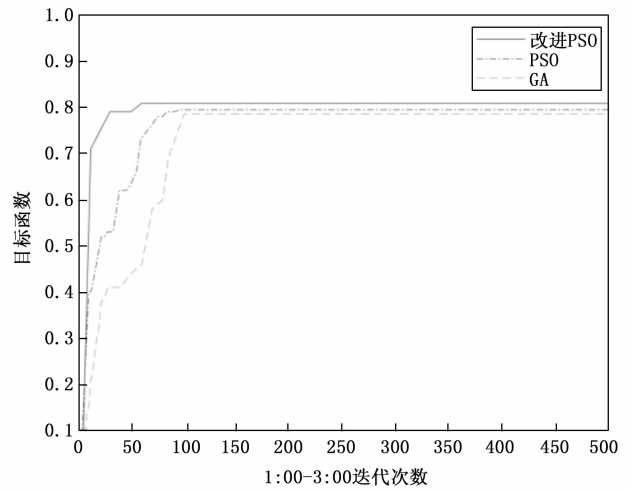


图 3 1:00~3:00 算法迭代过程

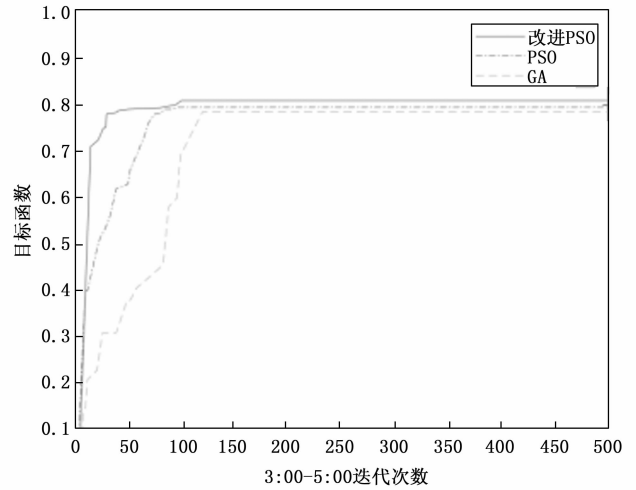


图 4 3:00~5:00 算法迭代过程

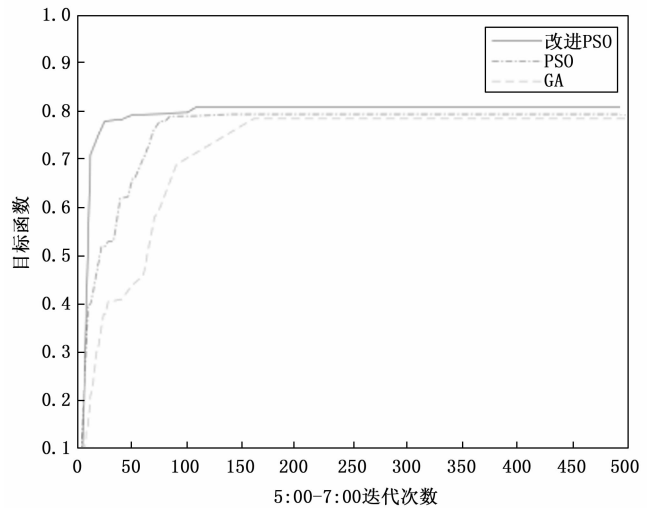


图 5 5:00~7:00 算法迭代过程

~5:00 和 5:00~7:00 内同样说明了改进 PSO 算法稳定地提高了海上遇险目标的搜救成功率。从图 3 看

出,在相同迭代次数下,改进 PSO 算法的目标函数高于 GA 算法和 PSO 算法。并且在整个迭代过程中始终改进 PSO 算法的迭代曲线始终高于 GA 算法和 PSO 算法。改进 PSO 算法比 GA 和 PSO 以更快的速度收敛,并且最终目标函数值改进 PSO 算法曲线高于 GA 算法和 PSO 算法,即累计搜救成功率高。说明改进粒子群算法达到收敛的迭代次数更少,计算速度更快。图 4、5 同样证明了改进的粒子群算法的效率高于 GA 算法和 PSO 算法。这证明了采用的自适应策略具有良好的效果,相比于其他两种算法,不仅提高了收敛速度,还增加了全局搜索能力。

#### 4 结束语

针对海上遇险目标搜寻问题,对传统的粒子群算法进行了改进。在算法中,设置了自适应策略更新惯性权重系数和加速度系数,使算法更具动态调节性。增加了扰动粒子更新机制,扩大了粒子的搜索范围。仿真实验证明,改进后的粒子群算法比传统的粒子群算法以更快的速度达到收敛值,能够快速跳出局部最优解,避免了算法陷入局部最优,在保证搜索效率的同时达到全局优化的效果,提高了海上遇险目标搜寻的成功率,得到可行的搜救优化路径,该算法可以集成应用在海上搜救系统中。为了进一步提高粒子群算法的性能和适应性,后续研究工作中,将搜索问题扩展到多目标和多约束优化问题,或者混合协同其他优化算法,形成混合优化算法,以提高粒子群算法的通用性、自适应性和实际应用效果,获得更合理的决策。

#### 参考文献:

- [1] 李笨帅. 基于最优搜寻理论的海上搜救方案规划方法研究 [D]. 青岛: 山东科技大学, 2020.
- [2] 邢胜伟. 海上立体搜寻全局优化模型及仿真研究 [D]. 大连: 大连海事大学, 2012.
- [3] 朱 崧. 海上搜救目标漂移预测模型研究及应用 [D]. 武汉: 中国地质大学, 2022.
- [4] 张 隽. 海上搜寻包含概率模型设计与仿真 [D]. 大连: 大连海事大学, 2022.
- [5] BERGER J, LO N. An innovative multi-agent search-and-rescue path planning approach [J]. *Computers and Operations Research*, 2015, 5324 - 5331.
- [6] SUN, J., PALADE, et al. Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization [J]. *IEEE Transactions on Industrial Informatics*, 2014, 10 (1): 222 - 232.
- [7] XU G, CUI Q, SHI X, et al. Particle swarm optimization based on dimensional learning strategy [J]. *Swarm and Evolutionary Computation*, 2019, 45: 33 - 51.
- [8] HA M P, NAZARI-HERIS M, MOHAMMADI-IVATLOO B, et al. A hybrid genetic particle swarm optimization for distributed generation allocation in power distribution networks [J]. *Energy*, 2020, 209: 118218.
- [9] 肖方兵. 海上搜救决策支持系统关键技术的研究 [D]. 大连: 大连海事大学, 2012.
- [10] SOZA & COMPANY, LTD. (1996). The theory of search: a simplified explanation: U. S. coast guard [J]. Contract Number: DTCG23-95-D-HMS026. 67 - 84
- [11] 李 岩, 邢胜伟, 张英俊. 多智能体海上搜寻任务分配方法 [J]. *中国航海*, 2018, 41 (3): 91 - 94.
- [12] 江 志. 基于群体智能算法的海上联合搜救 [D]. 大连: 大连海事大学, 2020.
- [13] 于安民. 海上搜寻船舶协同调度方法研究 [D]. 大连: 大连海事大学, 2021.
- [14] KRATZKE T M, STONE L D, FROST J R. Search and rescue optimal planning system [C] // In: 13th Conference on Information Fusion, 2010.
- [15] XIONG W, P H A J M VAN GELDER, YANG K. A decision support method for design and operationalization of search and rescue in maritime emergency [J]. *Ocean Engineering*, 2020, 207: 107399.
- [16] KRATZKE T M, STONE L D, Frost J R. Search and rescue optimal planning system [C] // 2010 13th International Conference on Information Fusion. IEEE, 2010: 1 - 8.
- [17] CHEN Z, LIU H, TIAN Y, et al. A particle swarm optimization algorithm based on time-space weight for helicopter maritime search and rescue decision-making [J]. *IEEE Access*, 2020, 8: 81526 - 81541.
- [18] Michael M, IRENE A, CLAUDE-GUY Q. Ant colony optimization for path planning in search and rescue operations [J]. *European Journal of Operational Research*, 2023, 305 (1): 53 - 63.
- [19] JEAN B, MOHAMED B, NASSIRO L. Near-optimal search-and-rescue path planning for a moving target [J]. *Journal of the Operational Research Society*, 2020, 1 - 13.
- [20] JEAN B, NASSIROU L, MARTIN N. Exact solution for search-and-rescue path planning [J]. *International Journal of Computer and Communication Engineering*, 2013, 266 - 271.
- [21] LV J, LIU M, ZHAO H, et al. Maritime static target search based on particle swarm algorithm [C] // Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016: Volume 2. Springer International Publishing, 2018: 917 - 927.