分类号： TH122　　学校代码：10363
密　级： 公开　　学　　号：2210120161

安徽工程大学
Anhui Polytechnic University

硕士学位论文

题目 海洋捕食者算法的改进及其应用

论 文 作 者　王梦娴
校 内 导 师　赵转哲（教授）
校 外 导 师　涂志健（高级工程师）
专业学位类别　机　械
研究方向（领域）　机器学习、智能计算

论文提交日期：2024 年 5 月 30 日

分类号：TH122　　单位代码：10363

密 级：公开　　学　号：2210120161

捕食者算法的改进及其应用

IMPROVEMENT AND APPLICATION OF MARINE PREDATOR ALGORITHM

学生姓名：王梦娴

指导教师：赵转哲（教授）

校外导师：涂志健（高级工程师）

专　　业：机 械

研究方向：机器学习、智能计算

论文答辩日期：2024 年 5 月 30 日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名： 

日期：　　2024 年　5 月 30 日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

保　密 □，在 ____ 年解密后适用本版权书。

不保密 ☑。

学位论文作者签名：王梦娴　　指导教师签名：赵艳春

日期：2024 年 5 月 30 日　　日期：2024 年 5 月 30 日

海洋捕食者算法的改进及应用

摘要

聚类分析和特征选择是基于机器学习的数据处理的关键技术。元启发式算法以其出色的搜索能力在这两方面备受瞩目。海洋捕食者算法（Marine Predation Algorithm，MPA）是2020年提出的一种元启发式算法，搜索能力极强，在连续搜索空间上具有良好的性能。然而，MPA存在收敛慢和易陷入局部最优的局限，且对自动聚类和离散问题的研究尚少。本文致力于改进MPA，并探索其在自动聚类和特征选择中的应用，以优化数据处理效果。

主要工作如下：

(1) 首先介绍MPA的数学模型和内在迭代机制，通过Markov链模型证明了全局收敛性，经基准函数测试验证其有效性。单因素方差实验揭示了初始参数对MPA性能的影响，减少了参数设置的盲目性。

(2) 提出一种连续型改进的海洋捕食者算法与自适应聚类结合，用于轴承故障状态分类。通过折射反向学习策略、正余弦算法和高斯-柯西变异算子增强种群多样性和算法性能，显著提高了收敛性。结合改进的MPA，提出一种新型自动聚类方法，以聚类有效性指标CS为标准，能迅速确定聚类中心并估计最佳聚类数量。在UCI数据集和西交大轴承数据集上的实验表明，该算法较其他元启发式特征选择算法具有更小适应度函数值和更准确聚类结果，尤其在轴承数据集上分类准确率高达100%，优于MPA的

83.33%，验证了其在实际应用中的有效性。

(3) 提出一种改进的二进制海洋捕食者算法并应用于特征选择。该算法通过引入全局最优个体和莱维飞行策略，提升搜索速度和跳出局部最优能力。在 13 个 UCI 数据集上，改进的算法在特征选择数量和准确率上均优于同类算法，特别是在高维数据上表现突出。在江南大学轴承数据集上，改进算法减少了 5 个冗余特征，验证了其效果和实用性。

关键词：海洋捕食者算法; 自动聚类; 特征选择; 莱维飞行; 机械故障诊断

IMPROVEMENT AND APPLICATION OF MARINE PREDATOR ALGORITHM

ABSTRACT

Cluster analysis and feature selection are key techniques of data processing based on machine learning. Meta-heuristic algorithm has attracted much attention in these two aspects because of its excellent search ability. Marine Predation Algorithm (MPA) is a meta-heuristic algorithm proposed in 2020, which has strong search ability and good performance in continuous search space. However, MPA has the limitations of slow convergence and easy to fall into local optimal, and there are few studies on automatic clustering and discrete problems. This paper aims to improve MPA and explore its application in automatic clustering and feature selection to optimize data processing.

The main work is as follows:

(1) Firstly, the mathematical model and internal iteration mechanism of MPA are introduced. The global convergence is proved by Markov chain model, and its validity is verified by benchmark function test. The single factor variance experiment reveals the influence of initial parameters on the performance of MPA and reduces the blindness of parameter setting.

(2) A continuous improved Marine predator algorithm combined with adaptive clustering is proposed for bearing fault state classification. The population diversity and algorithm performance are enhanced by refraction opposition-based learning, sine-cosine algorithm and Gauss-Cauchy mutation operator, and the convergence is significantly improved. Based on the improved MPA, a new automatic clustering method is proposed. Based on the clustering validity index CS, the clustering center can be quickly determined and the optimal number of clusters can be estimated. The experiments on UCI data set and Xijongda bearing data set show that the proposed algorithm has smaller fitness function values and more accurate clustering results than other meta-heuristic feature selection algorithms. In particular, the classification accuracy rate on bearing data set is as high as 100%, which is better than 83.33% of MPA, verifying its effectiveness in practical applications.

(3) An improved binary Marine predator algorithm is proposed and applied to feature selection. By introducing the global optimal individual and Levy flight strategy, the algorithm improves the search speed and the ability to jump out of the local optimal. On 13 UCI datasets, the improved algorithm outperforms similar algorithms in terms of feature selection quantity and accuracy, especially on high-dimensional data. On the bearing data set of Jiangnan University, the improved algorithm reduces 5 redundant features and verifies its effectiveness and practicability.

KEY WORDS: marine predation algorithm; automatic clustering; feature selection; Lévy flight; machinery fault diagnosis

目录

第一章 绪论

1.1 研究背景和目的

数据挖掘是一种从大规模数据集中自动提取和识别有用信息和关系的技术过程。传统的数据挖掘方法主要针对小规模、低维度以及线性数据集，通常采用手动统计和自主学习的方式进行处理[1]。然而，随着 21 世纪计算机和数字技术的飞速发展，数据采集能力的显著增强使得获取的数据量和复杂度大幅度上升。现代数据集的特征维度可达到千位以上，样本量也大幅增加，这使得手工处理这些数据变得不可行。在这种背景下，机器学习技术应运而生并迅速发展，成为处理大规模复杂数据集的关键技术。机器学习的应用范围广泛，包括图像处理、医学药理、机器视觉等多个领域[2-4]，有效地支持了自动化处理和分析大数据集的需求。当前，如何从庞大且复杂的数据集中提取有价值的信息，挖掘潜在的有用规律，并利用这些规律更好地服务于人类社会和经济活动，已成为机器学习领域面临的一大挑战。

聚类分析是机器学习领域中的一项基础而关键的技术，它将数据对象按照相似度划分为几个群体，使得同一群体内的对象之间相似度高，而不同群体间的对象相似度低[5]。在过去的几十年中，聚类分析已经在多个领域得到了成功的应用。例如，在机械故障模式识别中，通过聚类分析将故障数据划分为不同的簇，以便对其进行分类和标记；通过对历史故障数据的聚类分析，可以揭示出故障发生的规律和趋势，有助于预测机械寿命[6]。然而，随着数据的维度增加和内部结构变得更加复杂，传统的聚类方法在实际应用中遭遇了许多挑战。许多聚类算法需要事先确定聚类的数量，但这个数量在实际应用中往往难以预先知道。对此迫切需要开发出更加合理和高效的聚类方法来应对因数据量增长带来的自动确定最优聚类数量[7]，以便更好地挖掘和分析数据。

特征选择(Feature Selection, FS)是机器学习和数据分析中的一个核心过程，它旨在从初始数据集中挑选出对模型预测贡献最大的特征子集，以此提升模型效率并降低计算成本[8]。此过程对于简化模型结构、加速学习过程、降低过拟合风险及增强模型解释力等方面均具有重要价值。尽管特征选择技术在低至中等维度的数据处理上已经相对成熟，但在处理高维数据（如数千至数万维）时仍面临诸多挑战。一方面会造成特征冗余：高维数据常伴随着大量冗余特征，这些特征不仅无助于提升模型性能，反而可能导致性能下降。另一方面随着特征维度的增加，计算成本和时间消耗也会增加。因

此，如何用最少的评估次数寻找到高性能的特征子集成为关键挑战[9]。

针对上述背景，本文引入并优化了一种新型元启发算法 (Meta-heuristics, MH)[10]——海洋捕食者算法(Marine Predation Algorithm, MPA)[11]，并将其与自动聚类和特征选择技术相融合，提出一种新型智能数据处理技术，特别是在机械故障诊断领域中的应用。

1.2 海洋捕食者算法的研究现状

1.2.1 元启发算法的研究现状

元启发式算法是解决全局优化问题常用的一种方法，它通过模拟自然和人类智慧的方式来寻找最优解，特别适用于解决大规模和复杂问题。一般可根据其模拟机制的不同将其分为三大类[10]：进化算法(Evolutionary Algorithm，EA)、物理化学现象启发算法和群体智能(Swarm Intelligence, SI)算法。

（1）EA

EA 受到自然界中生物进化过程的启发，涵盖了遗传变异和自然选择的原则[12]。进化算法常见的种类包括遗传算法(Genetic Algorithm, GA)[13]、遗传编程(Genetic Programming, GP)[14]和差分进化 (Differential Evolution, DE) 算法[15]等。GA 是其中最知名和应用最广泛的一种，它是模拟达尔文生物进化理论的一种优化模型。在遗传算法中，每个种群成员都代表了解空间中的一个潜在解，通过模拟自然进化的机制来寻找最佳解，因其搜索速度快和高鲁棒性备受学者青睐。GP 则是 GA 的延伸，专注于进化程序或计算机代码的优化。而 DE 算法的核心思想与 GA 相似，不同之处在于 DE 中子代通过父代进行差分变异后再与父代进行交叉产生，最终通过贪婪机制进行选择，大大降低了算法在寻优过程中的计算复杂度，特别适用于处理连续优化问题。

（2）物理启发示算法

物理启发式算法通常源自某种物理和化学现象或启发，在寻找最佳解中首先随机生成一组个体，然后根据一些物理或化学规则，如引力、密度峰值、重磁电等，在整个搜索空间中相互影响和相对移动，以达到最优解[16]。原子轨道搜索(Atomic Orbital Search, AOS)[17]是一种受到量子力学原子轨道理论启发的优化算法，在量子力学中，电子环绕原子核的运动可用一系列原子轨道来描述，这些轨道表示电子出现在原子周围出现的概率分布。多元宇宙优化器(Multiverse Optimizer, MVO)[18]的设计受多元宇宙理论启发，提出存在多个宇宙（或多重宇宙），每个宇宙都有其独特的物理定律和常数，

主要概念涉及白洞、黑洞和虫洞，利用这些天体物理现象的特性引导搜索过程。

（3）SI 算法

1989 年 Beni 等第一次提出了“群体智能”这个概念。而 SI 算法是指一类受自然界中动物群体行为启发的元启发式算法，它汲取了诸如鸟群、鱼群、蚁群等动物群体中观察到的集体行为，如觅食、迁徙、筑巢等行为，以在解空间中寻找全局最优解或可接受的近似解[19]。该类算法的关键特征在于个体之间的简单行为和局部交互能够催生整体群体呈现出复杂的全局行为和智能。由于其简单的结构和较高的效率，SI 算法在优化领域受到了广泛关注，因而研究人员们提出了许多智能优化算法及其改进算法来解决优化问题。粒子群优化算法(Particle Swarm Optimization, PSO)[20]是由美国学者 Eberhart 和 Kennedy 提出的，其核心理念是用一个粒子代表鸟群中的一个个体，对其在寻找食物时的内部信息交流行为进行总结并建立模型，是一种经典的优化算法；李世民等学者受到黏菌在觅食过程中展现的行为和形态变化启发，提出了黏菌算法(Slime Mould Algorithm, SMA)[21]，这种单细胞生物能够通过生长和变形来搜索并连接食物源，其在解决复杂迷宫问题时展现出了惊人的路径优化能力，因此 SMA 被广泛地应用在路径优化、网络设计和运输等方面；Mirjalili 等学者以模拟自然界中座头鲸群体狩猎行为为基础，通过模拟鲸鱼群体搜索、包围、追捕和攻击猎物等过程，实现了优化搜索的目标，从而提出了鲸鱼优化算法(Whale Optimization Algorithm, WOA)[22]除此之外，近几年诸如胡桃夹子鸟优化算法(Nutcracker Optimization Algorithm, NOA)[23]、哈里斯鹰算法(Harris Hawk Optimization, HHO)[24]等新型算法被国内外学者相继提出。

1.2.2 海洋捕食者算法的研究现状

MPA 是一种受海洋生态系统中捕食行为启发的元启发式算法，它模仿如鲨鱼、巨蜥、太阳鱼、马鱼和箭鱼等海洋捕食者的追捕策略来解决优化问题。MPA 通过简化自然界捕食者和猎物之间的复杂交互，转化为算法中的搜索和优化过程，从而高效地寻找问题的最优解。张磊等人的研究通过将 MPA 中的捕食者根据适应度值分为三个不同的子群，对每个子群采取不同的操作策略，有效扩大了搜索的随机性和选择性，这种分层策略提高了算法的求解速度，但也指出在低维空间中算法容易陷入局部最优解的问题[25]；付华等人提出的改进方法在算法的不同阶段引入差分演化、正余弦策略以及柯西变异与反向学习策略，这一多阶段融合策略不仅提高了种群的质量，还有效避免了算法的早熟现象，尽管如此，其收敛精度仍需进一步提升[26]；Xing 等人将 MPA 与

反向学习算法及自适应权重结合，开发出用于红外图像故障诊断的多级阈值图像分割方法，展示了 MPA 在多目标优化场景下的应用效果[27]；Jia 等人的研究通过引入自适应权重机制使 MPA 能有效跳出局部最优解，应用此策略于太阳能电池板系统的多目标优化，显著提高了建筑物的能源效率[28]。Mohamed 等人将量子理论与 MPA 相结合，开发出一种新的算法用于医学图像的预处理和分割，这种方法可以有效处理图像中的离散数据，提高分割的准确性，特别是在处理复杂的医学图像结构时，显示了优越的性能[29]；Sowmya 等研究者针对电动汽车在长时间停车期间的充电和放电调度问题，提出了一种融合反向学习算法的增强型 MPA，此算法优化了电动汽车的调度方案，最大限度地降低了电力消耗和等待时间，提高了电动汽车使用的经济性和便捷性[30]；Ajayi 等人将 MPA 与金枪鱼群算法结合，开发出一种新的优化方案来处理电力系统的经济调度问题，旨在最小化燃料成本[31]。

尽管 MPA 的应用范围不断扩大，其理论基础目前研究较为稀缺，尤其在收敛性分析和初始参数设置的优化方面，这些理论缺陷限制了 MPA 的潜在优势。且 MPA 在离散优化问题领域的运用还相对较少，这表明 MPA 的应用潜力仍有很大的扩展空间。

1.3 元启发算法在自动聚类上的研究现状

聚类分析广泛应用于各种领域，如市场研究、社会网络分析、图像处理、生物信息学等，以发现数据中的自然分组、模式识别或数据压缩。K-均值聚类(K-means Clustering, KCM)[32]最常用的聚类方法之一，通过迭代优化聚类中心，将数据点分配到最近的聚类中心，以最小化每个聚类中点到聚类中心的距离之和，KCM 适用于大规模数据集，但需要预先指定聚类数目 K，并对初始聚类中心敏感。层次聚类[33]是通过构建聚类的层次树状结构（聚类树或树状图）来组织数据，可以是自底向上的聚合方法或自顶向下的分裂方法。层次聚类无需预先指定聚类数目，但计算复杂度较高，不适合大规模数据集。高斯混合模型利用高斯分布（正态分布）的概率模型来表示聚类，适合于估计聚类的形状和大小[34]。

许多聚类算法需要预先指定聚类数目，而在实际应用中这一数目往往不是显而易见的，因此针对自动聚类分析是一个挑战热点。自动聚类分析通常依赖于算法和评估指标来确定最佳聚类数。其中基于群体智能优化的自动聚类分析是一种常见有效的方法。Abolfazl 等提出了将模拟退火算法和模糊 C-means 聚类方法结合，以解决使用 GT2 模糊集进行微阵列数据聚类的问题，并实验证明了该框架在高度不确定的环境下的有

效性和灵活性[35]；Alokananda 等将增强粒子群优化算法与量子计算框架相结合产生了量子启发增强粒子群优化算法，并将其应用于彩色图像的自动聚类中[36]；李飞等人采用了一种无需额外参数的自动快速密度峰值聚类方法来生成子种群，通过选择那些自身密度大或与其他粒子的相对距离较远的点作为聚类中心，从而自动产生子种群，这样做的好处是避免了引入外部参数的需求[37]；Zhang 等以全局-局部优化为思想，构建了基于燃气温度分布的混合遗传算法-准牛顿法优化参数模型，该模型可以有效地识别出故障燃烧器[38]。

尽管国内外有许多关于元启发算法和聚类分析结合的研究，但对自动聚类的研究相对较少。

1.4 元启发算法在特征选择上的研究现状

FS 的主要目的是在保持性能精度的同时减小特征集的维度。FS 方法大致可以分为三类：过滤法、包装法和嵌入法 。过滤方法[39]一般使用评价标准来增强特征与类之间的相关性，降低特征之间的相关性。该方法优点是计算速度快，不依赖于任何预测模型。尽管该方法能够自动产生优化子集，但它无法确保选出的特征子集是最优化的，特别是在特征与分类器高度相关的情况下。因此，即便能找到符合要求的优化子集，其规模往往较大，并且可能会包含一些容易识别的噪声特征。包裹方法[40]通常在数据集的基础上生成多个特征子集，由分类器对这些特征子集进行评估和比较，直接利用这些能够达到较高分类性能的特征得到分类性能更高的分类模型。虽然这种方法在执行速度上不如过滤法快，但它能够选择出更小规模的特征子集，这对于识别关键特征非常有帮助。此外，尽管它的精确度较高，但泛化能力不足，并且具有较高的时间复杂度。嵌入式[41]是过滤法和包装法的组合算法，将 FS 过程与学习过程联系起来。其基本思想是：首先采用滤波方法对特征进行预选，去除冗余特征并降低数据集的维数，然后采用包装法对特征子集进行选择。许多传统 FS 算法存在搜索效率低下的问题，表现为耗时长或搜索能力差。随着时间的推进，基于 SI 算法技术的特征选择方法开始受到广泛关注和研究。这些方法模仿自然规律，采用种群机制进行工作，使得种群中的信息能够相互交流，进而拥有出色的全局搜索能力。

Somnath 等使用深度神经网络 VGG16 与蜻蜓算法相结合来检测乳腺癌，同时使用迁移学习来避免 CNN 模型和小型数据集的过度拟合，并且在公开的 DMR-IR 乳腺癌数据集上取得了令人印象深刻的结果[42]。Zhang 等提出了一种用于风速预测的二元回溯

算法，其中采用极限学习机进行特征选择[43]。Arpan 等提出了一个端到端的用于从 CT 扫描图像中检测 COVID-19 框架，其中的 FS 部分采用和谐搜索算法结合自适应爬山算法来获得更好的性能，拟议的框架在 SARS-COV-2 CT 扫描数据集上进行了评估，结果优于许多比较算法[44]。为了对 MR 脑肿瘤图像进行分类，Kaur 等通过无参数的蝙蝠算法与 SVM 分类器一起应用于此[45]。Zhao 等提出了一种基于改进播种策略和混沌种群的二元蒲公英算法，并将其应用在极限学习机的特征选择问题[46]。

尽管元启发式算法在解决特征选择问题方面取得了巨大成功，但仍存在一些挑战和问题。比如特征选择的稳定性和分类精度选择的不平衡。为了找到最佳分类，当特征之间存在高度相关性时，特征选择就会出现不稳定性，并且由于获得最佳分类精度而将其重要特征删除。

1.5 主要研究内容安排

本文总共分为 5 章，各章节研究内容如下，具体研究路线如图 1-1，

第 1 章 介绍了课题的研究背景与目的、元启发算法和 MPA 算法以及元启发算法在自动聚类和特征选择方面的研究现状，最后对本文的主要内容进行了详细介绍。

第 2 章 概述 MPA 算法，分析其理论基础，建立 Markov 链模型以探究全局收敛性，并通过基准函数测试其性能。同时，探讨了 MPA 参数对算法性能的影响。

第 3 章 针对标准 MPA 的局限，提出融合多种策略的改进算法，结合 K-means 实现自动聚类，并通过机械故障诊断实例验证其有效性。

第 4 章 针对离散 MPA 的局部最优问题，提出基于全局最优个体和莱维飞行的增强算法，并引入新特征选择方法，在 UCI 数据集上证实其在高维数据预测聚类中的卓越性能。

第 5 章 总结了全文内容，并指出了论文研究存在的不足之处，并对未来的工作进行了展望。

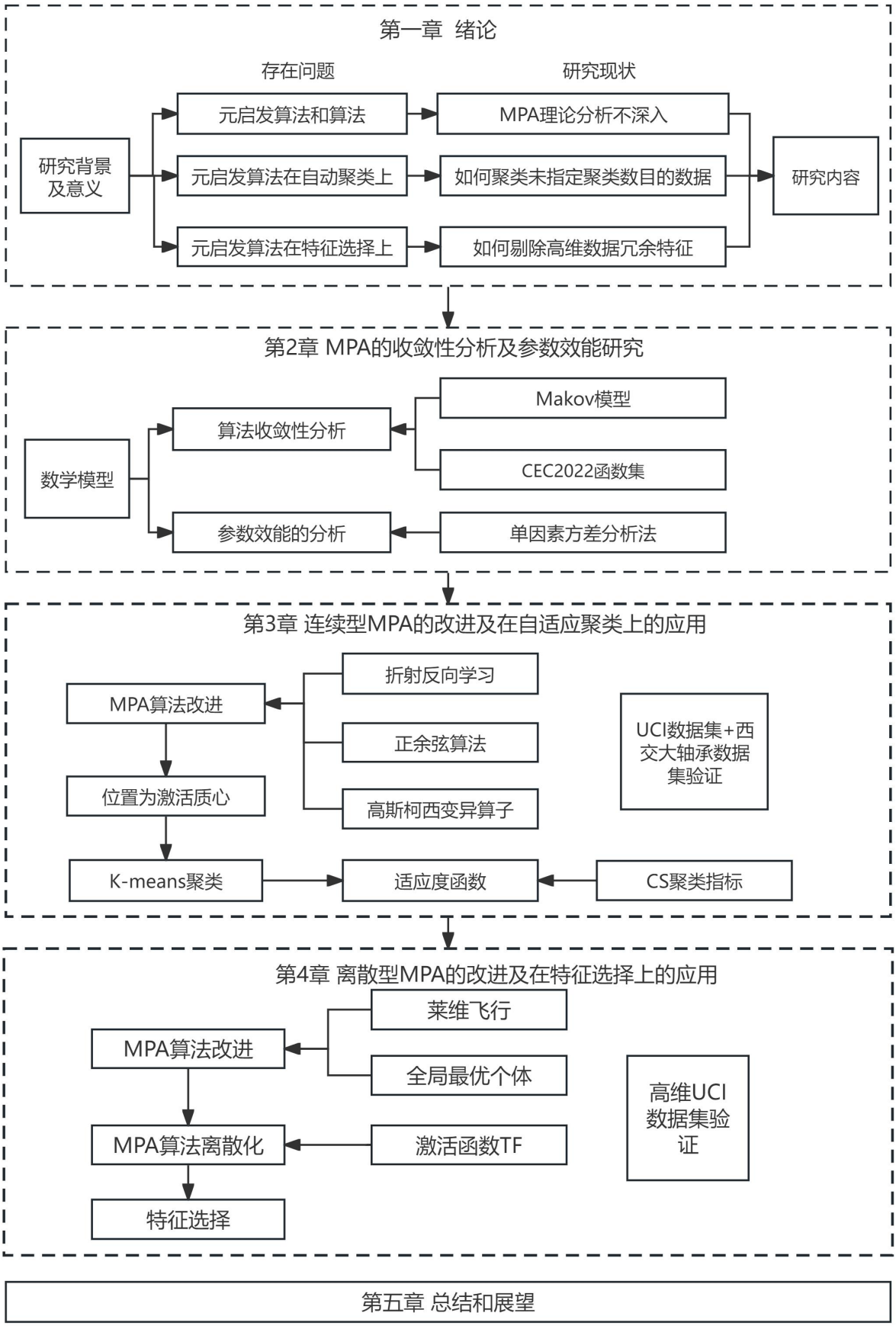

第一章 绪论
存在问题
研究现状
研究背景及意义
元启发算法和算法
MPA理论分析不深入
元启发算法在自动聚类上
如何聚类未指定聚类数目的数据
元启发算法在特征选择上
如何剔除高维数据冗余特征
研究内容
第2章 MPA的收敛性分析及参数效能研究
数学模型
算法收敛性分析
Makov模型
CEC2022函数集
参数效能的分析
单因素方差分析法
第3章 连续型MPA的改进及在自适应聚类上的应用
MPA算法改进
折射反向学习
正余弦算法
高斯柯西变异算子
位置为激活质心
K-means聚类
适应度函数
CS聚类指标
UCI数据集+西交大轴承数据集验证
第4章 离散型MPA的改进及在特征选择上的应用
MPA算法改进
莱维飞行
全局最优个体
MPA算法离散化
激活函数TF
特征选择
高维UCI数据集验证
第五章 总结和展望

图 1-1 研究路线

第 2 章 MPA 的收敛性分析及参数效能研究

本章中，首先系统地介绍 MPA 的基础理论和具体实现过程。鉴于目前对 MPA 理论基础的研究尚未深入，进一步探讨 MPA 的收敛性并进行理论推演。此外，为探究 MPA 的参数效能，设计相关实验并根据实验结果总结参数对算法性能的影响规律。通过这一系列工作，希望能够深化对 MPA 的理解，并为其在实际应用中的优化提供理论支持。

2.1 MPA 的基本原理

2.1.1 仿生原理

MPA 是一种模拟海洋生物觅食规律和生物间相互作用的最优遭遇策略的元启发式算法，觅食策略可概括为交替进行的莱维飞行和布朗运动。下面介绍这两种运动方式，图 2-1 为布朗运动和莱维飞行在[–2,2.5]间的随机步长图。

（1）布朗运动[47]：是一种服从正态分布的连续随机过程，概率密度函数如下，其中均值 $\mu_B = 0$，单位方差 $\sigma_B{}^2 = 1$：

$$f_B(x) = \frac{1}{\sqrt{2\pi\sigma_B{}^2}}\exp\left(-\frac{(x-\mu_B)^2}{2\sigma_B{}^2}\right) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right) \tag{2-1}$$

（2）莱维飞行[48]：是一种随机移动机制，许多昆虫和动物都遵循这种类似自然界莱维飞行的方式。其步长的概率分布呈幂律渐进的特性，在这种移动方式中，粒子大部分时间内会进行短距离微小移动，但偶尔也会出现长距离大跳跃。定义如下：

$$Levy(\eta) = 0.05\cdot\frac{\delta}{|\vartheta|^{1/\eta}} \tag{2-2}$$

式中，幂律指数 $\eta = 1.5$；δ 决定莱维分布的形状，$\delta \sim N\left(0,\sigma_\delta^2\right)$；$\vartheta$ 为莱维飞行的步长，$\vartheta \sim N(0,1)$。其中 σ_δ 由下式给出：

$$\sigma_\delta = \left[\frac{\Gamma(1+\eta)\sin\left(\frac{\pi\eta}{2}\right)}{\Gamma\left(\frac{(1+\eta)}{2}\cdot\eta\cdot 2^{\frac{(\eta-1)}{2}}\right)}\right]^{\frac{1}{\eta}} \tag{2-3}$$

式中，Γ 是标准的 Gamma 函数。

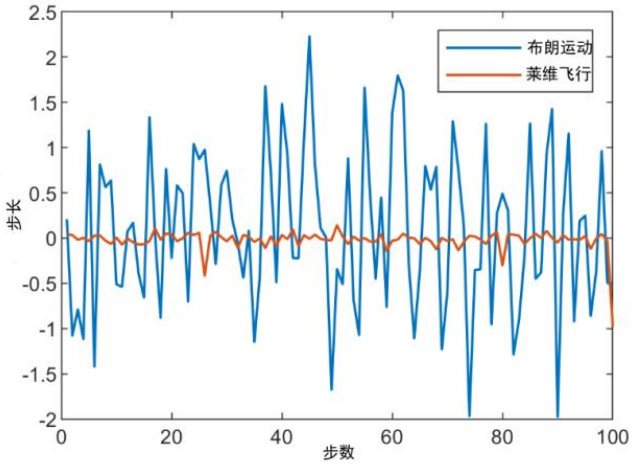


图 2-1 布朗运动和莱维飞行随机步长图

2.1.2 基本思想与数学模型

MPA 的优化过程可按照捕食者和猎物的速度比划分为三个不同阶段，猎物同时充当捕食者的角色，更贴近自然界规律，又考虑了海洋生态环境的影响，从而有助于减少捕食者陷入局部最优解的风险。接下来将详细介绍 MPA 算法的数学模型。

（1）初始化

设种群数量为 N，求解问题的维度为 D，求解空间的上下界分别为 $UB=[ub_1,\cdots,ub_D]$ 和 $LB=[lb_1,\cdots,lb_D]$，$R_i\left(i=1,2,3,4\right)$ 为服从 $[0,1]$ 均匀分布的随机数。所有捕食者构成 $N\times D$ 维的矩阵 X，第 i 个捕食者在第 j 维的位置初始化为：

$$x_{i,j}=lb_j+R_1(ub_j-lb_j) \tag{2-4}$$

设最优种群位置为 gb，重复 N 次构成精英矩阵 E。Max_Iter 为最大迭代次数，$Iter$ 为当前迭代次数。

（2）位置更新阶段

方式 1：在猎物的移动速度比捕食者的移动速度快时，$Iter<Max\_Iter/3$，此时捕食者执行勘探任务，数学模型可用如下方程表示：

$$\begin{cases} stepsize_i=R_B\otimes\left(E_i-R_B\otimes x_i\right) \\ x_i=x_i+0.5\cdot R_2\otimes stepsize_i \end{cases} i=1,2,\cdots,N \tag{2-5}$$

其中，R_B 表示采用布朗运动的随机向量。

方式 2：在猎物与捕食者的速度相等时，$Max\_Iter/3 \le Iter \le 2*Max\_Iter/3$ 。此时捕食者种群被一分为二，一半继续勘探，采用布朗运动策略；而另一半转向开发任务，实施莱维飞行策略。该过程的数学模型描述如下：

$$\begin{cases} stepsize_i = R_L \otimes \left(E_i - R_L \otimes x_i\right) \\ x_i = x_i + 0.5 \cdot R_2 \otimes stepsize_i \end{cases} i = 1, 2, \cdots, \frac{N}{2} \tag{2-6}$$

$$\begin{cases} stepsize_i = R_B \otimes \left(R_B \otimes E_i - x_i\right) \\ x_i = E_i + 0.5 \cdot C_F \otimes stepsize_i \end{cases} i = \frac{N}{2} + 1, \cdots, N \tag{2-7}$$

$$C_F = \left(1 - \frac{Iter}{Max\_Iter}\right)^{\frac{2 \cdot Iter}{Max\_Iter}} \tag{2-8}$$

式中，R_L 是服从莱维分布的随机向量，C_F 是调整移动步长的自适应参数。

方式 3：捕食者的移动速度比猎物快，$Iter > 2*Max\_Iter/3$ 。为了更有效地实施觅食，捕食者此时采用莱维飞行作为最佳策略。数学模型如下：

$$\begin{cases} stepsize_i = R_L \otimes \left(R_L \otimes E_i - x_i\right) \\ x_i = E_i + 0.5 \cdot C_F \otimes stepsize_i \end{cases} i = 1, 2, \cdots, N \tag{2-9}$$

（3）涡流形成和鱼群聚集装置(Fish Aggregating Devices, FADs)的影响

在海洋生态系统中，涡流形成以及 FADs 的环境因素的变动会对生物的捕食策略产生深远影响，这些复杂因素可通过以下公式来描述。

$$x_i = \begin{cases} x_i + C_F\left[LB + R_L \otimes \left(UB - LB\right)\right] \otimes U, & R_3 \le P_{FADs} \\ x_i + \left[P_{FADs}\left(1 - R_3\right) + R_3\right]\left(x_{i1} - x_{i2}\right) & R_3 > P_{FADs} \end{cases} \tag{2-10}$$

式中，P_{FADs} 代表受海洋环境变动影响的概率值，通常设为 0.2。U 代表一个包括 0 和 1 二元矩阵向量，它由[0,1]生成的随机数数组决定，如果随机数低于 0.2，则对应位置被置为 0；反之置为 1。下标 $i1$ 和 $i2$ 表示捕食者矩阵中的随机索引位置。

MPA 算法具备记忆存储功能，这一特性极大地增强了其搜索效率和解的质量。这种记忆功能使得 MPA 不仅可以存储历史上的最佳位置，而且能够在算法的每次迭代中进行有效的贪婪选择，即通过持续比较个体之间的适应度值，并保留最优解，从而导向全局最优。

2.1.3 算法伪代码和流程图

结合 2.1.2 节对算法概念及其数学模型的描述，归纳出 MPA 的伪代码如表 2-1 和流

程图如图 2-2 所示：

表 2-1 MPA 伪代码

算法参数(种群数量、最大迭代次等)设定以及根据式(2-4)进行种群初始化
While 在算法的终止条件未得到满足的情况下
计算适应度值，并构建精英矩阵以及执行海洋记忆存储
If $Iter < Max\_Iter/3$
利用公式(2-5)实现更新捕食者位置
Else If $Max\_Iter/3 \le Iter \le 2 \cdot Max\_Iter/3$
For 种群的前一半$(i = 1, 2, \ldots, N/2)$
利用公式(2-6)对捕食者位置进行更新
For 种群的后一半$(i = N/2+1, \cdots, N)$
利用公式(2-7)对捕食者位置进行更新
Else If $Iter > 2 \cdot Max\_Iter/3$
利用公式(2-9)实现位置更新
End If
执行海洋记忆存储并对精英矩阵进行刷新
利用公式(2-10)模拟 FADs 效应，进一步对捕食者位置进行调整
End While
返回最优个体的位置及其适应度值

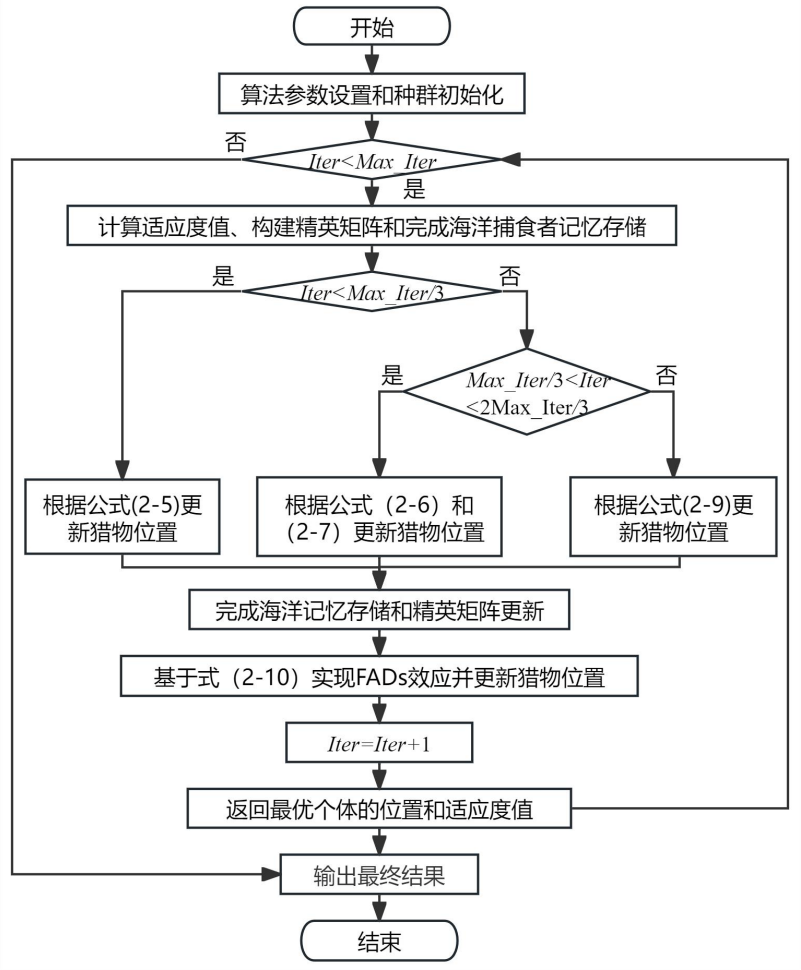


图 2-2 海洋捕食者算法的流程图

2.2 MPA 的收敛性分析

尽管 MPA 算法在各个领域展现出了有效性和优越性，引起研究者们的广泛关注，但其理论基础仍相对薄弱。目前尚未有关 MPA 算法收敛性分析的研究成果。通过对 MPA 的收敛性能展开分析，可揭示该算法的运行原理，为算法改进提供指导。本节将运用马尔可夫(Markov)链对 MPA 算法的收敛性进行总结和深入分析，并通过对测试函数的仿真实验来进一步验证 MPA 算法的收敛性。

2.2.1 基本定义

先介绍一些基本定义以便更好理解后续的理论分析。

定义 1（随机过程）随机过程是随机变量的集合，表示为$\Omega=\{H_t|t\in T\}$，即对于参数集T，H_t是一个随机变量。通常t表示时间，H_t是时间t时的过程状态。

定义 2（离散时间随机过程和连续时间随机过程）$\Omega=\{H_t|t\in T\}$是一个离散时间的随机过程，若参数集T是可数且连续，则将其定义为连续时间随机过程。

定义 3（Markov 状态和 Markov 链）Markov 状态表示任何时刻的状态空间只取决于它前一刻的状态空间，与其他状态无关，且状态空间为可列集。假设存在一个随机过程$\Omega=\{H_t|t\in T\}$，其参数集T构成一个离散的时间序列，形式为$T=\{0,1,2,\cdots\}$，其中对应可能取值H_t全集构建了一个离散的状态集合$I=\{i_0,i_1,i_2,\cdots\}$。若对任意整数$t\in T$以及任何可能的$\{i_0,i_1,\cdots,i_{t+1}\}\in I$条件概率都成立，即：

$$P\{H_{t+1}=i_{t+1}|H_0=i_0,H_1=i_1,\cdots,H_t=i_t\}=P\{H_{t+1}=i_{t+1}|H_t=i_t\} \tag{2-11}$$

若满足上述条件，则随机过程$\{B_t,t\in T\}$被定义为 Markov 链。

定义 4（有限 Markov 链）若一个 Markov 链的状态空间有限，则称之为有限 Markov 链。

定义 5（齐次 Markov 链）若从一个状态i到另一个状态j的转移概率不随时间变化，即状态转移过程与时间t无关，则称为齐次 Markov 链。

2.2.2 MPA 算法的 Markov 链模型

MPA 算法本质上是一种基于群体的随机优化算法，种群的运动可用基于随机过程的 Markov 链来描述。简化捕食者位置更新公式如下：

$$x_{t+1} = c^1 \cdot x_t + c^2 \cdot E_t \tag{2-12}$$

其中，c^1, c^2 取固定值。

定义 6（捕食者状态及其状态空间）捕食者的状态是其在种群的位置 x。一个捕食者所有可能状态的集合构成其状态空间，由集合 $\mathbf{IS} = \{x | x \in \mathbf{B}\}$ 表示，其中 **B** 是可行域。

定义 7（捕食者群状态及其状态空间）若 x_i 表示第 i 个捕食者，则种群的状态用 $X = \{x_1, x_2, \cdots, x_N\}$ 表示，其状态空间由 $\mathbf{GS} = \{X = \{x_1, x_2, \cdots, x_N\} | x_i \in \mathbf{IS}, 1 \le i \le N\}$ 表示。

定义 8（状态等价）定义一个函数 $\varphi(X, x) = \sum^N \chi|\mathbf{IS}|(x_i)$，其中 $X \in, x \in \mathbf{IS}, \chi|\mathbf{IS}|$ 表示事件 **IS** 的示性函数，$\varphi(X, x)$ 是种群状态 X 中包含的个体 x。若存在两个捕食者群 $X_1, X_2 \in \mathbf{GS}$，对于 $\forall x \in \mathbf{IS}$，若有 $\varphi(X_1, x) = \varphi(X_2, x)$，则称 X_1 和 X_2 等价，记作 $X_1 \sim X_2$。

等价关系有以下性质：

性质 1：设 **L** 为等价类，则 **L** 中包含的任意两个种群状态彼此等价。

性质 2：对于任意两个等价类 $\mathbf{L}_1, \mathbf{L}_2$，$\mathbf{L}_1$ 的种群状态与 $\mathbf{L}_2$ 的种群状态不相等。

性质 3：所有等价类都是不相交的，即 $\mathbf{L}_1 \cap \mathbf{L}_2 = \varnothing$。

定义 9（状态转换）在 MPA 算法中，对 $\forall x_i, x_j \in \mathbf{IS},$，一步状态转移 $x_i \to x_j$ 由 $T_{\mathbf{IS}}(x_i) = x_j$ 表示，转移概率为 $P(T_{\mathbf{IS}}(x_i) = x_j)$。把种群视为超空间上点的集合，位置的更新相当于在超空间中进行集合点的交换操作。捕食者位置的状态转移概率推导如下：

$$P(T_{\mathbf{IS}}(x_i) = x_j) = \begin{cases} \dfrac{1}{|-c^2 E_i|}, & x_j \in [c^1 x_i, c^1 x_i + c^2 E_i] \\ 0, & \text{其他} \end{cases} \tag{2-13}$$

定义 10（MPA 的状态转移概率）一步状态转移 $X_i \to X_j$ 表示种群 X_i 内所有捕食者的状态同时转移到种群 X_j 中相应捕食者的状态，其状态转移概率表达式如下：

$$\begin{aligned} &P(T_{\mathbf{GS}}(X_i) = X_j) = P(T_{\mathbf{IS}}(x_{i1}) = x_{j1}) \times P(T_{\mathbf{IS}}(x_{i2}) = x_{j2}) \times \cdots \\ &\times P(T_{\mathbf{IS}}(x_{iN}) = x_{jN}) = \prod_{k=1}^{N} P(T_{\mathbf{IS}}(x_{ik}) = x_{jk}) \end{aligned} \tag{2-14}$$

定理 1 种群状态序列$\{X(t),t>0\}$是一条有限的齐次 Markov 链。

证明：根据 MPA 定义的状态转移概率，可知对于种群状态序列为$\{X(t),t>0\}$，群转移概率为$P(T_{\mathrm{GS}}(X(t))=X(t+1))$，其中$X(t),X(t+1)\in\mathbf{GS}$，该转移概率又取决于个体转移概率$P(T_{\mathrm{IS}}(x(t))=x(t+1))$，由公式(2-13)可知$x(t+1)$的状态仅于$t$时刻$x(t)$和$gb$有关，和时间$t$无关，则种群状态序列$\{X(t),t>0\}$具有 Markov 性，且满足定义 5 具有齐次性。假设 MPA 的所有搜索空间都是有限的，则状态空间**GS** 是有限的，满足定理 4。因此种群状态序列$\{X(t),t>0\}$是一条有限的齐次 Markov 链。

2.2.3 随机算法收敛的标准

借助随机搜索算法的收敛规则，研究随机搜索算法 MPA 的收敛特性。设一个随机优化算法Z应用于优化问题$\langle\mathbf{B},f\rangle$，其中第$k$次迭代的结果为$x_k$，下一次迭代后的结果是$x_{k+1}=Z(x_k,\varsigma)$，$f$是适应度函数，$\varsigma$代表 MPA 已搜索到的解。对于随机搜索算法，收敛意味着找到了一个单调序列$\{f(x_k)\}_{k=0}^{\infty}$最终能趋近$f$在可行域**B** 上的下确界（理想值）。当 inf(f) 是奇点时找不到f的最小值的。故引入 Lebesgue 测度空间来定义搜索的下确界：

$$\Psi=\inf\left(t\middle|\, v\left[x\in\mathbf{B}\middle|\, f(x)<t\right]\right) \tag{2-15}$$

其中，$v[X]$表示在集合X上的 Lebesgue 测度。式(2-15)表明搜索空间内存在一个非空子集，其成员的适应度值能够无限逼近Ψ。最佳区域定义为：

$$R_{\varepsilon,\rho}=\begin{cases}\{x\in\mathbf{B}|f(x)<\Psi+\varepsilon\}, & \Psi\text{有限}\\ \{x\in\mathbf{B}|f(x)<-\rho\}, & \Psi=-\infty\end{cases} \tag{2-16}$$

其中，$\varepsilon>0$，$\rho<0$，若随机算法Z在区域$R_{\varepsilon,M}$内找到一个点，则认为该算法能够接近全局最优或可接受的全局最优解。

定理 2（全局收敛定理）设f可测，**B** 是R^D的可测子集，且f满足下列条件：

(1) $f(Z(x,\xi))\le f(x)$，且若$\xi\in\mathbf{B}$，则有$f(Z(x,\xi))\le f(\xi)$

(2) 对$\forall s\in\mathbf{B},v[X]>0$，有

$$\prod_{k=0}^{\infty}\left[1-v_k\left(X\right)\right]=0 \tag{2-17}$$

且$\left\{x_k\right\}_{k=0}^{\infty}$是算法$Z$生成的序列，则有：

$$\lim_{k\to+\infty} P\left[x_k \in R_{\varepsilon,M}\right]=1 \tag{2-18}$$

其中，$P\left[x_k \in R_{\varepsilon,M}\right]$是第$k$次迭代的解$x_k$位于$R_{\varepsilon,M}$中的概率。

2.2.4 MPA 算法的收敛性

定理 3 MPA 算法满足条件 1。

证明：根据条件 1，对于 MPA 算法，迭代中的当前最优解为

$$E_t=\begin{cases}E_{t-1}, & f\left(x_t\right)>f\left(E_{t-1}\right)\\ x_t, & f\left(x_t\right)\le f\left(E_{t-1}\right)\end{cases} \tag{2-19}$$

这意味着 MPA 满足条件(1)。

定义 11（最优个体状态集**J**和最优种群状态集**G**）对于优化问题$\left\langle \mathbf{B},f\right\rangle$的个体全局最优解$gb$，最优个体状态集定义为$\mathbf{J}=\left\{X=\left(x\right)\middle|f\left(x\right)=f\left(gb\right),X\in\mathbf{GS}\right\}$。若$\mathbf{J}=\mathbf{GS}$，则可行空间中的解集都是最优解，因此优化没有意义，则$\mathbf{J}\subset\mathbf{GS}$。最优种群状态集定义为$\mathbf{G}=\left\{g=\left(X_1,X_2,\cdots,X_N\right)\middle|\exists X_i\in\mathbf{J},1\le i\le N\right\}$。

定义 12（闭集）设**S**是一个一般状态空间，若对$\forall i\in\mathbf{S}$，由$\sum_{j\in\mathbf{S}}P_{i,j}=1$，称状态空间**S**为闭集。

定义 13（不可约闭集）若闭集**S**有真闭子集，则称**S**为可约，反正称**S**不可约。

定理 4 最优个体状态集**J**是 MPA 算法中状态空间**IS**的闭集。

证明：通过证明算法生成的总体序列满足定义 12 来证明在**IS**中闭合的最优状态集**J**。在 MPA 算法中，位置更新策略中使用了最佳的个体保留机制。假设当前最佳个体的值比先前的最佳个体更合适，则替换前一个个体。随着迭代的增加，可知得到的值比前者更好，而不是更糟。因此$P\left(x_{k+1}\notin\mathbf{J}\middle|x_k\in\mathbf{J}\right)=0$，则最优状态集**J**满足定义 12。

定理 5 最优种群状态集**G**是 MPA 算法中状态空间**GS**的闭集。

证明：从定义 11 中可得$\mathbf{G}=\left\{g=\left(X_1,X_2,\cdots,X_N\right)\middle|\exists s_i\in\mathbf{J},1\le i\le N\right\}$，若在第$k$次迭代时$g_k=\left(X_1,X_2,\cdots,X_N\right)\in\mathbf{G},X_i\in\mathbf{J}$，**J**是概率为 1 的闭集，根据定理 4 可知必须存在

$x_{k+1}\in\mathbf{J}$。由定义 11 可知在第 $k+1$ 次迭代时 $g_{k+1}=\left(X_1,X_2,\cdots,X_N\right)\in\mathbf{G}$。因此 $P\left(g_{k+1}\notin\mathbf{G}\middle|g_k\in\mathbf{G}\right)=0$，根据定义 12 可得出结论：**G** 是一个闭集。

定理 6 在 MPA 算法种群状态空间 **GS** 中不存在满足 $\mathbf{\Xi}\cap\mathbf{G}=\varnothing$ 的非空闭集 **Ξ**。

证明：设 $\forall X_i,X_j\in\mathbf{G}$，对任意移动步长 l，$l\geq1$，由 Ckapman-Kolmogorov 方程[49]可得：

$$P_{X_i,X_j}^{l}=\sum_{X_{r1}\in\mathbf{GS}}\cdots\sum_{X_{rl-1}\in\mathbf{GS}}P\left(T_{\mathbf{IS}}(x_j)=x_{r1}\right)P\left(T_{\mathbf{IS}}(x_{r1})=x_{r2}\right)\cdots P\left(T_{\mathbf{IS}}(x_{rl+1})=x_i\right)\tag{2-20}$$

因此，对于无限次迭代，每次一步转移概率 $P\left(T_{\mathbf{IS}}\left(x_{r_{c+j}}\right)\right)=x_{r_{c+j+1}}$ 的展开式(2-20)满足方程(2-14)，即 $P\left(T_{\mathbf{IS}}\left(x_{r_{c+j}}\right)\right)=x_{r_{c+j+1}}>0$。因此 **Ξ** 不是一个非空闭集，与假设矛盾。因此，在状态空间 **GS** 中只有一个闭集 **Ξ**。

引理 1 假设 Markov 链有一个非空集合 **C**，并且不存在一个非空闭集 **D**，那么 $\mathbf{C}\cap\mathbf{D}=\varnothing$，则：

$$\begin{cases}\lim\limits_{n\to\infty}P\left(x_n=j\right)=\pi_j, & j\in\mathbf{C}\\ \lim\limits_{n\to\infty}P\left(x_n=j\right)=0, & j\notin\mathbf{C}\end{cases}\tag{2-23}$$

定理 7 当迭代次数接近无穷大时，种群状态序列将收敛到最优状态集 **G**。

证明：根据定理 5、6 和引理 1，可得出该定理成立的结论。

定理 8 种群状态集 **GS** 是可约的

证明：从定理 5 中，可知最优种群状态集是一个闭集 $\mathbf{G}\subset\mathbf{GS}$。因此，根据定义，得出结论，种群状态集 **GS** 是可约的。

定理 9 MPA 算法全局收敛。

证明：MPA 算法中每次迭代的最优解都被保存，故从定理 3 中，可以看到满足条件 1。根据定理 7，可知在足够多的迭代或无穷大后，种群状态序列将收敛到最优集。因此，未找到全局最优解的概率为 0，满足条件 2。因此，根据全局收敛定理，得出结论，海洋捕食者算法具有全局收敛性。

2.3 性能测试

本研究的所有程序均在计算机配置为 CPU 2.30GHz、内存 16GB， MATLAB R2021b 软件进行实验。为客观评估 MPA 算法的性能，本文选择了 CEC2022 测试函数，每个函数包含偏差、移位和旋转三种操作。设置维数为 10，具体信息如表 2-2 所示。

选择蜣螂优化算法(Dung beetle optimizer, DBO)[50]、PSO[20]、灰狼优化算法(Grey Wolf Optimization, GWO)[51]、麻雀搜索算法(Sparrow Search Algorithm SSA)[52]作为对比对象。为保证实验的公平性，所有算法共有参数统一：种群规模 30，最大迭代次数 500 次。针对各个算法特有的参数，则根据相应的文献进行设置。为了降低随机误差的影响，每个算法均独立执行 30 次。搜索范围为[-100,100]。将最优值作为算法的全局寻优潜力指标、平均值作为算法收敛精 度指标、标准差作为收敛结果的稳定性指标，详细统计结果展示在表 2-3 中。为了更直观的比较结果，展示了结果的箱式图如图 2-3 所示。

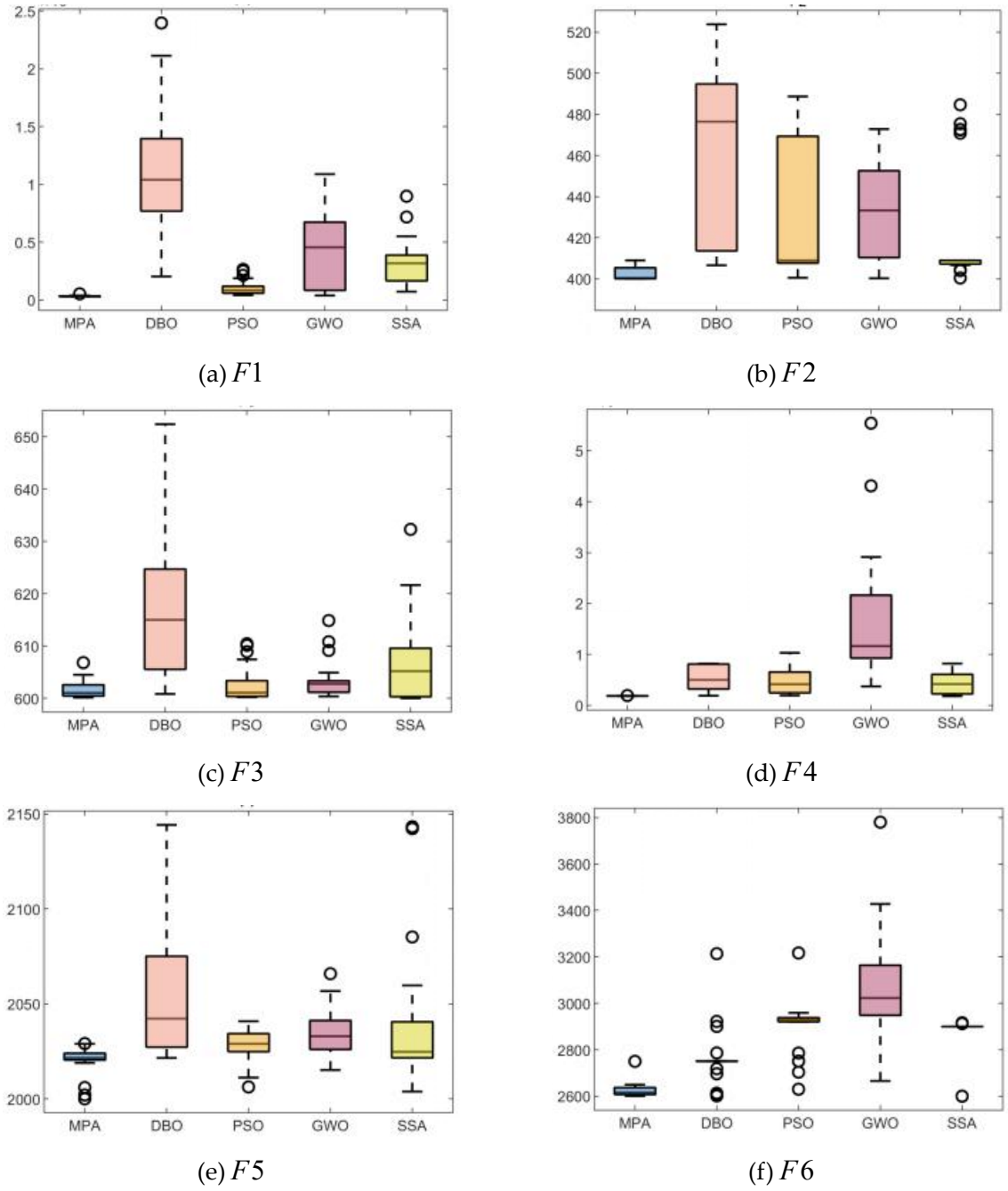
表 2-2 CEC2022 测试函数基本信息

函数名	序号	函数	最小值
单峰函数	$F1$	Shifted and full-rotated Zakharov function	300
基本函数	$F2$	Shifted and full-rotated Rosenbrock's function	400
	$F3$	Shifted and full-rotated noncontinuous Rastrigin's function	800
多峰函数	$F4$	Hybrid function 1 $(D=3)$	1800
	$F5$	Hybrid function 2 $(D=6)$	2000
	$F6$	Composition function 4 $(D=6)$	2600

表 2-3 CEC2022 函数集测试结果统计表

函数	评价指标	MPA	DBO	PSO	GWO	SSA
$F1$	最优值	300.20	2039.56	422.01	397.24	726.03
	标准差	4.77E+01	4.72E+03	6.00E+02	2.98E+02	1.86E+03
	平均值	332.10	11183.32	1031.392	4323.91	3207.51
$F2$	最优值	400.01	406.57	400.49	400.24	400.39
	标准差	3.67E+00	4.47E+01	3.22E+01	2.40E+01	2.57E+01
	平均值	402.63	458.06	430.75	433.28	418.95
$F3$	最优值	807.96	818.41	803.35	805.01	809.95
	标准差	5.69E+00	1.20E+01	6.23E+00	1.31E+01	1.05E+01
	平均值	816.01	839.27	816.67	822.08	832.15
$F4$	最优值	1820.30	1897.44	1941.43	3737.99	1818.41
	标准差	2.51E+01	2.41E+03	2.46E+03	1.16E+04	2.19E+03
	平均值	1858.64	5463.37	4737.92	16295.31	4383.88
$F5$	最优值	2000.06	2021.60	2006.33	2015.24	2003.92
	标准差	6.54E+00	2.88E+01	7.93E+00	1.21E+01	3.24E+01
	平均值	2020.53	2051.20	2028.74	2035.15	2038.17
$F6$	最优值	2601.10	2601.76	2630.30	2666.15	2600.01
	标准差	7.28E+01	1.07E+02	1.05E+02	1.99E+02	1.30E+02
	平均值	2624.86	2761.03	2902.50	3065.13	2831.02

最优值被视作衡量全局寻优能力的指标，平均值用于评价收敛精度，而标准差显示收敛结果稳定性。基于这些指标，得出以下结论：MPA 在所有函数中的平均值和标准差都是最好的，说明算法的收敛精度和稳定性都优于其他算法；其中在 $F2$和$F4$ 函数中，MPA 算法和其他算法结果差距不大，说明算法在基本函数方向还有改进的空间；MPA 算法在函数 $F1$和 $F6$ 中表现效果最好，甚至与 DBO 算法相差两个数量级。



(a) $F1$ (b) $F2$

(c) $F3$ (d) $F4$

(e) $F5$ (f) $F6$

图 2-3 CEC2022 函数集测试结果箱式图

为直观比较 MPA 算法与其他算法在寻优性能上的表现，选择五种算法每次迭代的平均适应度值，并将其绘制成了收敛曲线图，展示于图 2-4。图中的(a)(c)(d)(e)显示，MPA 算法在迭代中相较于其他算法，较少出现在未接近理论最优解时的停滞现象，这点对于 MPA 进行深入寻优非常有利。在这四个函数上，MPA 的迭代性能曲线在到达第 120 代时就已经找到了最优适应度值，彰显了其卓越的寻优精度。然而，MPA 算法也存在明显的不足，尤其是在迭代的初期阶段，其收敛速度并不明显，表明标准 MPA 在早期迭代阶段缺乏足够快的收敛速度。而从图(b)(f)可以观察到，面对更为复杂的函

数时，MPA 的收敛速度与其他算法相比并无显著差异。

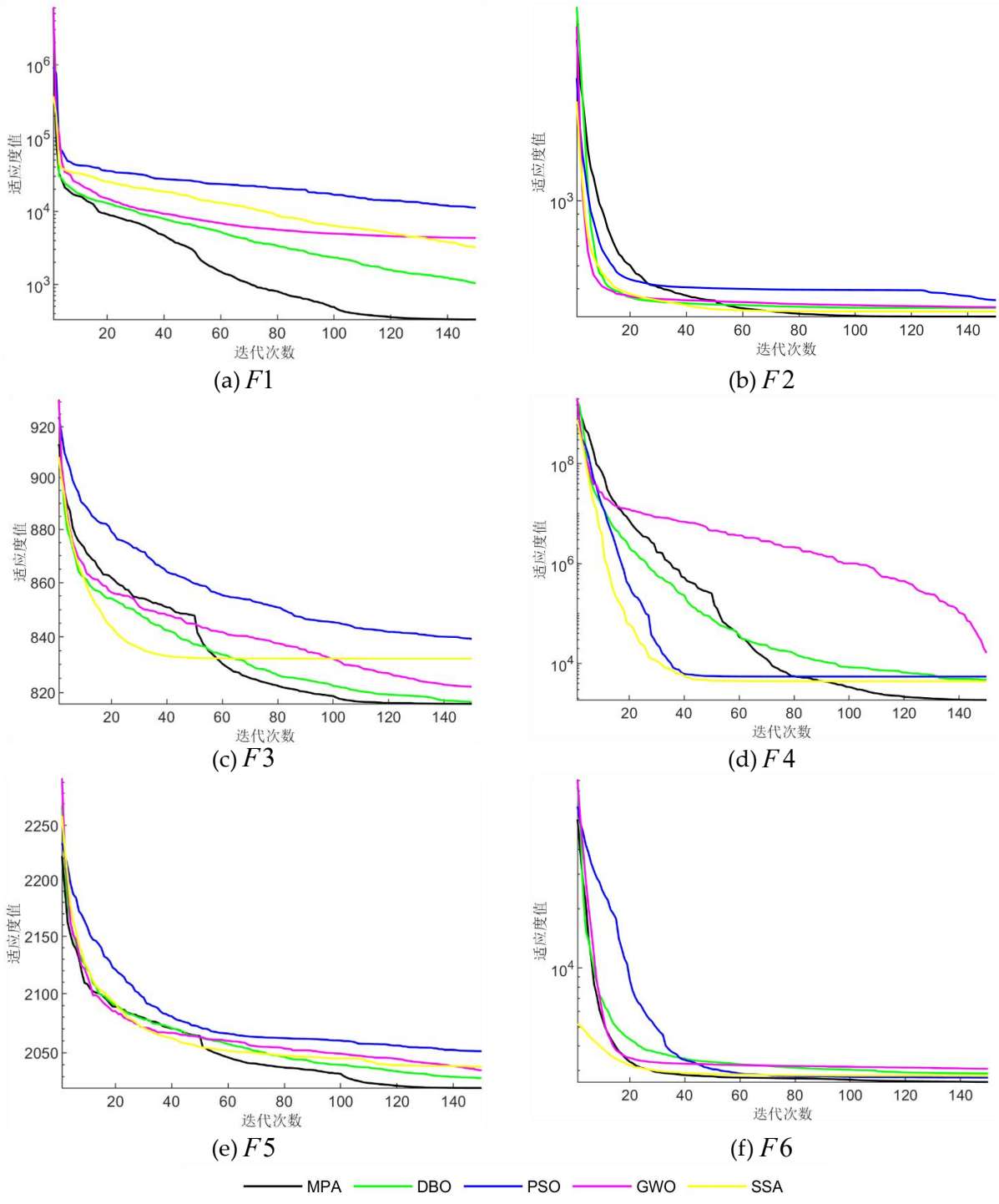


图 2-4 CEC2022 函数集测试结果收敛曲线图

2.4 MPA 的参数效能研究

2.4.1 单因素方差分析的相关概念

方差分析是一种数据分析方法，用于研究实验中的控制变量（称为因素）及其不

同取值（称为水平）对结果指标的影响。单因素方差分析用于评估一个控制变量在不同水平下对样本均值差异的显著性[53]。本节将探讨 MPA 的初始参数设置如何作为方差分析中的因素，以及这些参数设置的不同范围（参数水平）如何影响算法性能。

表 2-4 单因素方差分析表

方差类型	离差平方和	自由度	均方差	F 比值
组间	$SS_t = p\sum_{i=1}^{m}\left(\bar{a}_i - \bar{a}\right)^2$	$df_t = m-1$	$MS_t = SS_t/df_t$	$F = MS_t/MS_e$
组内	$SS_e = \sum_{i=1}^{p}\sum_{j=1}^{m}\left(\bar{a}_i - \bar{a}\right)^2$	$df_e = m(p-1)$	$MS_e = SS_e/df_e$	
总和	$SS_T = SS_e + SS_t$	$df_T = df_t + df_e$		

设实际的因素 O 设置 s 个水平，每个水平有 u 次重复实验，因此第 i 个水平的第 j 次运行的实验结果为 $a_{i,j}\left(i=1,2,\cdots,s; j=1,2,\cdots,u\right)$。单因素方差分析表如表 2-4 所示，其中 SS_T 表示差异程度，df_T 表示相同水平的不同样本之间的差异性。$\bar{a}$ 表示所有样本的平均值。

通过计算 F 值，根据显著性水平 a_0 和 a_1，与自由度 df_t 和 df_e 在 F 分布表里找出对应的上下边界值 $F_{a_0}\left(df_t, df_e\right), F_{a_1}\left(df_t, df_e\right)$。由文献设 $a_0 = 0.05$，$a_1 = 0.01$。显著性检验规则如下：若 $F < F_{a_0}$，则无显著影响；若 $F_{a_0} \le F \le F_{a_1}$，则有一定影响；若 $F > F_{a_1}$，则显著影响。

2.4.2 单因素实验

本实验研究两个参数：种群规模 N 和涡流形成和鱼群聚集装置效应参数 P_{FADs}。设 10 种水平，每种水平下 10 次独立测试(表 2-5)。参数基本值设为 $N=30, P_{FADs}=0.2$。

表 2-5 实验参数水平设置

参数	1	2	3	4	5	6	7	8	9	10
N	20	50	100	150	300	400	500	600	800	1000
P_{FADs}	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

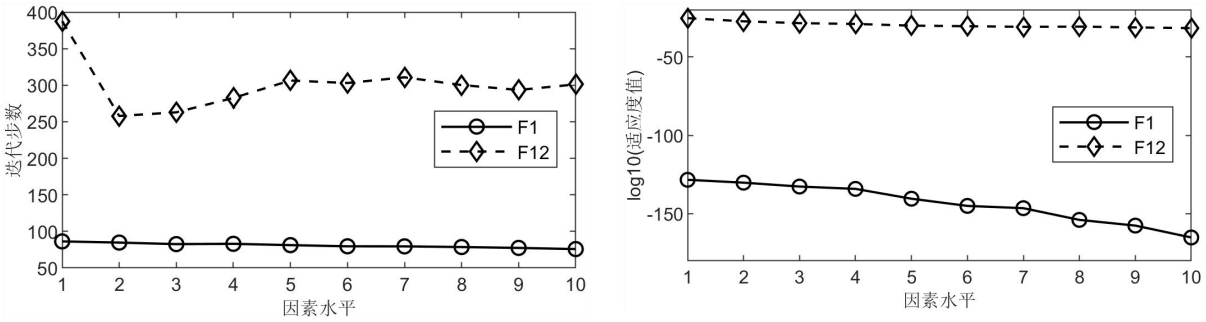
本实验采用 Sphere($F1$)与 Generalized Penalized($F2$)两个测试函数函数。后者在多模函数中更复杂，能有效反映算法性能，因此被选为复杂多模函数的代表，在 $x=(1,1,\cdots,1)$ 时，函数有最小值 $f_{\min}=0$。函数维数 $D=15$。

本实验以算法的收敛速度和精度为评判标准，具体标准如下：收敛速度判别标准：

设置精度阈值（$F1$ 和 $F2$ 函数分别为 $1.00E-10$ 和 $1.00E-03$），当算法达到阈值停止迭代，记录此时的迭代步数作为收敛速度衡量标准，步数越少速度越快。设定最大迭代次数限制（$Max\_Iter=1000$），未达阈值则认为优化失败。收敛精度判别标准：允许算法迭代 1000 代，记录这期间的最优解，解的数值越小，收敛精度越高。

2.4.3 结果分析

图 2-5 至 2-6 中展示了 MPA 在两种参数设置下的迭代曲线结果。横坐标表示参数水平；图(a)展示了收敛速度曲线，即不同参数下的平均迭代步数；图(b)的收敛精度曲线，纵坐标采用对数坐标轴显示平均适应度值。同时，表 2-6 至 2-9 详细列出了这两个参数的单因素方差分析结果。



(a) (b)

图 2-5 N 的水平对算法收敛速度及求解精度的影响

表 2-6 单因素方差分析表（关于 N 的设置水平对收敛速度的影响）

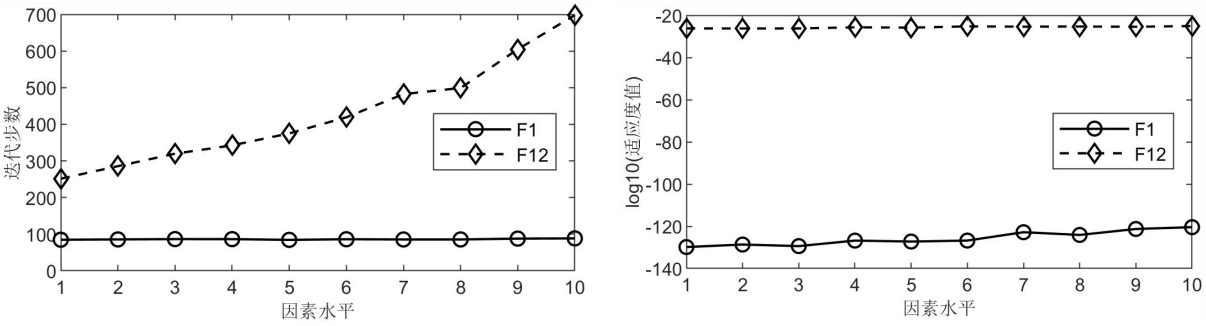
函数	方差类型	平方和	自由度	均方差	F 比值
$F1$	组间	9.82E+02	9	1.09E+02	15.9
	组内	6.18E+02	90	6.96E+00	
	总和	1.60E+03	99		
$F2$	组间	1.13E+05	9	1.26E+04	16.36
	组内	6.93E+04	90	7.70E+02	
	总和	1.83E+05	99		

表 2-7 单因素方差分析表（关于 N 的设置水平对收敛精度的影响）

函数	方差类型	平方和	自由度	均方差	F 比值
$F1$	组间	2.89E-111	9	3.22E-112	3.64
	组内	7.95E-111	90	8.84E-113	
	总和	1.08E-110	99		
$F2$	组间	7.96E-22	9	8.85E-23	38.12
	组内	2.09E-22	90	2.32E-24	
	总和	1.01E-21	99		

从图 2-5 分析可得，随着种群规模 N 的增加，算法在处理 $F1$ 函数时，尽管种群数量的增加对收敛速度的影响微乎其微，但其收敛精度显著提高；在处理 $F2$ 函数时，当种群数量为 50（因素水平 2）时，算法展现出最快的收敛速度，其整体收敛精度呈缓慢提升趋势。这表明增大种群数量对提高收敛速度的作用有限，但能在一定程度上促

进收敛精度的提高。通过查 F 分布表，得出 $F_{0.05}(9,90)=1.99$，$F_{0.01}(9,90)=2.61$，鉴于表 2-6 至 2-7 中所记录的 F 比值皆比 $F_{0.05}(9,90)$ 与 $F_{0.01}(9,90)$ 要大，故因素 N 对单峰函数效果显著。



(a) (b)

图 2-6 P_{FADs} 的水平对算法收敛速度及求解精度的影响

表 2-8 单因素方差分析表（关于 P_{FADs} 的设置水平对收敛速度的影响）

函数	方差类型	平方和	自由度	均方差	F 比值
$F1$	组间	1.40E+02	9	1.55E+01	0.92
	组内	1.52E+03	90	1.69E+01	
	总和	1.66E+03	99		
$F2$	组间	1.85E+06	9	2.06E+05	136.12
	组内	1.36E+05	90	1.51E+03	
	总和	1.99E+06	99		

表 2-9 单因素方差分析表（关于 P_{FADs} 的设置水平对收敛精度的影响）

函数	方差类型	平方和	自由度	均方差	F 比值
$F1$	组间	2.40E-104	9	2.66E-105	2.37
	组内	1.01E-103	90	1.12E-105	
	总和	1.25E-103	99		
$F2$	组间	9.08E-22	9	1.01E-22	3.13
	组内	2.90E-21	90	3.22E-23	
	总和	3.80E-21	99		

从图 2-6 可见，随着 FADs 因子 P_{FADs} 的水平增加，MPA 算法在处理 $F2$ 函数时的收敛速度显著变慢，变化趋势相对稳定。对于其他迭代曲线，变化不甚明显，只有在 $F1$ 的收敛精度上观察到轻微的波动。这表明 P_{FADs} 的水平设置对单峰函数的寻优精度影响不大，而在多峰函数的处理结果上无显著变化，但随着 P_{FADs} 水平的提升，却牺牲了 MPA 在局部搜索上过多的时间。

根据表 2-8 至 2-9 的数据，对于 $F1$ 函数，其收敛速度的 F 比值小于 $F_{0.05}(9,90)$，收

敛精度的 F 比值也小于 $F_{0.05}(9,90)$，说明 P_{FADs} 的大小设置对算法在求解 $F1$ 函数的收敛速度和收敛精度的影响不显著。对于 $F2$ 函数，其收敛速度和收敛精度的 F 比值均超过了临界值 $F_{0.01}(9,90)$，说明 P_{FADs} 的设置显著影响了 MPA 在处理时的收敛速度和精度。

根据本文的参数测试结果，在考虑的参数范围内，对于种群规模 N，由于其在解决单峰问题时对算法性能的影响不显著，因此建议将其设定为默认值 25。对于 FADs 效应的参数 P_{FADs}，建议设定一个较小的值以利于 MPA 的收敛速度提升，推荐值在水平 $1(P_{FADs}=0.1)$ 附近，这样做可以显著提高 MPA 的收敛性能。

2.5 本章小结

本章详细分析了海洋捕食者算法 MPA 的基本原理、实现步骤，并重点研究了其收敛性与参数效能。首先，针对 MPA 收敛性研究的不足，构建并推导了基于 Markov 链的 MPA 模型，验证了算法的全局收敛性，深化了理论理解。其次，通过方差分析对 MPA 的种群规模、涡流形成和鱼群聚集效应参数进行了效能评估，揭示了参数设置对算法性能的影响规律。这些研究不仅充实了 MPA 的理论基础，也为算法性能提升和应用开发提供了重要指导。

第 3 章 连续型 MPA 的改进及在自适应聚类上的应用

当处理包含有不同密度和尺寸的数据对象的数据集时，通常需要依赖于特定的领域知识来挑选合适的聚类算法。一般情况下，KCM 算法[32]被广泛用于执行预测性聚类。然而，KCM 算法面临着收敛速度缓慢和准确度不高的问题，这些问题可能会降低故障诊断的效率与准确性。因此，本文引入了一种基于改进的海洋捕食者算法(Improved Marine Predation Algorithm，IMPA) 的自适应聚类(AC-IMPA)方法。通过在 UCI 数据集上的测试，证明了本研究提出方法的有效性。此外，本文还探讨了将提出的自动聚类算法应用于机械设备故障诊断的可能性。

3.1 连续型 MPA 的改进：IMPA

MPA 优化算法中，若使用随机因素进行初始化，可能会降低优化效率和鲁棒性。种群在优化中期受其移动速度的限制，从而制约了种群之间的交换空间。当猎物成功觅食时，将转变为捕食者，引导种群走向局部最优解。为了提高算法的搜索和求解能力，需要对捕食者搜索机制进行改进研究。

3.1.1 改进策略

(1)改进点一：折射反向学习策略

反向学习策略(Opposition-based Learning, OBL)[54]是一种常见的用于改善种群初始化质量的常用手段。其思想是考虑当前解及相反的候选解，以获得最优解，反向解 x^* 如下：

$$x^* = lb + ub - x \tag{3-1}$$

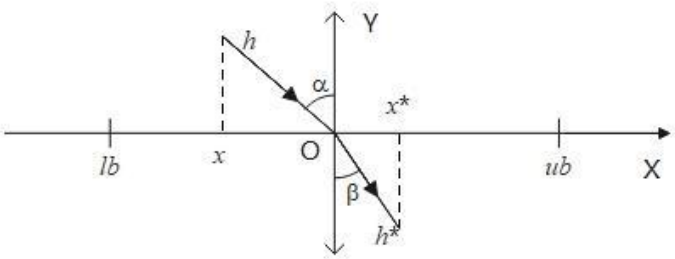


图 3-1 折射反向学习原理图

OBL 手法简单但灵活性低，不利于处理种群的动态变化。为了提高 OBL 的性能，加入光的折射定律，得到基于折射对立的学习(Refraction opposition-based learning,

ROBL)[55]。它可以帮助生成高质量的种群，增加种群的多样性，从而避免在初始迭代中盲目搜索，加速早期收敛。图 3-1 为 ROBL 工作原理图。其中，解的搜索范围为 X 轴上的$[lb,ub]$， Y 轴为法线，点 O 是区间$[lb,ub]$的中点，入射、折射光线的对应长度分别为h和h^*，入射角和折射角分别为α和β。

由图中线段的几何关系，可得折射率z：

$$\begin{cases} \sin\alpha = \left(\dfrac{lb+ub}{2} - x\right) \Big/ h \\ \sin\beta = \left(x^* - \dfrac{lb+ub}{2}\right) \Big/ h^* \end{cases} \tag{3-2}$$

$$z = \frac{\sin\alpha}{\sin\beta} = \frac{h^*\left(\left(lb+ub\right)/2 - x\right)}{h\left(x^* - \left(lb+ub\right)/2\right)} \tag{3-3}$$

设比例因子$f_s = \dfrac{h}{h^*}$，将式（3-3）改为：

$$x^* = \frac{lb+ub}{2} + \frac{lb+ub}{2zf_s} - \frac{x}{zf_s} \tag{3-4}$$

将式(3-4)推广到 MPA 的 D 维空间，得到折射反向解：

$$x_{i,j}^* = \frac{lb_j+ub_j}{2} + \frac{lb_j+ub_j}{2f_s} - \frac{x_{i,j}}{f_s} \tag{3-5}$$

(2)改进点二：正余弦算法

根据式(2-6)和式(2-7)所述的位置更新方程可知，MPA 的过渡阶段是固定的，使得搜索空间广，搜索速度缓慢，可能会提前收敛，无法充分探索搜索空间。正余弦算法(Sine cosine algorithm, SCA)[56]利用正弦函数和余弦函数的振荡切换特性来寻找最优解。该算法具备优化速度快、鲁棒性强等优点。SCA 的核心在于位置更新方法，其位置更新的数学模型如下：

$$x_i = \begin{cases} x_i + r_1 \times \sin r_2 \times \left| r_3 \times gb - x_i \right|, & r_4 < 0.5 \\ x_i + r_1 \times \cos r_2 \times \left| r_3 \times gb - x_i \right| & r_4 \geq 0.5 \end{cases} \tag{3-6}$$

$$r_1 = 1 - \frac{Iter}{Max\_Iter} \tag{3-7}$$

其中，$r_2 \in [0, 2\pi], r_3 \in [0,2], r_4 \in [0,1]$，$r_1$是最关键的参数，它控制着算法的开发和勘探之间的平衡。由于r_1是一个线性递减函数，函数的变化是唯一的，需要更加灵活。为

此，采用文献[57]中的非线性递减函数表达式：

$$r_1 = 2e^{-\frac{Iter}{Max\_Iter}} \tag{3-8}$$

由于 C_F 是非线性下降曲线，采用布朗运动的种群速度比率随着迭代次数的增加而逐渐减小，符合标准迭代规律。因此，只有具有莱维运动的种群才需要改进搜索策略。

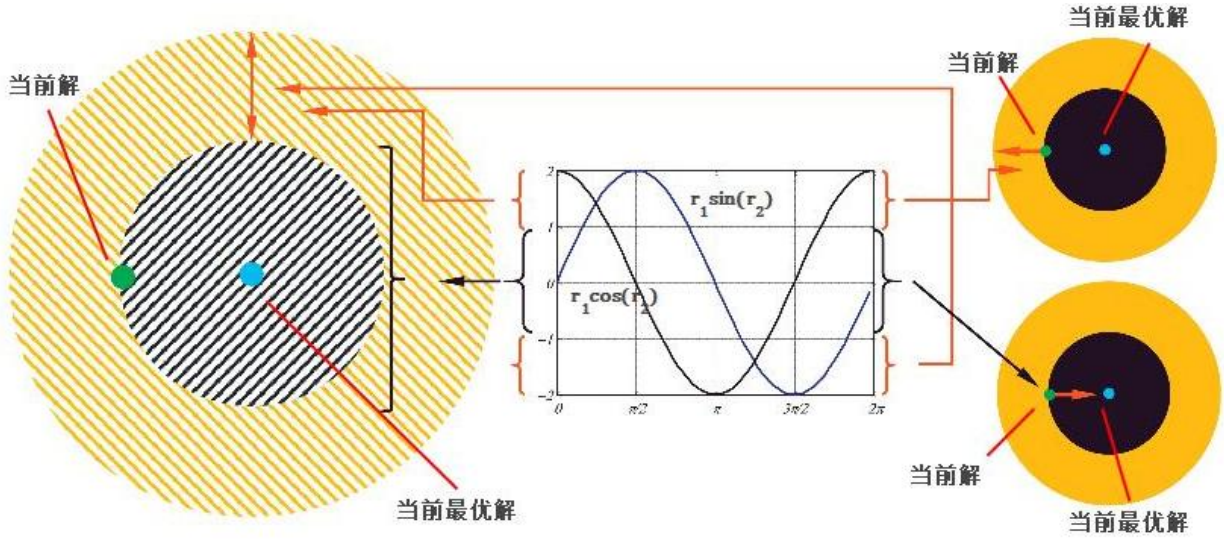


图 3-2 SCA 原理图

(3)改进点三：高斯-柯西变异算子

为了解决 MPA 精度低、易陷入局部最优的问题，需要引入突变策略。为此，采用高斯-柯西变异算子(Gaussian-Cauchy Mutation，GCM)[58]策略。高斯和柯西分布如图 3-3 所示。柯西分布中间小，两端大，搜索步长更大，搜索范围更广。高斯分布中间最高，两端最低，搜索步长更小，提高了局部搜索的能力。高斯分布有均值$(\mu=0)$和单位方差$(\sigma^2=1)$，柯西分布有位置参数$(x_0=0)$和尺度参数$(\gamma=1)$。因此，在 MPA 中采用 GCM 策略来提高跳出局部最优的概率，具体描述如下。

$$GCM=\begin{cases} f_g(x;\mu;\sigma)=\dfrac{1}{\sigma\sqrt{2\pi}}e^{\frac{(x-\mu)^2}{2\sigma^2}} \\ f_c(x;x_0;\gamma)=\dfrac{1}{\pi}\bullet\dfrac{\gamma}{\gamma^2+(x-x_0)^2} \end{cases} \quad -\infty<x<\infty \tag{3-9}$$

$$x=\begin{cases} x\bullet(1+C(1,0)), & R_4\le 1-\dfrac{Iter}{Max\_iter} \\ x\bullet(1+G(0,1)), & R_4>1-\dfrac{Iter}{Max\_iter} \end{cases} \tag{3-10}$$

其中$C(0,1)$是基于柯西分布的随机变量，$G(0,1)$是基于高斯分布的随机变量。在早期迭代中，$1-Iter/Max_Iter$的值很小，捕食者执行柯西分布来扩大搜索范围。在迭代的后期，对于高斯分布，其值增大。这两种策略交替使用，这相当于一个随迭代次数增加的自适应动态参数。它增加了算法跳出停滞状态并在以后加速收敛的机会。

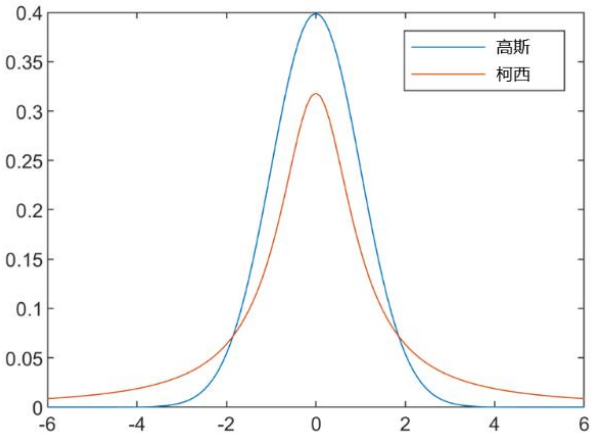


图 3-3 高斯和柯西分布图

3.1.2 IMPA 算法的实现

综合上述阐述，IMPA 算法的步骤总结如下：

（1）设置初始参数：种群数量、搜索空间上下界、算法维度和最大迭代次数；对种群位置进行初始化，然后计算初始种群个体的适应度值；

（2）通过折射方向学习策略（即式（3-5））获得初始种群的折射反向解，并计算反向种群个体的适应度值；

（3）合并并排序两组种群的适应度值，从中选择适应度值较低的前一半种群作为新的猎物种群；

（4）迭代开始：若在迭代前期，根据式(2-5)进行勘探阶段的位置更新；若在迭代中期，前一半种群根据式(3-6)进行正余弦算子的位置跟新，后一半种群根据原 MPA 算法式（2-7）进行位置更新；若在迭代后期，根据式（2-9）正常进行位置更新；

（5）根据式(3-10)对种群位置进行柯西高斯变异；

（6）应用海洋记忆存储和 FADs 效应，即根据式(2-10)更新种群位置；

（7）计算新生种群个体适应度值，根据适应度值构成精英矩阵；

（8）判断是否达到终止条件，若是，则输出最优值及最优解，若不是，则继续执行步骤(2)。

综上所述，得到 IMPA 算法的流程图如图 3-4 所示。

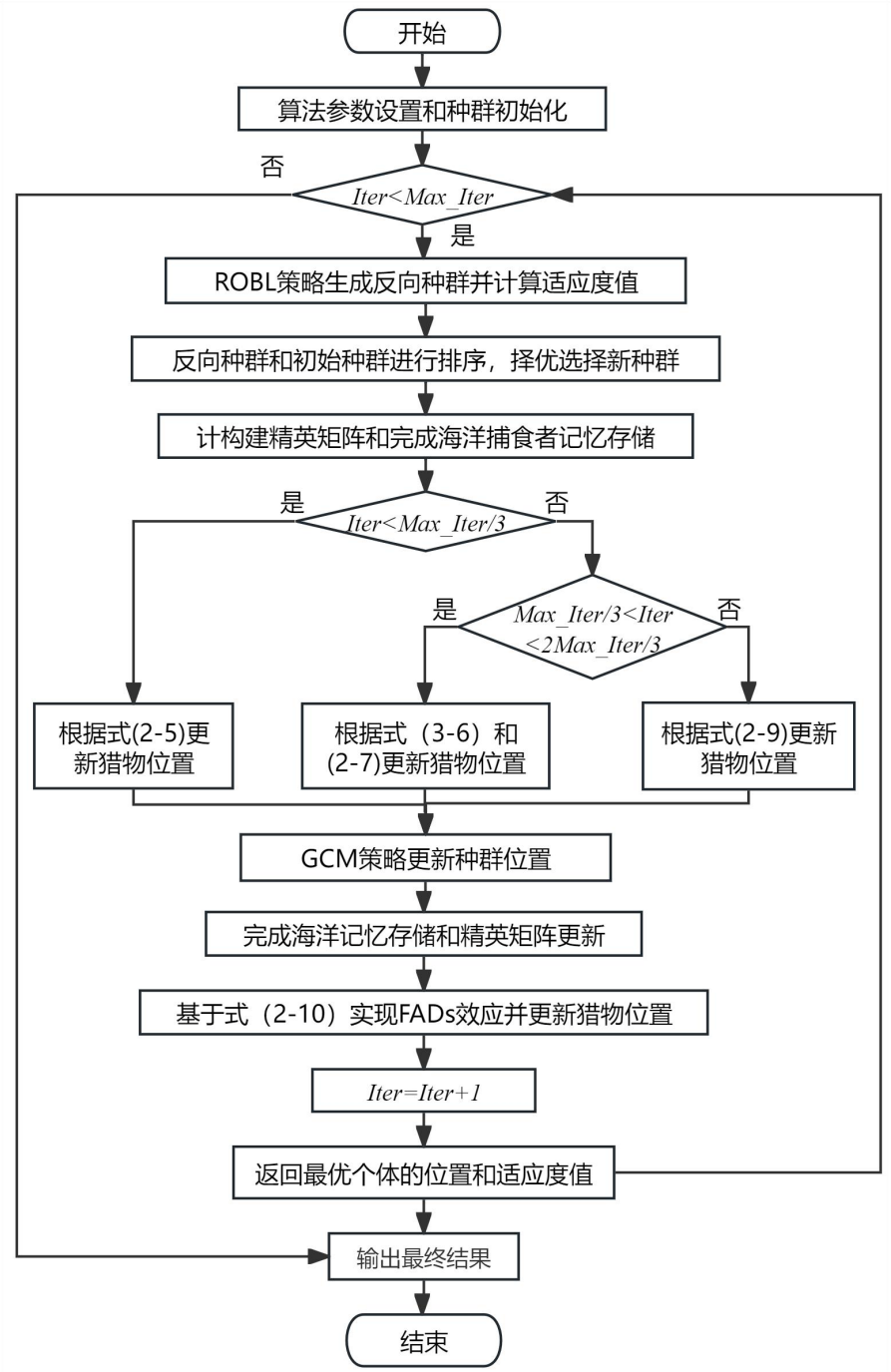


图 3-4 IMPA 算法流程图

3.1.3 算法性能测试

本节将使用测试函数来评估建议的 IMPA 模型的性能。实验在 10 个标准基准函数上进行，这些函数包括复杂单峰函数$(F1-F4)$、多维多峰函数$(F5-F9)$和固定维度多峰函数$(F10)$，详细描述及相关信息见表 3-1。为了保证算法间比较的公平性，将算法的基本参数设置为相同的值， $N=30$、$Max\_Iter=500$。

通过六种算法比较 MPA 性能：MPA、基于准对立学习和 Q 学习的海洋捕食者算法(Quasi-opposition learning and the Q-learning based marine predators algorithm, QQLMPA)[59]、非线性海洋捕食者算法(Nonlinear Marine Predator Algorithm, NMPA)[60]，WOA[22]、鹈鹕优化算法(Pelican Optimization Algorithm, POA)[61]和 PSO[20]共 6 种算法进行比较。

表 3-1 基准函数

函数	函数名	维度	搜索范围	理想值
$F1$	Sphere	30	[−50,100]	0
$F2$	Schwefel 2.22	30	[−10,7]	0
$F3$	Rosenbrock	30	[−30,30]	0
$F4$	Quartic	30	[−1.28,1]	0
$F5$	Rastrigin	30	[−10,5.12]	0
$F6$	Ackley	30	[−32,50]	0
$F7$	Griewank	30	[−600,500]	0
$F8$	Alpine	30	[−10,40]	0
$F9$	Penalized	30	[−50,100]	0
$F10$	Schaffer	2	[−10,10]	0

独立运行 7 种不同的算法各 30 次，记录最佳结果、平均值和标准差。为了清晰展示算法的性能差异，在排序时将首先按照平均最佳适应度值的顺序排列。如果平均最佳适应度值相同，则按照标准差进行排序。测试结果见表 3-2。为了更好地对比结果，图 3-5 展示了测试函数的迭代收敛情况。

$F1-F4$ 都是单峰函数，这类函数仅有一个全局最优解，主要用于测试算法的开发能力。从表 3-2 和图 3-5 的实验结果可以看出，对于两个单峰测试函数 $F1$、$F2$ 和一个多峰测试函数 $F8$，IMPA 和 NMPA 在三种不同的评价指标下均能达到理论最优值。从图中可以看出，IMPA 可以在 150 次迭代前达到理想值，说明了 ROBL 策略的有效性。然而，对于 $F3$、$F4$，寻找全局最优解是相当具有挑战性的。$F3$ 被称为香蕉型山谷函数，$F4$ 的形状为抛物线。两者都有很多局部优化，容易导致算法陷入局部最优，使优化搜索停滞不前。

$F5-F9$ 代表的是多峰函数，这类函数有多个局部极小值或极大值，随着问题的维度增加时，局部极值的数量也会增加，因此，这些函数通常用于测试算法的搜索能力和是否容易落入局部最优的情况。在 $F5$ 和 $F8$ 中，除 WOA 和 PSO 外，算法的三个指标均为 0。$F6$ 的特征表示外部区域近乎平坦，而中心有一个较大的孔洞。这种特性可能会导致优化算法陷入许多局部极小值之中的风险。三种改进的 MPA 算法的均值为 8.88E-16，均值差为 0。然而，由图可知 IMPA 可以在早期迭代中更快地接近理想的最

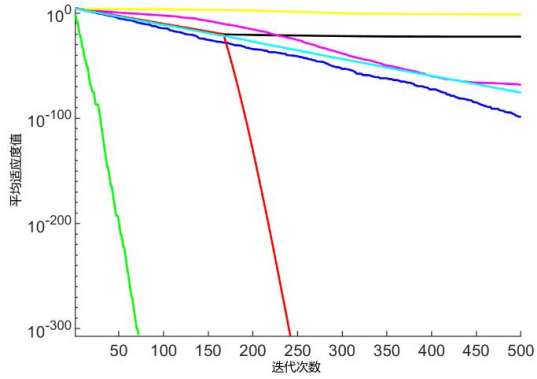
优值。

表 3-2 函数的优化结果对比

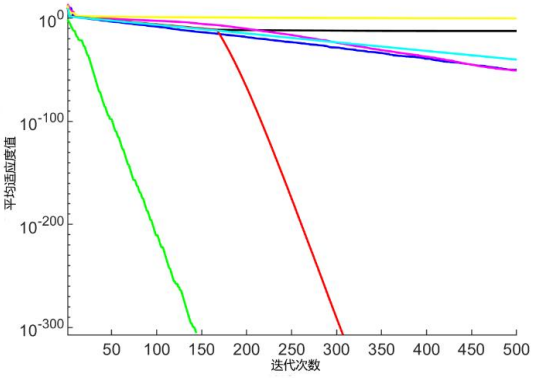
函数	指标	MPA	NMPA	QQLMPA	POA	WOA	PSO	IMPA
$F1$	最优值	1.09E-24	0	1.27 E-110	2.21E-116	2.30E-87	2.30E-02	0
	平均值	4.86E-23	0	2.76 E-76	3.28E-99	9.93E-69	6.84E-02	0
	标准差	9.74E-23	0	4.30E-76	1.71E-98	5.44E-68	2.82E-02	0
	排序	5	1	3	2	4	6	1
$F2$	最优值	1.07E-14	0	5.53 E-56	4.07E-59	2.87E-56	4.89E-01	0
	平均值	4.56E-13	0	8.33 E-41	7.45E-51	1.49E-51	7.75E-01	0
	标准差	4.00E-13	0	1.70E-40	3.50E-50	3.91E-51	2.21E-01	0
	排序	5	1	4	3	2	6	1
$F3$	最优值	2.45E+01	2.44E+01	2.43 E+01	2.64E+01	2.72E+01	2.92E+01	2.37 E+01
	平均值	2.53E+01	2.51E+01	2.57 E+01	2.80E+01	2.79E+01	1.03E+02	2.47 E+01
	标准差	4.34E-01	6.95E-01	7.39E-01	7.84E-01	4.24E-01	1.01E+02	6.78E-01
	排序	3	2	4	6	5	7	1
$F4$	最优值	1.19E-04	4.07 E-07	1.13E-04	3.42E-05	1.35E-04	7.87E-02	1.41E-06
	平均值	1.31E-03	1.29E-04	4.96 E-04	1.95E-04	1.98E-03	1.74E-01	5.55 E-05
	标准差	7.97E-04	1.16E-04	2.59 E-04	1.07E-04	1.64E-03	7.56 E-02	4.60 E-05
	排序	5	2	4	3	6	7	1
$F5$	最优值	0	0	0	0	0	3.96E+01	0
	平均值	0	0	0	0	0	6.91E+01	0
	标准差	0	0	0	0	0	1.93E+01	0
	排序	1	1	1	1	1	2	1
$F6$	最优值	2.53E-13	8.88E-16	8.88 E-16	8.88E-16	8.88E-16	8.74E+00	8.88E-16
	平均值	1.33E-12	8.88E-16	8.88E-16	3.61E-15	4.80E-15	1.35E+01	8.88 E-16
	标准差	8.76E-13	0	0	1.53E-15	3.00E-15	1.74E+00	0
	排序	4	1	1	2	3	5	1
$F7$	最优值	0	0	0	0	0	1.49E+01	0
	平均值	0	0	0	0	0	3.93E+01	0
	标准差	0	0	0	0	0	2.01E+01	0
	排序	1	1	1	1	1	2	1
$F8$	最优值	0	0	0	0	0	1.49E+01	0
	平均值	0	0	0	0	0	3.93E+01	0
	标准差	0	0	0	0	0	2.01E+01	0
	排序	1	1	1	1	1	2	1
$F9$	最优值	3.28E-09	9.68E-05	1.15 E-04	5.92E-02	4.54E-03	7.85E+00	5.02E-07
	平均值	5.83E-05	8.37E-03	1.12 E-02	1.82E-01	2.60E+02	1.48E+01	9.12E-06
	标准差	2.88E-04	6.56E-04	8.61 E-03	1.09E-01	2.57E-02	5.36E+00	1.38 E-05
	排序	2	3	4	5	7	6	1
$F10$	最优值	0	0	0	0	0	3.33E-16	0
	平均值	1.95E-08	0	0	0	4.21E-03	6.48E-03	0
	标准差	1.06E-07	0	0	0	4.90E-03	4.66E-03	0
	排序	2	1	1	1	3	4	1

$F10$ 是一个具有多个峰值的固定维度函数，所以它不存在局部最小值，因此根据表 3-2 的结果可以看出几种算法的结果非常相似。在 $F10$ 的最优值之外存在两个圆形尖峰，导致局部最优位置与全局最优位置之间具有较大的波动。由图可知，IMPA 可以快速找到 $F10$ 的最优点。一般来说，对于其他不同类型的测试函数，IMPA 在大多数时候

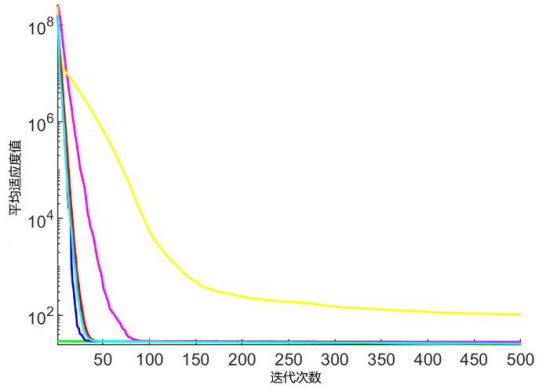
都处于迭代曲线的底部。结果表明，该算法具有较高的收敛效率，验证了算法优化策略的有效性。



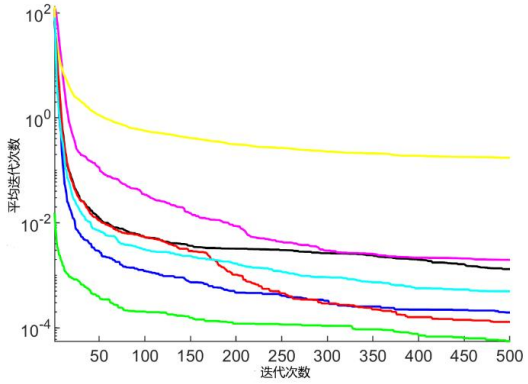
(a) $F1$



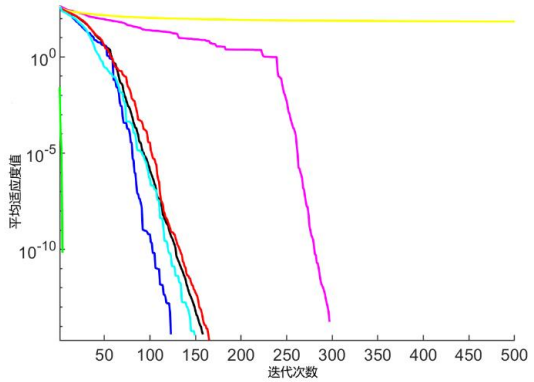
(b) $F2$



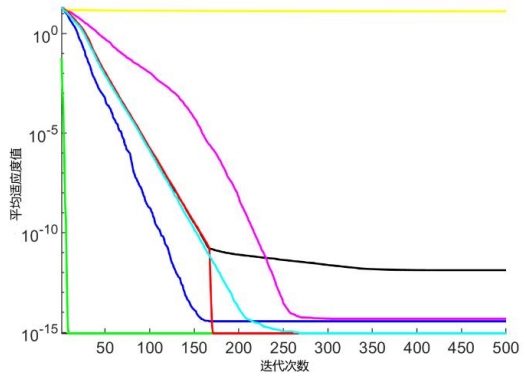
(c) $F3$



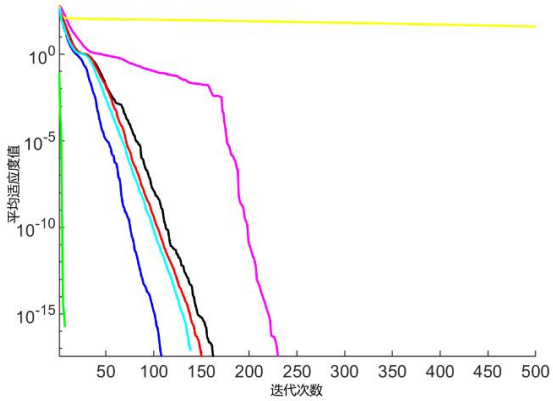
(d) $F4$



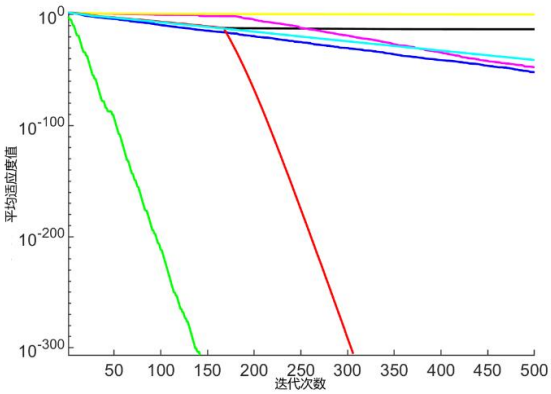
(e) $F5$



(f) $F6$



(g) $F7$



(h) $F8$

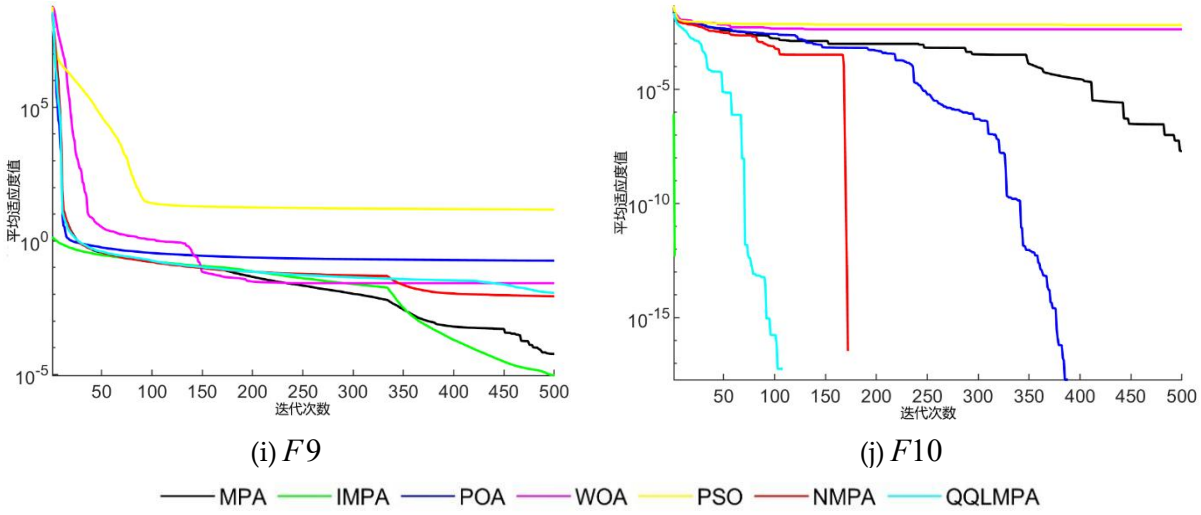


图 3-5 函数的收敛图

鉴于元启发算法的随机性，有必要进行一系列统计实验以确保数据的合理性。为验证两组数据的显著性差异，通常可以采用 Wilcoxon 秩和检验方法进行统计学比较[62]。当 $p<0.05$ 时，表示两种算法的数据之间存在显著差异。在表 3-3 中列出了以 IMPA 为基准和其他算法的 p 值比较结果，结果显示在 10 个基准函数中，所有算法的 p 值均低于 5%。这表明该算法在统计上的优越性显著。即 IMPA 算法的收敛精度高于其他算法。

表 3-3 基准函数的 p 值测试结果

函数	MPA	NMPA	QQLMPA	POA	WOA	PSO
$F1$	1.26E-83	1.89E-41	1.26E-83	1.26 E-83	1.26 E-83	1.26 E-83
$F2$	1.26 E-83	4.36E-52	1.26 E-83	1.26 E-83	1.26 E-83	1.26 E-83
$F3$	8.04E-59	1.26E-83	1.26 E-83	1.26 E-83	1.26 E-83	1.26 E-83
$F4$	1.26E-83	1.26E-83	1.26E-83	1.24 E-83	1.26 E-83	1.26 E-83
$F5$	1.12 E-27	7.91E-29	2.30E-26	6.32 E-22	1.88 E-50	1.26E-83
$F6$	1.16E-83	1.20E-29	1.52 E-45	3.49 E-90	6.56 E-85	1.26 E-83
$F7$	1.69 E-28	1.58E-26	1.48 E-24	1.28 E-19	1.20 E-39	1.26 E-83
$F8$	1.26 E-83	6.35E-52	1.26 E-83	1.26 E-83	1.26 E-83	1.26 E-83
$F9$	8.60 E-07	2.55E-29	6.74 E-63	1.26 E-83	3.34 E-25	1.26 E-83
$F10$	1.26 E-83	5.62E-30	1.86 E-19	2.49 E-65	1.25E-83	1.26 E-83
+/-/=	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0

3.2 .基于 IMPA 的自动聚类：AC-IMPA

3.2.1 KCM 算法

K-means 算法是常见的聚类分析算法之一。KCM 的目标是将数据集分成 K 个独立的部分，以最小化每个对象与其所属聚类之间的欧式距离，其表达式如下：

$$d\left(e_i,C_j\right)=\sqrt{\sum_{k=1}^{D}\left(e_{i,k}-C_{j,k}\right)^2}=\left\|e_i-C_j\right\| \tag{3-11}$$

假设一个数据集 Y 包含 N 个未标记的样本，记作 $Y=\{y_1,y_2,\cdots,y_N\}$。标准 KCM 算法的主要步骤通常表示如下：

（1）设定初始聚类中心：从 N 个数据项中随机选择 K 个作为初始的聚类中心集合 $\{C_i\}$；

（2）重新分配数据点：针对每个数据点，通过计算其与 K 个初始聚类中心的欧式距离，根据最小距离原则将数据点分配至相应的聚类中心；

（3）更新聚类中心：对每个聚类计算所有维度的均值，确定新的聚类中心。循环执行步骤 2 和步骤 3，直至聚类中心的更新不再显著或低于预设的阈值，此时确认并输出最终的聚类中心。

在进行聚类分析时，如果选择的初始聚类中心不恰当，可能会导致得到局部最优解。更新聚类中心的方法是根据以下计算公式进行。

$$C_j=\frac{1}{C_j}\sum_{i=1}^{N}e_i,\ j=1,2,\cdots,K \tag{3-12}$$

因此，KCM 的初始化具有随机性，可能以噪声或异常值为初始中心，导致每次聚类结果差异较大，从而降低算法的鲁棒性。同时，简单的更新过程也容易导致 KCM 陷入局部最优解。

3.2.2 基于 IMPA 算法的自动聚类

为了实现自动聚类，需要设置聚类中心的激活阈值或切换向量。设置一个常量，若激活阈值大于该值，则表明该质心是有效的。假设有 K 个中心，则中心有 $K\times D$ 个元素，激活阈值部分有 $K\times 1$ 个元素，则总共 $K\times(D+1)$ 个元素。聚类质心与激活阈值的关系如下所示：

$$\left.\begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,D} & a_1 \\ m_{2,1} & m_{2,2} & \cdots & m_{2,D} & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{K,1} & m_{K,2} & \cdots & m_{K,D} & a_K \end{bmatrix}\right\} K \tag{3-13}$$

其中，$m_i(i=1,2,\cdots,K)$ 为第 i 个质心向量，，$a\in[0,1]$ 为激活阈值。在本章中，阈值设置为 0.4；若 $a<0.4$，则该聚类中心不采用。图 3-6 显示了如何激活或停用聚类中心。

在自动聚类分析中，为了评估聚类的质量，通常会使用聚类有效性指标(Cluster Validity Index，CVI)作为目标函数。目标函数(即 CVI) 所追求的是在数值上越小越好，

表示聚类结果的紧凑性和分离度越理想。在自动聚类中，如果指定了聚类的数量，则前面的目标函数(式(3-11))可以很好地工作；但如果簇的数量未知，则必须使用另一个目标函数。迄今为止，研究人员已经为许多聚类提供了有效性指标，如 Davis-Bouldin 指数(DB)[63]、Calinski-Harabasz 指数(CH)[64]、Pakhira Bandyopadhyay Maulik 指数(PBM)[65]、Silhouette[66]和 Compact Separation 指数(CS)[67]。图 3-7 为基于 IMPA 算法的自适应聚类流程图。

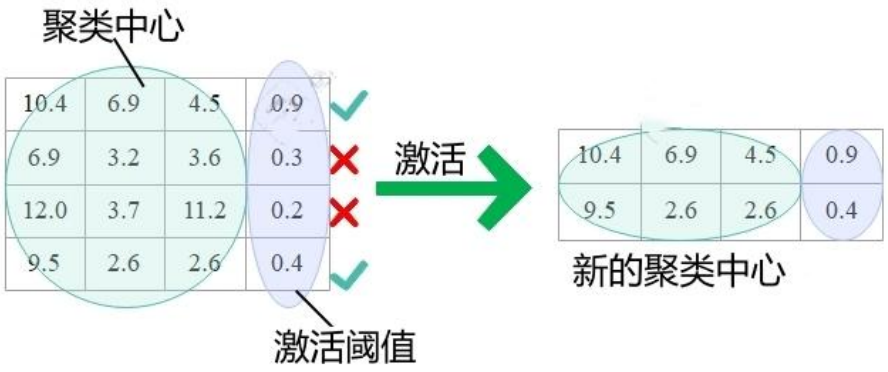


图 3-6 群集质心激活或不激活的图示

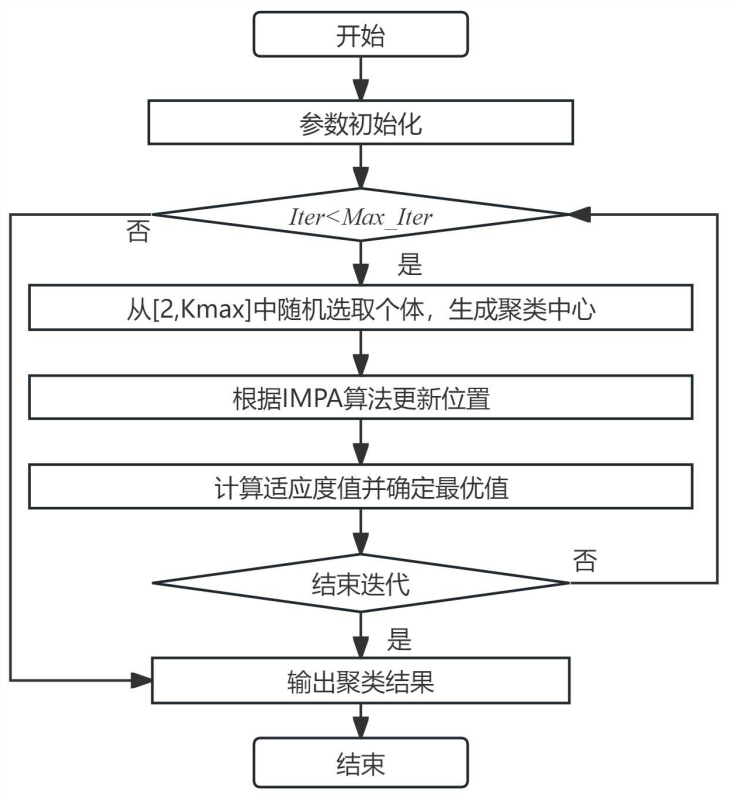


图 3-7 自适应聚类流程图

CS 是指聚类内的散点与聚类之间的分离之和的比率。CS 值越低，分离效果越好，聚类越紧凑。根据文献，在处理不同密度、尺寸和大小的聚类方面，它是一种高效的 CVI[67]，因此使用 CS 指数作为目标函数。CS 表示如下：

$$CS=\frac{\sum_{i=1}^{K}\left[\frac{1}{Q_i}\sum_{e_i\in Q}\max Y_j\in Q\left\{d\left(Y_i,Y_j\right)\right\}\right]}{\sum_{i=1}^{K}\left[\min_{j=K,j\neq i}\left\{d\left(y_i,y_j\right)\right\}\right]} \tag{3-14}$$

其中，簇 C 的数据点个数表示为 Q，簇内散射 Y_i 与簇间分离 Y_j 的距离表示为函数 $d\left(Y_i,Y_j\right)$。数据点与其质心之间的距离表示为 $d\left(y_i,y_j\right)$。

3.2.3 基于 IMPA 的自适应聚类在常见数据集的实验

为了验证 AC-IMPA 的整体性能，我们选择了 KCM、模糊 C 均值算法(Fuzzy C-means, FCM)[68]、MPA、WOA[22]和 SSA[52]共 6 种算法进行比较。最大迭代次数 100。

表 3-4 数据集详细信息

数据类型	数据名	数据维度	数据数量	分类数	分类组
人工	Data 1	2	299	3	[139,99,61]
UCI	Wine	13	178	3	[59,71,48]
	Balance	4	625	3	[49288288]
	Cancer	9	683	2	[444239]
	Jain	2	373	2	[276,97]
	Thyroid	5	215	3	[150,35,30]
	Haberman	3	306	2	[225,81]

首先使用了 7 个常见数据集对算法进行实验验证，包括 1 个人工数据集和 6 个 UCI 数据集。表 3-4 给出了 7 个数据集的基本信息，其中 UCI 数据集是来自不同领域的真实经典数据集，所选择的数据维度是不同的。

表 3-5 簇数

数据名	指标	KCM	AC-IMPA	AC-WOA	AC-SSA	FCM	AC-MPA
Data 1	平均值	3	3	2.9	3.22	6.4	3.04
	标准差	0	0	0.36	0.89	0.55	0.20
Wine	平均值	8.54	3.12	5.62	4.38	7.8	3.7
	标准差	1.18	0.48	1.61	1.89	0.84	1.11
Balance	平均值	8.96	2.78	5.74	6.8	8.8	3.32
	标准差	0.97	1.25	3.51	2.89	1.79	2.24
Cancer	平均值	2	2	2.02	2.86	2	2.1
	标准差	0	0	0.14	1.98	0	0.30
Jain	平均值	7.86	2	2	3.32	8.4	2.34
	标准差	0.83	0	0	1.43	0.55	0.87
Thyroid	平均值	8.58	3.08	2.54	2.76	8	3.16
	标准差	1.37	0.40	1.09	1.04	0	0.47
Haberman	平均值	7.78	2	4.3	3.52	4	2
	标准差	1.54	0	2.13	1.50	0	0

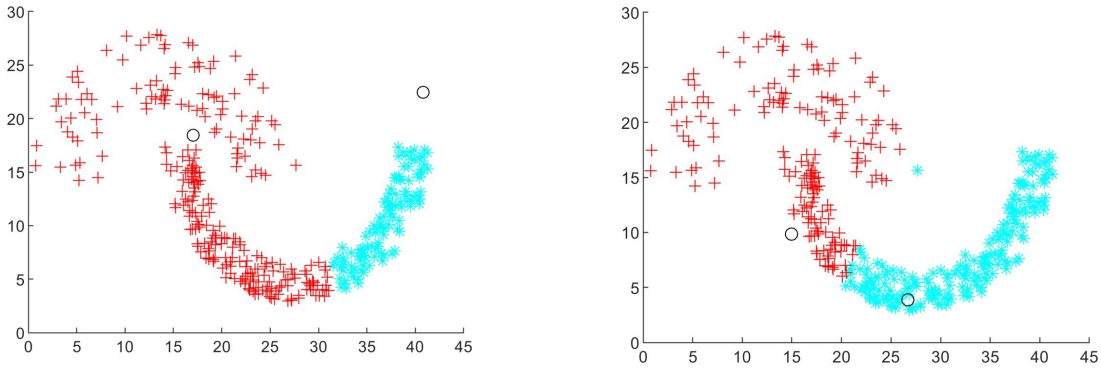
表 3-5 和表 3-6 显示了基于 CS 的这些算法 50 次独立运行的结果。表 3-5 显示了 CS 确定的簇数和标准差，表 3-6 显示了 CS 测量的平均值和标准差。本文首先比较聚类数，然后比较 CS 值。在 Cancer 数据集中，K-means 和 AC-IMPA 的簇数与实际一样，

具有高度的稳定性。在 Jain 数据集中，AC-IMPA 和 AC-MPA 具有最好的聚类数。在其余 5 个数据集中，AC-IMPA 表现最好。CS 表的值 AC-IMPA 在 4 个数据集的平均值最好，Cancer、Jain、Thyroid、Haberman 等数据集排名较高，说明 AC-IMPA 的性能优越。

表 3-6 CS 值

数据名	指标	KCM	AC-IMPA	AC-WOA	AC-SSA	FCM	AC-MPA
Data 1	平均值	0.6059	0.6059	0.6428	0.6059	1.2030	0.6057
	标准差	0	0	6.05E-02	0	3.17E-02	1.04E-03
Wine	平均值	0.6095	0.3960	0.4527	0.3976	0.7313	0.3947
	标准差	1.27E-01	1.53E-03	2.52E-02	9.81E-03	1.20E-02	2.55E-03
Balance	平均值	1.35078	0.9694	1.1229	1.1121	1.7521	1.0058
	标准差	3.22E-02	1.54E-01	2.72E-01	2.10E-01	1.22E-01	1.83E-01
Cancer	平均值	1.0973	0.9166	1.02334	0.9617	1.0906	0.9592
	标准差	4.99E-04	1.60E-01	1.19E-01	1.35E-01	0	1.23E-01
Jain	平均值	0.8725	0.6547	0.65467	0.6481	0.8764	0.6413
	标准差	1.47E-02	5.83E-16	5.61E-16	1.95E-02	6.23E-03	3.56E-02
Thyroid	平均值	0.8873	0.2924	0.3177	0.3019	1.1892	0.2927
	标准差	8.13E-02	6.15E-03	2.87E-02	9.15E-03	0	4.75E-03
Haberman	平均值	1.1072	0.4674	0.5620	0.4825	1.4517	0.4674
	标准差	7.44E-02	5.62E-16	9.09E-02	3.89E-02	0	5.62E-16

图 3-8 显示了 MPA 和 IMPA 对人工数据的聚类结果。此数据为二维数据，有利于直观地观察聚类结果，数据的形状像两个月亮，代表实际的 2 个分类。对于两月亮相交处的数据，AC-IMPA 比 AC-MPA 有更好的分类效果，图中的圆圈表示聚类中心，AC-IMPA 的聚类中心更合理。



(a) MPA　　(b) IMPA

图 3-8 人工数据上 AC-MPA 和 AC-IMPA 的聚类结果

3.4 改进算法在西交大滚动轴承数据集自动聚类中的应用

3.4.1 数据来源及划分

数据采用西交大滚动轴承(XJTU-SY)实验数据集[69]。实验平台如图 3-9 所示，实验轴承为 LDK UER204 滚动轴承，实验共设计了 3 类工况，如表 3-7 所示。采样频率设

为 25.6 kHz，采样间隔设为 1 min，每次采样时间设为 1.28 s。

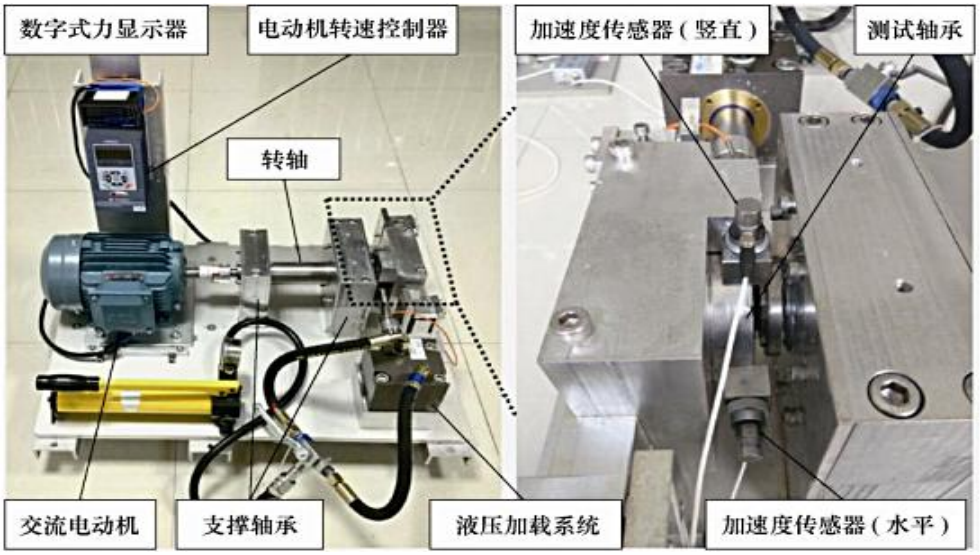


图 3-9 轴承加速寿命试验台

表 3-7 轴承分类工况

工况编号	1	2	3
转速$(r/\min)$	2100	2250	2400
径向力(kN)	12	11	10

本节选择工况 1，其中包含五个分类，使用数据集中的 Bearing1_1、Bearing1_4 和 Bearing1_5。数据详细信息如表 3-8 所示。

表 3-8 轴承数据信息

数据集	样本数量	实际寿命	故障位置
Bearing1_1	123	2 h 3 min	外圈
Bearing1_4	122	2 h 2 min	滚动体
Bearing1_5	52	52 min	外圈、内圈

3.4.2 实验结果分析

表 3-9 详细展示了 6 种算法在 XJTU-SY 滚动轴承数据集上的自适应聚类值和 CS 值。从表中可以观察到，只有 AC-MPA、AC-IMPA 和 AC-SSA 算法的聚类数接近数据集的最优类别数。当 $K=3$ 时，三种算法的适应度函数变化曲线如图 3-10 所示，这些曲线进一步验证了改进算法在求解聚类问题时的高精度优势。为了全面评估算法的有效性，表 3-10 记录了这三种算法在 $K=3$ 设定下，独立运行 10 次的所有样本识别率结果。从统计结果来看，AC-IMPA 算法在这类复杂的聚类分析任务中具有最高的识别率，显示出其优越的聚类分析能力。具体数值上，AC-IMPA 的识别率最高，显著优于其他算法。综上所述，本文提出的 AC-IMPA 算法在 XJTU-SY 数据集上的应用表现了高精度及高稳定性的聚类能力。

表 3-9 聚类结果

	指标	KCM	AC-MPA	AC-IMPA	AC-WOA	AC-SSA	AC-CM
聚类簇数	平均值	9.44	3	2.98	2.02	2.9	9.6
	标准差	0.81	0.95	0.47	0.14	1.90	0.52
CS 值	平均值	1.5089	0.6595	0.6335	0.7602	0.7277	1.6050
	标准差	3.51E-02	7.42E-02	5.72E-02	1.82E-02	4.25E-02	1.66E-02

表 3-10 三种算法识别率结果的比较

类别	算法		
	AC-MPA	AC-IMPA	AC-SSA
Bearing1_1	81.25%	100%	56.25%
Bearing1_4	100%	100%	12.50%
Bearing1_5	68.75%	100%	81.25%
整体	83.33%	100%	56.94%

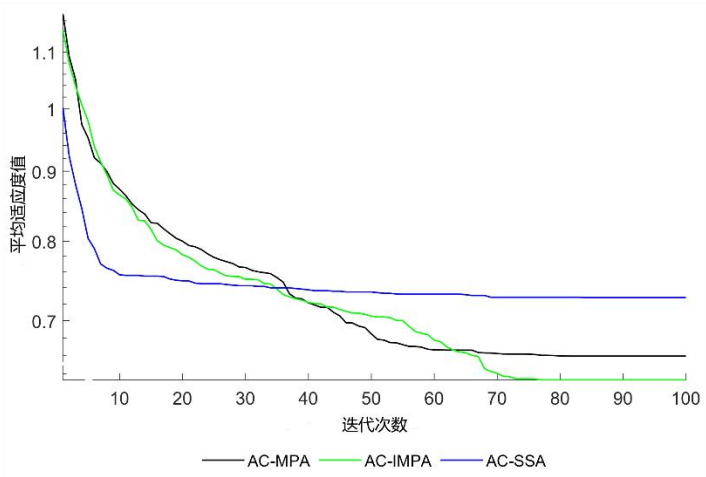


图 3-10 三种算法的收敛图像

3.5 本章小结

本章介绍了基于改进海洋捕食者算法的自适应聚类分析方法。首先，针对 MPA 收敛精度低和速度慢的问题，提出连续型改进的 IMPA 算法，引入折射反向学习策略、正余弦算子和高斯柯西变异算子，增强了全局搜索能力和收敛精度。其次，针对传统 K-means 算法需预设聚类数和聚类中心不稳定的局限，本文利用 IMPA 提出了一种新型自适应聚类方法，结合 CS 指标快速确定最佳聚类数和聚类中心，实现了无监督数据的精确分类。通过测试数据集和机械故障诊断案例的验证，证明了该方法的有效性和优越性。

第 4 章 离散型 MPA 的改进及在特征选择中的应用

在数据挖掘和模式识别领域，特征选择被视为一项至关重要的数据预处理技术。现实世界中获取的数据特征是否有效往往未知，依赖单一信号处理技术进行特征选择常无法高效准确地表征数据状态。此外，若不对特征维度进行优化而直接进行模式识别，可能会遭遇“维数灾难”，这将严重影响特征选择的效率和模型的性能。为应对这些挑战，特别是在面对复杂和多样化的特征时，采用基于智能识别的高效特征选择方法来消除多余特征显得尤为重要，这类方法旨在通过智能算法自动识别并剔除冗余或无关紧要的特征，最终得到一个既能准确反映数据本质特性又具有较低维度的特征子集。本文提出了一种基于离散的增强型 MPA(Binary enhanced Marine Predator algorithm，BEMPA)的特征选择算法，并应用在高维 UCI 数据集和江南大学轴承故障数据集上。

4.1 特征选择方法

4.1.1 K 邻近算法

本文采用的 FS 方法是 K-最近邻算法。由于 KNN 算法的简单性、模型训练时间快和易于理解的特点，它在机器学习的诸多领域中得到了广泛的应用。其核心原理是：设一个待分类的样本 y，通过距离公式计算该样本与其他样本的距离，将 y 视作一个超立方体，里面只包含与它距离最近的 K 个训练样本，统计这 K 个样本中出现次数最多的标签类别，据此判定样本 y 的类别。图 4-1 为 KNN 算法原理图。本文选择的距离公式为欧氏距离见公式(3-11)。

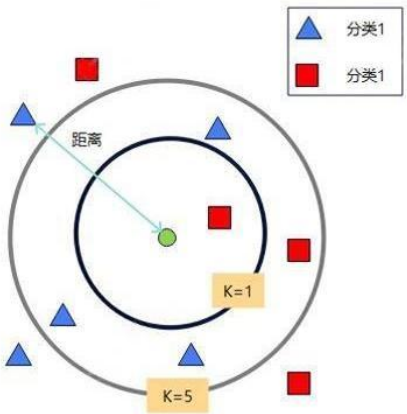


图 4-1 KNN 原理图

4.1.2 评估指标

本章所有的数据集按照保留策略被随机划分为 80%的训练集和 20%的测试集。为了确保结果具有统计学上的意义，每种算法将独立执行 30 次。同时，种群规模设为 30，迭代次数设定为 100。评估算法性能的标准包括：

（1）特征子集的规模：量化算法在 FS 过程中所选特征的总数。

（2）分类的准确率：算法正确预测的次数与总预测次数的比例百分比。

（3）适应度函数值：是衡量算法解质量的指标，同时也量化了算法的收敛性能。FS 的主要目标是在保持分类精度不降低或提升的前提下，最小化所选取的特征子集的数量。这是一个涉及多个目标的优化问题，为了更便于求解，通过加权的方式将适应度函数转换为一个单一目标的问题。因此，可以根据如下公式来定义适应度函数[70]：。

$$f\left(x\right)=\gamma_1\cdot E_{rate}+\gamma_2\cdot\frac{\left|R\right|}{\left|N\right|} \tag{4-1}$$

式中，E_{rate}代表所有类别分类正确率的平均值，$\left|R\right|$为数据集中选中特征的数量，$\left|N\right|$则是数据集中所有特征的数量，γ_1和γ_2是权衡分类精度与特征子集规模的权重参数，满足$\gamma_1+\gamma_2=1$。鉴于提升分类器的准确率是首要目标，初始权重值设定为$\gamma_1=0.99$和$\gamma_2=0.01$。

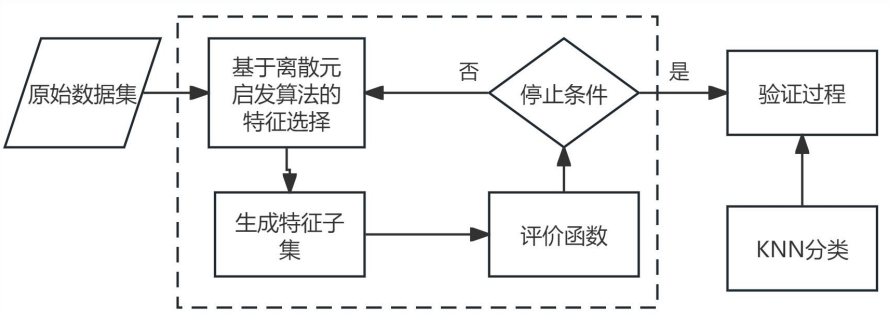


图 4-2 特征选择流程图步骤

4.2 离散型 MPA 算法的改进：BEMPA

4.2.1 改进策略

(1)改进点一：全局最优个体

在群体智能优化算法的搜索过程中，维持种群多样性与加快收敛速度之间的平衡是至关重要的。在基本 MPA 算法的早期迭代中，尽管种群分布较为广泛，显示出较高

的多样性，但其收敛速度相对较慢。因此，提高算法在开发阶段的收敛速度成为改进后算法的一个主要目标。在许多情况下，利用全局最优个体的信息能够有效加速群体智能优化算法的搜索过程[71]。但在原始 MPA 算法中，这一全局最优信息未被充分利用。为此，本研究通过将当前全局最优个体的位置信息整合入 MPA 算法中，旨在提升算法的收敛效率。特别地，对高速收敛阶段的位置更新公式进行了改进，表达式更新为：

$$\begin{cases} stepsize_i = R_B \otimes \left(E_i - R_B \otimes x_i \right) \\ x_i = gb + 0.5 \cdot R_v \otimes stepsize_i \end{cases} i = 1,2,\cdots,N; Iter < \frac{1}{3} Max\_Iter \tag{4-2}$$

(2)改进点二：莱维飞行

MPA 算法本身的运动行为就包含莱维飞行，根据 2.1.1 节可知莱维飞行的步长具有各向同性的特点，其引起的剧烈变化有利于算法跳出局部最优。因此，针对 MPA 收敛速度慢、搜索效率低的缺点，采用莱维飞行并将其集成到算法中，其数学表达式如下：

$$x_i = x_i + rand \cdot sign\left[rand - 0.5 \right] \otimes R_L \tag{4-3}$$

式中，$sign\left[rand - 0.5 \right]$有三个值，即 1、0、−1。

4.2.2 基于激活函数的离散化

特征选择问题被界定为一个二元优化问题，意味着每个特征的选择状态只有两种可能性：选中或未选中（即候选解的取值仅限于 0 和 1）。然而，在原始的 MPA 算法中，捕食者个体在搜索空间内的移动是以连续值进行的，这与特征选择场景下的候选特征二元取值要求不符。因此，必须把捕食者个体在搜索空间中的位置表示转化为二进制向量，以适应特征选择问题的需求。在基于元启发式的特征选择算法中，激活函数是一种将连续搜索空间映射到离散搜索空间的常用手段。公式(4-4)为本文选择的 S 形激活函数。

$$T\left(x_{i,j}\left(k \right) \right) = \frac{1}{1 + e^{-x_{i,j}(k)}} \tag{4-4}$$

$$x_{i,j} = \begin{cases} 1, & if\ rand < T\left(x_{i,j}\left(k \right) \right) \\ 0, & 其他 \end{cases} \tag{4-5}$$

算法的每个个体代表一个完整的特征选择方案，其中个体的维度与数据集中的特征数一致，每个维度的值 $x_{i,j}$ 仅能为$\left\{ 0,1 \right\}$。为了确保离散种群能与特征选择问题准确匹配，需要制定一套种群编码规则。按照这个规则，如果 $x_{i,j} = 1$，则意味着第 i 个个体对

第 j 个特征的选择；反之，如果 $x_{i,j}=0$，则表示第 i 个个体未选择第 j 个特征。具体做法是将 $X_{i,j}$ 的值代入 S 形函数中，转换结果表示为公式(4-5)：

4.2.3 BEMPA 的实现步骤

根据上节所述，归纳出 BEMPA 的伪代码如表 4-1 和流程图如图 4-3 所示。

表 4-1 BEMPA 伪代码

算法参数(种群数量、最大迭代次等)设定以及根据式(2-4)进行种群初始化
While 在算法的终止条件未得到满足的情况下
计算适应度值，并构建精英矩阵以及执行海洋记忆存储
If $Iter < Max\_Iter/3$
利用公式(4-2)实现更新捕食者位置
Else If $Max\_Iter/3 \le Iter \le 2\cdot Max\_Iter/3$
For 种群的前一半$\left(i=1,2,\ldots,N/2\right)$
利用公式(2-6)对捕食者位置进行更新
For 种群的后一半$\left(i=N/2+1,\cdots,N\right)$
利用公式(2-7)对捕食者位置进行更新
Else If $Iter > 2\cdot Max\_Iter/3$
利用公式(2-9)实现位置更新
End If
利用公式(4-4)和(4-5)将种群位置进行离散化处理
利用公式（4-3）生成基于莱维飞行的新位置，计算个体适应度值，和原位置进行排序，选择适应度值较小的一半种群位置作为最新的解
执行海洋记忆存储并对精英矩阵进行刷新
利用公式(2-10)模拟 FADs 效应，进一步对捕食者位置进行调整
End While
返回最优个体的位置及其适应度值

4.3 仿真实验及结果分析

4.3.1 0-1 背包问题验证

为了验证上述算法的收敛性能和求解精度，用 0-1 背包问题进行试验。 0-1 背包问题属于典型的整数规划问题，数学模型如下：

$$\max f=\sum_{i=1}^{D}p_i x_i \tag{4-6}$$

$$s.t.\begin{cases}\sum_{i=1}^{D}q_i x_i \le V\\ x_i\in\{0,1\},i=1,2,\cdots,D\end{cases} \tag{4-7}$$

其中，D 问题维度即物品数量，x_i 为旅行者选择第 i 种物品，第 i 种物品的重量为 q_i，价值为 p_i，本身所能承受的总重量不超过 V。

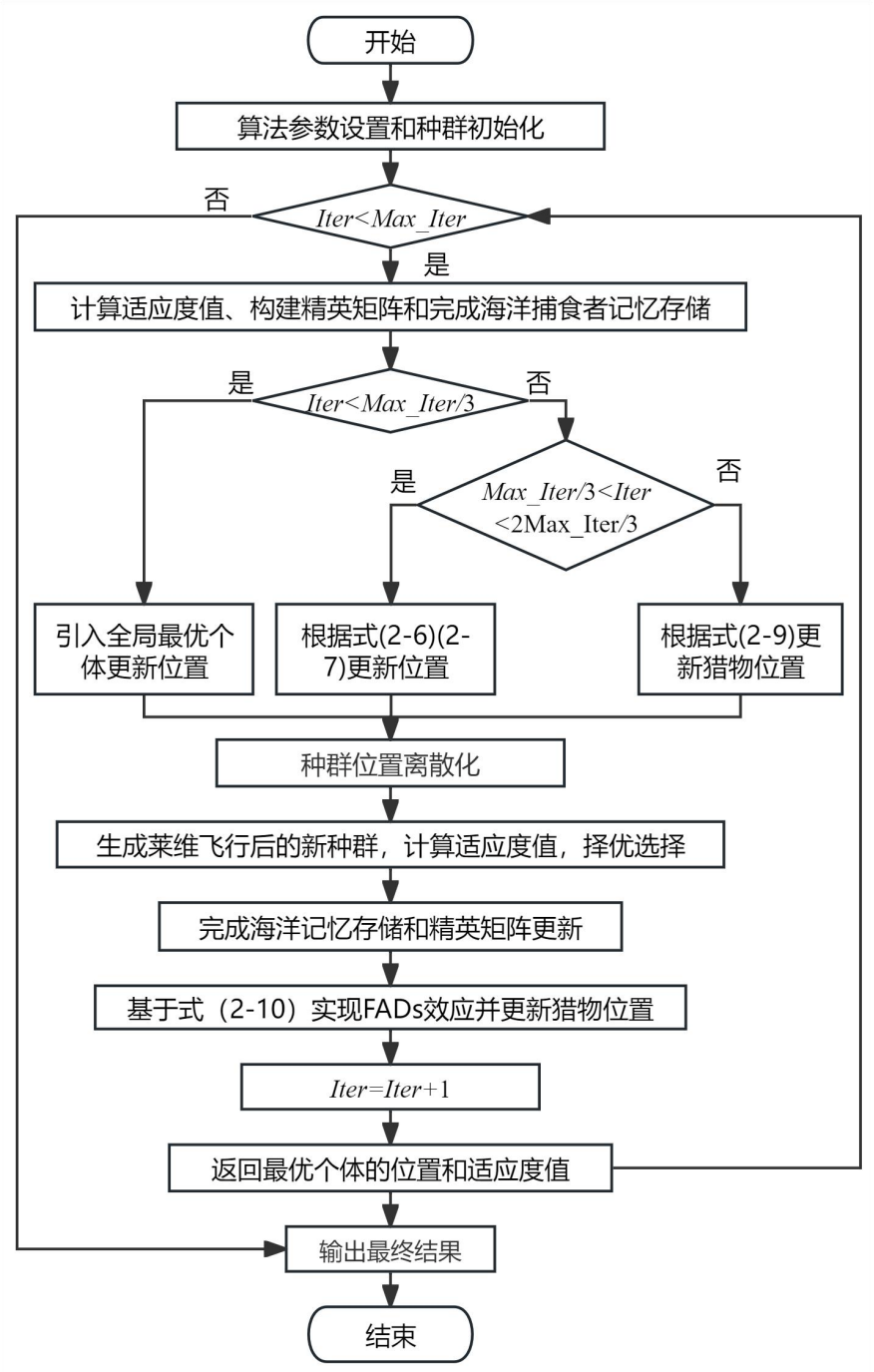


图 4-3 BEMPA 算法流程图

设 $Q=\{q_i\}=\{92,4,43,83,84,68,92,82,6,44,32,18,56,83,25,96,70,48,14,58\}$，$V=878$，$P=\{44,46,90,72,91,40,75,35,8,54,78,40,77,15,61,17,75,29,75,63\}$。这是一个典型的维数 $D=20$ 的 0-1 整数求解问题，通过构造罚函数将上述约束条件转化为无约束的目标

函数，从而形成算法的适应度函数。算法独立运行 20 次。离散海洋捕食者算法记为 BMPA。计算结果如表 4-2 所示：

表 4-2 背包问题优化结果比较

算法	适应度值			
	平均值	标准差	最小值	最大值
BMPA	969.35	20.26	945	1018
BEMPA	995.5	18.95	960	1024

上述问题的标准解为 $f_{\max}=1024$[72]，由表 4-1 可知，对于此类复杂的问题，BMPA 由于自身局部搜索机制限制，收敛精度受到很大影响，性能不好，而 BEMPA 的指标结果均高于 BMPA，说明本文提出的改进方法的有效性。

4.3.2 UCI 数据集验证

本研究从 UCI 机器学习数据库中选取了 13 个不同大小的数据集来评估 BEMPA 的性能。这些数据集的大小和规模各不相同，维度范围从几十到几千不等。根据维度将这些数据集分为三类：小规模（维度小于或等于 100）、中规模（维度大于或等于 100 但小于 1000）和大规模（维度大于或等于 1000）。表 4-3 总结了每个数据集的详细信息。

表 4-3 实验中使用的数据集列表

类型	名称	特征	实例	分类
小型 (数量≤ 100)	Glass	9	214	6
	Heartstalog	13	270	2
	Robotnavigation	24	5456	4
	landsat	36	2000	6
	hillvalley	100	606	2
中型 (100<数量 ≤1000)	urban	147	507	9
	musk1	166	476	2
	arrhythmia	276	450	16
	gastroin	698	152	2
大型 (数量>1000)	qsar	1024	1687	2
	toxicity	1203	171	2
	drivface	6400	606	2
	arcenetrain	10000	100	2

本研究使用了 5 种 MHs 方法进行基于包装方法的特征选择，并与 BEMPA 算法进行了性能比较，包括 SSA[52]、GOA[24]、AOS[17]、蜻蜓算法(Dragonfly algorithm, DA)[66]，将这些算法按照 4.2.2 节的方式变成离散形式。

对于小尺度数据集（特征数≤100），表 4-4~4-6 数据显示，BEMPA 在数据集 Glass、Heartstalog、landsat 和 hillvalley 上选择的特征数量是最小的，在数据集 Robotnavigation 中，BSSA 的表现最佳，但 BEMPA 选择的特征数量平均小于 MPA。在

标准差方面，除了数据集 hillvalley 外，BEMPA 比 BMPA 更稳定，其他数据集相差不大，但稳定性不及其他元启发算法。然而小尺度数据集的选择相差精度大多在 E-01 内波动，综合差距没有实际意义，所以要选择高维数据集进行下一步实验。这意味着当特征数量非常少时，BEMPA 可能会在删除特征的同时消除一些信息特征，从而降低分类精度。由此可见，BEMPA 在小尺度数据集上建立的预测分类模型质量有待提高。

表 4-4 使用 KNN 的元启发算法在特征数量方面的比较

数据	指标	BMPA	BSSA	BGOA	BAOS	BDA	BEMPA
Glass	平均值	3.83	3.5	3.67	**2.4**	3.67	3.07
	标准差	1.21	**0.94**	1.40	1.35	1.24	0.98
Heartstalog	平均值	5.93	4.73	4.7	4.97	4.73	**4.6**
	标准差	1.28	1.14	1.02	2.01	1.20	**0.86**
Robotnavigation	平均值	7.47	**5.07**	5.57	9.03	5.47	5.27
	标准差	1.68	**0.98**	1.19	2.48	1.35	1.26
landsat	平均值	15.43	16.53	15.7	**15**	15.23	15.23
	标准差	2.30	2.46	2.61	3.68	2.36	**2.13**
hillvalley	平均值	37.93	39.97	36.23	40.5	36.5	**34.8**
	标准差	6.099	7.07	6.11	**4.98**	5.19	6.18
urban	平均值	69.87	54.07	56.2	60.87	55.67	**44.93**
	标准差	9.25	7.50	6.33	**6.17**	6.24	12.63
musk1	平均值	83.13	70.3	66.37	68.97	65.23	**62.37**
	标准差	12.23	9.26	8.22	**6.23**	5.16	10.89
arrhythmia	平均值	132.03	107.9	111.33	112.8	109.6	**103.5**
	标准差	19.15	13.46	10.00	9.36	**8.45**	19.71
gastroin	平均值	329.4	263.23	269.87	282.13	269.1	**255.6**
	标准差	30.97	30.60	14.90	**12.72**	14.23	21.76
qsar	平均值	496.5	414.4	406.93	412.53	403.9	**395.7**
	标准差	60.53	43.08	20.40	**14.14**	14.42	18.72
toxicity	平均值	569.9	466.9	466.87	486.13	473.43	**460**
	标准差	47.24	46.55	27.15	17.08	**16.77**	58.81
drivface	平均值	2760	2355.87	2416.93	2590.07	2484.03	**2255.17**
	标准差	318	115.14	152.68	**43.13**	72.49	474.99
arcenetrain	平均值	4351	3743.7	3774.9	4045.9	3918.4	**3707.03**
	标准差	604.1	127.87	190.54	**48.61**	104.84	564.59

对于中等规模的数据集（100<数量≤1000），表 4-4~4-6 数据显示，BEMPA 在 4 个数据集上的特征选择数量均小于其他算法，但在稳定性方面不是最好的，但在数据集 musk1 和 gastroin 上的表现优于 BMPA。由于适应度函数中分类精度占比最多，所以分类精度和适应度函数结果类似。在分类精度方面，BEMPA 的分类精度明显高于 MPA，其中数据集 arrhythmia，BEMPA 在所有算法中结果最好，而 BEMPA 在数据集 urban 和 musk1 的分类精度也排名第二。可以看出，BEMPA 在中型数据集上的表现非常出色。

对于大规模数据集（特征数量>1000），表 4-4~4-6 数据显示，BEMPA 在这 4 个数据集上的性能最好，每个特诊选择都明显优于其他所有算法。特别在数据集 drivface 和 arcenetrain 上，BEMPA 的性能远远高于其他算法。在分类精度方面，BEMPA 在数据集 toxicity、drivface 和 arcenetrain 上表现最佳，且对比 BMPA，其稳定性更佳。

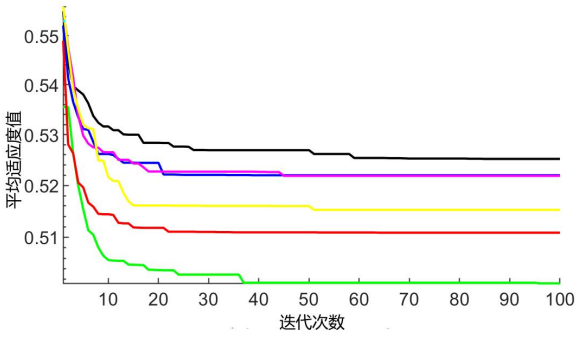
表 4-5 使用 KNN 的元启发算法在分类准确方面的比较

数据	指标	BMPA	BSSA	BGOA	BAOS	BDA	BEMPA
Glass	平均值	47.38	**49.76**	47.70	31.11	48.81	47.62
	标准差	**4.26**	5.13	4.61	7.04	4.83	4.96
Heartstalog	平均值	85.49	87.96	88.27	68.52	**88.46**	86.91
	标准差	4.10	**3.76**	3.93	9.50	3.94	4.32
Robotnavigation	平均值	91.19	**94.26**	93.35	85.89	93.84	93.15
	标准差	0.94	0.68	0.78	3.66	0.69	**0.62**
landsat	平均值	90.88	**91.21**	90.88	87.25	90.74	91.04
	标准差	1.11	1.23	1.16	2.02	**1.10**	1.23
hillvalley	平均值	58.51	**59.12**	57.41	49.26	56.64	58.15
	标准差	3.15	**2.91**	4.01	4.97	4.11	2.99
urban	平均值	58.68	**72.64**	63.76	42.21	65.94	66.37
	标准差	3.43	6.38	**3.25**	5.70	5.24	4.22
musk1	平均值	92.81	**95.51**	93.12	83.44	93.33	93.51
	标准差	1.99	**1.51**	2.21	4.01	1.78	2.18
arrhythmia	平均值	69.67	68.33	69.41	59.81	68.78	**69.74**
	标准差	4.18	5.65	4.75	5.09	**3.80**	4.91
gastroin	平均值	74.67	78.33	**79.56**	58	78.11	77.56
	标准差	5.30	**3.25**	4.61	7.14	4.85	4.71
qsar	平均值	92.48	**93.14**	92.19	90.05	92.66	92.27
	标准差	1.29	1.30	1.55	1.79	1.44	**1.19**
toxicity	平均值	71.76	**75.59**	73.82	56.76	74.80	74.61
	标准差	7.51	5.99	5.75	6.61	5.71	**5.21**
drivface	平均值	96.34	96.25	96.45	95.26	96.25	**96.58**
	标准差	1.72	2.01	1.77	2.00	1.89	**1.37**
arcenetrain	平均值	83.33	84.83	87	70.5	81	**88.17**
	标准差	9.86	7.82	**6.38**	7.81	8.35	7.37

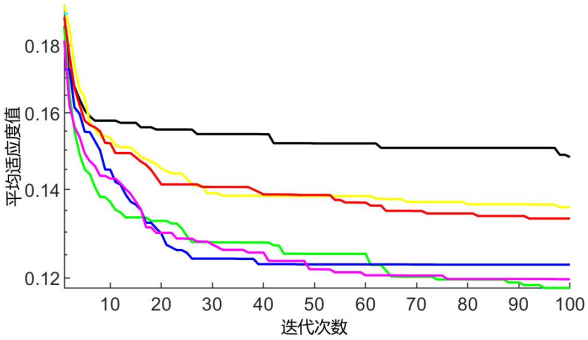
图 4-4 显示了 BEMPA 和其他元启发式算法的收敛曲线。除了数据集 drivface 外，BEMPA 具有出色的收敛速度和精度。显然，在具有大量特征的数据集上，BEMPA 能够显著减少冗余特征，精确选择最重要的特征，实现最高的分类准确度。此外，通过分析表 4-3~4-5 中所有数据的标准差可以发现，BEMPA 在所有数据集上的平均值结果并不理想，这表明 BEMPA 在特征选择方面能够获得比 BMPA 更为稳定、令人满意的解决方案，然而其他元启发式算法的稳定性有待提高。总而言之，BEMPA 在大多数情况下都能保持出色的分类性能，尤其是对于特征数量较多的数据集。BEMPA 能够充分发挥其性能优势，与其他算法相比具有显著优势。

表 4-6 使用 KNN 的元启发算法在适应度函数方面的比较

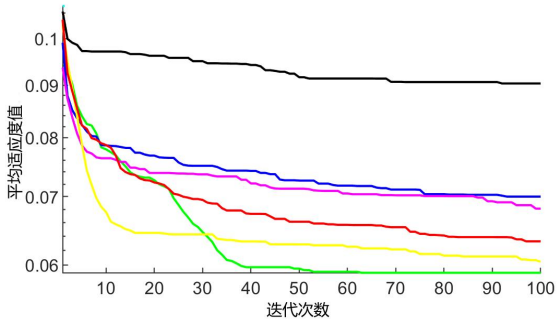
数据	指标	BMPA	BSSA	BGOA	BAOS	BDA	BEMPA
Glass	平均值	0.5252	**0.5012**	0.5219	0.5152	0.5109	0.5220
	标准差	**0.0420**	0.0504	0.0451	0.0430	0.0476	0.0487
Heartstalog	平均值	0.1482	0.1228	**0.1197**	0.1357	0.1179	0.1331
	标准差	0.0407	0.0369	0.0387	0.0328	0.0390	0.0426
Robotnavigation	平均值	0.0904	**0.0589**	0.0681	0.6004	0.0632	0.0700
	标准差	0.0096	0.0067	0.0078	0.0066	0.0068	**0.0062**
landsat	平均值	0.0945	**0.0916**	0.0946	0.0950	0.0959	0.0929
	标准差	0.0109	0.0123	0.0114	**0.0099**	0.0102	0.0122
hillvalley	平均值	0.4145	**0.4087**	0.4253	0.4168	0.4329	0.4178
	标准差	0.0312	**0.0288**	0.0398	0.0401	0.0407	0.0295
urban	平均值	0.4138	0.2745	0.3626	**0.2501**	0.3410	0.3360
	标准差	0.0340	0.0631	**0.0323**	0.0773	0.0594	0.0421
musk1	平均值	0.0762	**0.0487**	0.0721	0.0658	0.0699	0.0680
	标准差	0.0198	**0.0150**	0.0220	0.0208	0.0175	0.0217
arrhythmia	平均值	0.3051	0.3174	0.3069	0.3061	0.3131	**0.3033**
	标准差	0.0414	0.0558	0.0470	0.0439	**0.0376**	0.0485
gastroin	平均值	0.2555	0.2182	0.2063	**0.1945**	0.2206	0.2259
	标准差	0.0524	**0.0321**	0.0456	0.0452	0.0479	0.0465
qsar	平均值	0.0793	**0.0720**	0.0813	0.0786	0.0766	0.0803
	标准差	0.0126	0.0129	0.0154	0.0150	0.0142	**0.0118**
toxicity	平均值	0.2843	**0.2456**	0.2630	0.2487	0.2534	0.2552
	标准差	0.0743	0.0591	0.0569	0.0565	0.0565	**0.0515**
drivface	平均值	0.0406	0.0408	0.0390	**0.0322**	0.0410	0.0373
	标准差	0.0170	0.0199	0.0174	0.0172	0.01872	**0.0138**
arcenetrain	平均值	0.1694	0.1539	0.1325	0.1631	0.1920	**0.1209**
	标准差	0.0974	0.0774	**0.0631**	0.0855	0.0826	0.0726



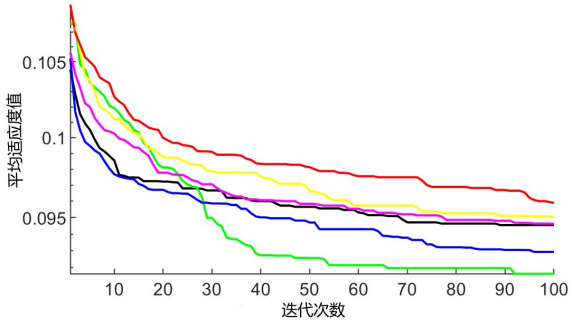
(a)glass



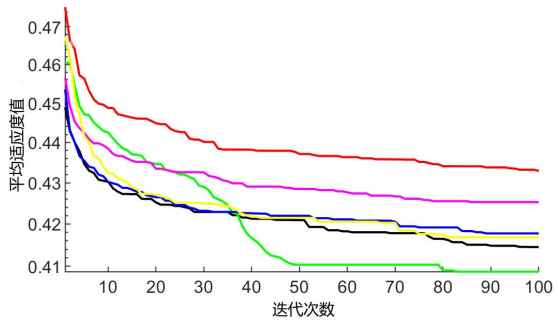
(b)heartstalog



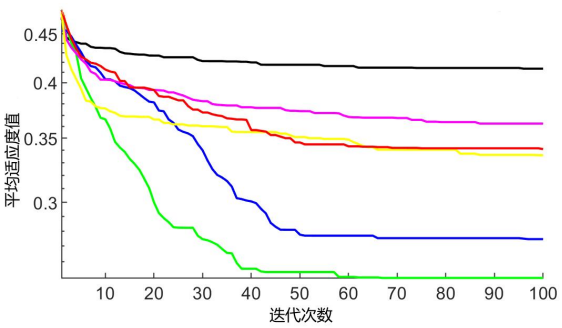
(c)robotnavigation



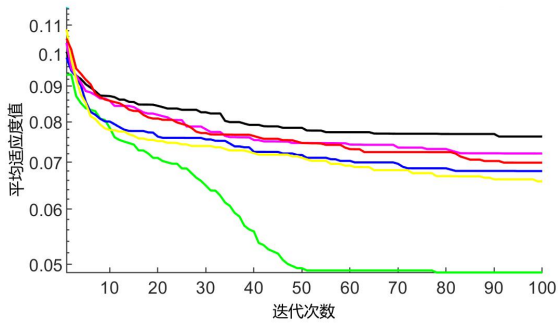
(d)landsat



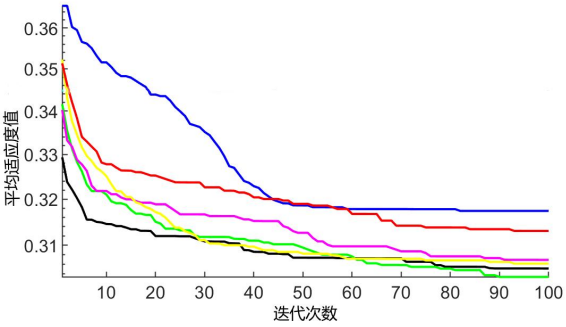
(e)hillvalley



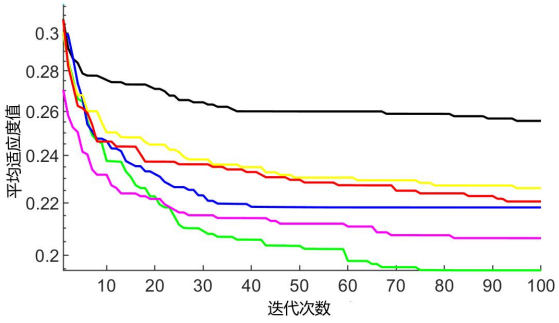
(f)urban



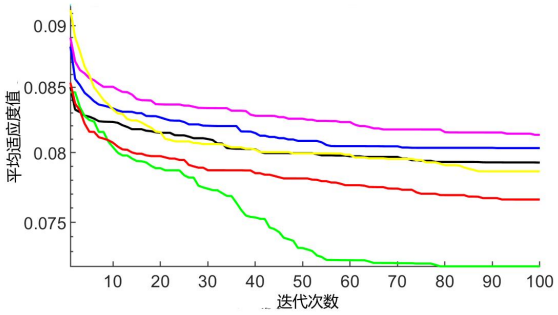
(g)musk1



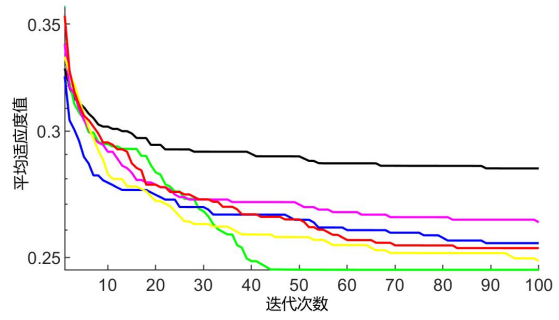
(h)arrhythmia



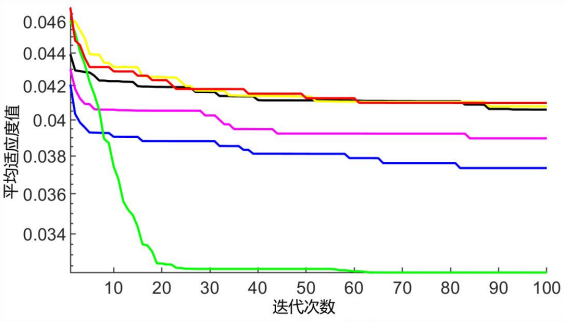
(i)gastroin



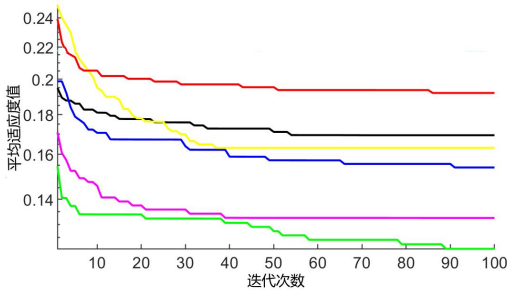
(j)qsar



(k)toxicity



(l)drivface



(m)arcenetrain

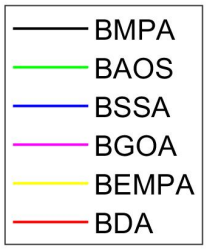


图 4-4 BEMPA 等元启发式算法的收敛曲线

4.4 改进算法在江南大学轴承数据集特征选择中的应用

4.4.1 数据来源

本节使用的是江南大学提供的轴承故障诊断数据集[74]，相应的实验平台和故障类型展示在图 4-5 和图 4-6 中。数据集包括四种不同的状态：正常状态、外圈故障、内圈故障以及滚动体故障。每种故障类型下都包含 3 种工况（转速）：600 rpm、800 rpm 和 1000 rpm。数据采集过程中使用了单个 PCBMA352A60 型号的振动信号加速度传感器，采样频率设置为 50 kHz，每次采样时长为 20 s。本文选择了 600 rmp 下的工况，设每组数据集对原始数据的采样点数均为 2048，每个状态下分别采集 400 个样 本，共计 1600 个样本。

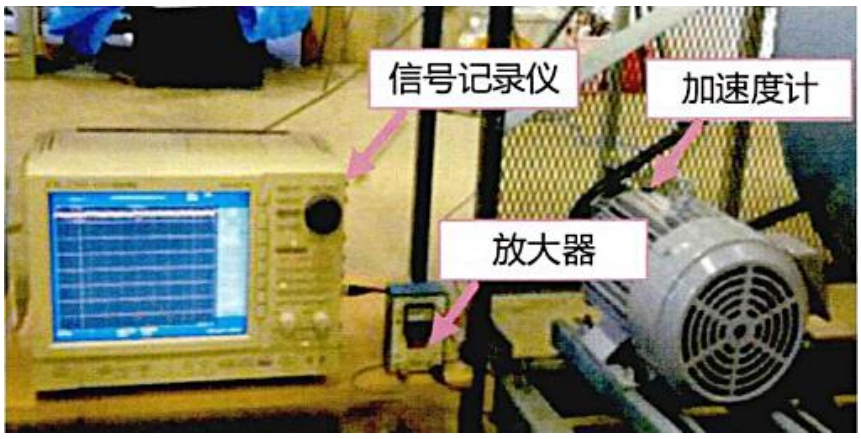
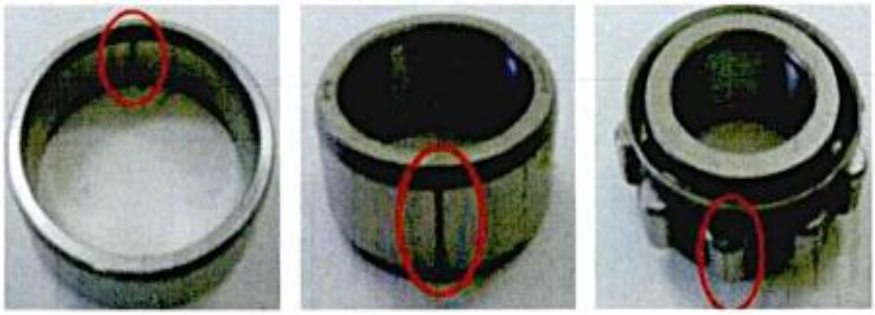


图 4-5 江南大学轴承故障诊断平台



(a)内圈故障　(b)外圈故障　(c)滚动体故障

图 4-6 轴承故障类型

4.4.2 特征集构造

为了提升分类过程的精度和速度，本文综合运用了时域、频域以及时-频域特征，这些特征通过多种信号处理方法提取，并已被证实在精确诊断方面的有效性。基于此，研究采用了一个包含 66 个混合特征的特征集[75]，旨在全面捕捉滚动轴承振动信号的故障特征，以便进行更准确的分类。具体参数见表 4-7。

表 4-7 特征集基本信息

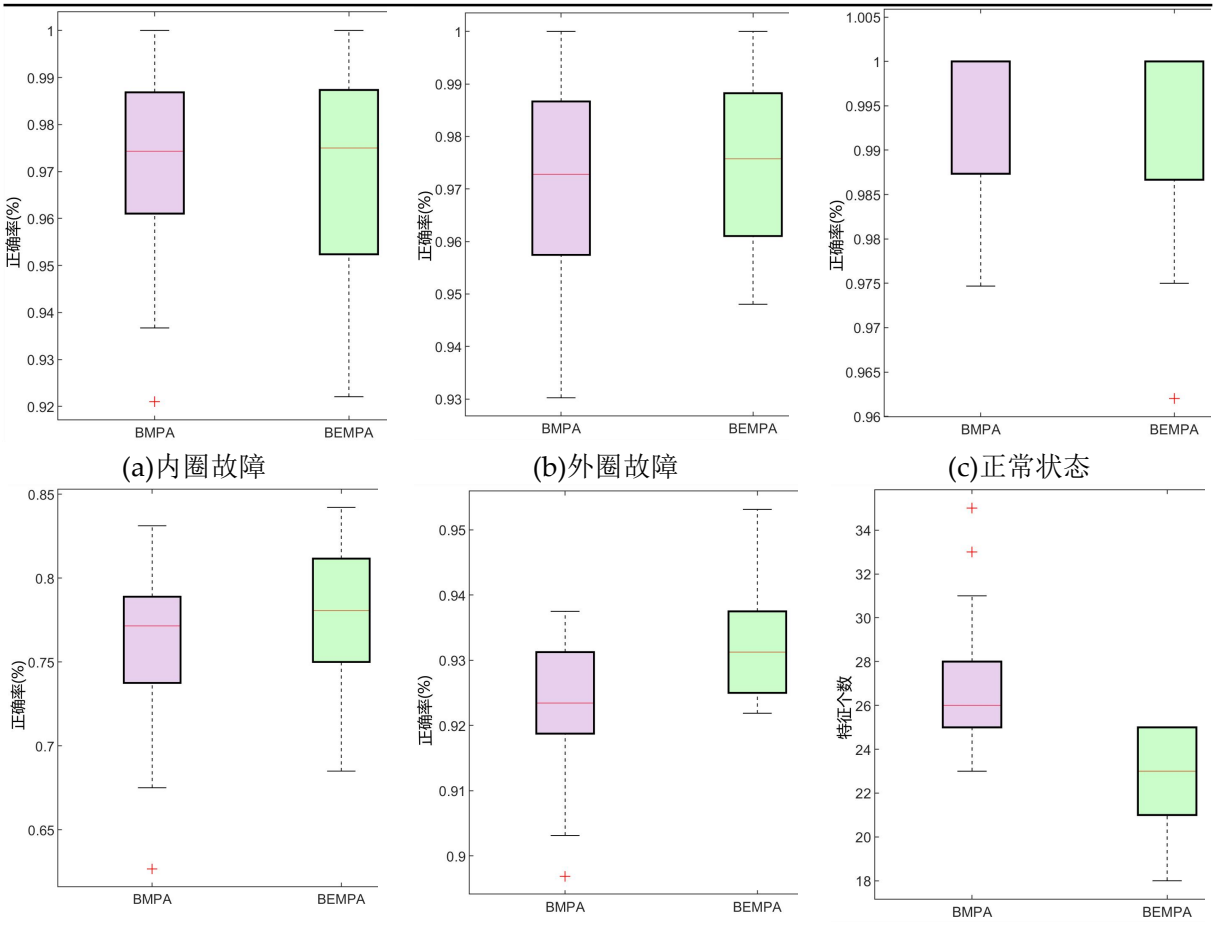
序号	特征名称	基本信息
1-15	时域特征	最大值、最小值、平均值、峰值、整流平均值、方差、标准差、峭度、偏度、均方根、波形因子、峰值因子、脉冲因子、方根幅值、裕度因子
16-19	频域特征	频率均值、频率重心、频率均方根、频率标准差
20-36	三层小波包特征	8 个子带的小波能量比、8 个子带的小波熵、小波奇异谱熵
37-66	奇异值特征	排名前 30 个奇异值特征

4.4.3 实验结果分析

表 4-8 为故障诊断结果数据，图 4-7 为运行结果的箱式图，其中特征维度的最优值是多次迭代时的最佳值，图 4-8 算法最优值时的混淆矩阵分析。

表 4-8 算法识别率(%)与特征子集维度对比

算法		轴承状态				平均值	特征维度
		正常状态	外圈故障	内圈故障	滚子故障		
BMPA	最优值	1	98.73	96.55	77.92	93.44	23
	平均值	99.49	96.93	97.08	75.85	92.28	26.57
BEMPA	最优值	1	1	98.75	82.27	95.32	18
	平均值	99.24	97.57	97.09	78.02	93.20	22.6



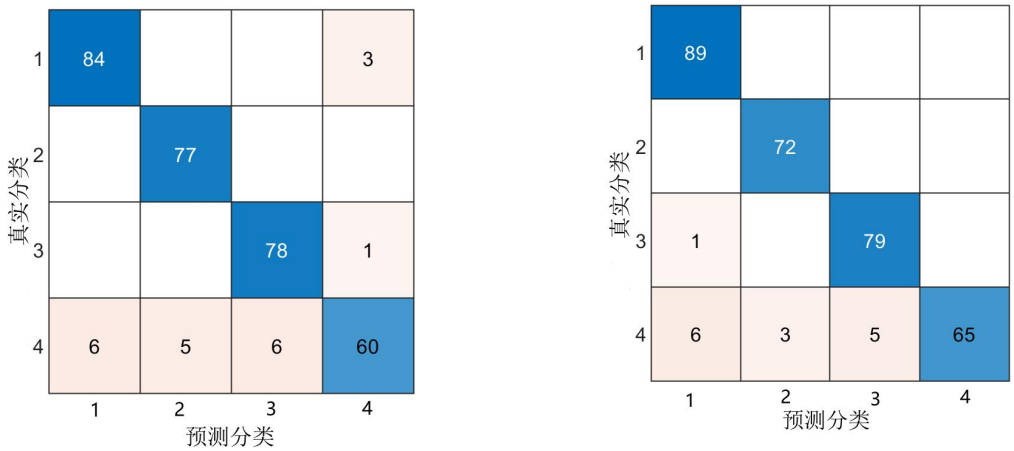
(a)内圈故障 (b)外圈故障 (c)正常状态

(d)滚子故障 (e)综合正确率 (f)特征子集

图 4-7 算法结果箱式图

由图表可知，本章所提出的 BEMPA 算法的平均分类精度为 93.2%，比 BMPA 算

法高 0.92%，在内圈故障和正常状态下，两个算法的识别率几乎一样，但在外圈故障和滚子故障方面，BEMPA 的表现性更好。在保证算法分类精度不下降的前提下，BEMPA 的特征子集数量也比 BMPA 少 5 个，占总特征数量的 7.57%。由此得出本文算法 BEMPA 在轴承故障的特征选择任务中展现出了显著的优越性，特别是在提升识别率和缩减特征维数方面。



(a)BMPA　　(b)BEMPA

图 4-8 算法最优值时的混淆矩阵分析

4.5 本章小结

本章研究了将连续空间转化为离散空间的特征选择方法，并基于改进的海洋捕食者算法提出了离散增强型 BEMPA 算法。BEMPA 融合了莱维飞行和全局最优个体信息，提高了收敛速度和逃避局部最优的能力。针对复杂数据特征选择效率低的问题，提出了基于 BEMPA 的特征选择模型，并在 0-1 背包问题、UCI 数据集和江南大学轴承故障数据集上验证了其有效性。结果显示，BEMPA 在大多数情况下优于其他算法，即使在特征子集较小时也能获得最高分类精度。

第 5 章 总结与展望

5.1 总结

随着数据需求在多个领域的持续增长，精确的数据挖掘方法变得尤为重要，旨在降低时间成本并提高数据分类的精度。数据聚类和特征选择作为数据分析的重要手段，已经广泛应用并受到数据挖掘学者的重视。随着科技的发展，数据变得越来越复杂，处理未分类数据集和优化高维数据集成为研究的热点问题。MPA 作为一种新颖的启发式群智能优化算法，在全局搜索能力方面表现出色，特别是在解决优化数学问题时。因此，本文对 MPA 算法进行了改进，并将其应用于 K-Means 自动聚类和特征选择的研究中，旨在提高传统数据挖掘的精度与性能。本文的主要研究内容包括：

（1）本文全面阐述了 MPA 的基本原理、数学模型和实现流程。针对理论研究的不足，构建了 MPA 的 Markov 链模型并证明了其全局收敛性，为算法有效性提供了理论支撑。通过性能测试，验证了 MPA 在优化问题中的应用潜力和实际性能。此外，对 MPA 的初始参数进行单因素方差分析，揭示了参数设置对求解性能的影响规律，为算法优化提供了重要参考。

（2）提出一种融合多种策略的连续型改进算法 IMPA。通过在初始种群中应用折射反向学习策略，扩大了搜索范围，提高了初始种群质量。利用正余弦算子平衡探索与开发阶段，有效缩减搜索空间。在开发阶段引入柯西高斯变异，加强了局部搜索能力。实验表明，IMPA 在多个测试函数中表现出色，显著提升了收敛精度和寻优速度。在自动聚类实验中，IMPA 优于 MPA 及其他基准算法，如鲸鱼优化、模糊 C-means 和 K-means，显示出更好、更稳定的结果，成功应用于西交大学轴承数据的故障分类问题。

（3）提出一种基于莱维飞行与全局最优个体的二进制增强型算法 BEMPA。该算法利用全局最优实体增强开发能力，并通过莱维飞行算子提高摆脱局部最优性的能力。在 UCI 机器学习数据库的 13 个数据集上测试，BEMPA 在特征选择和分类精度上均优于其他 5 种经典算法。特别是在大规模数据集上，BEMPA 可以获得极小的特征子集，同时保持较高的分类精度。

5.2 展望

本文在探索海洋捕食者算法 MPA 及其改进版本在特征选择和自动聚类问题上的应

用中做出了一系列尝试，但由于研究时间限制和个人学术能力的约束，仍存在一些问题和不足之处，需要在未来的工作中进行更为深入的探讨和完善：

（1）本文对原始 MPA 算法的收敛性进行了基于 Markov 链的初步分析，但这种分析对算法的全面描述并不足够，同时原始算法的收敛性不能直接推断出改进算法的收敛性。因此，在未来的研究中，将对两种改进的 MPA 算法的收敛性进行更为详尽的理论验证和分析，以提供更全面的理解；

（2）虽然基于 MPA 的多种改进策略已在数据挖掘领域显示出良好的应用潜力，但仍有进一步探索的空间。后续研究将考虑将 MPA 与其他高性能的智能算法进行结合，利用各自的优势来提高整体的算法性能。在适当的情况下，也会对这种混合算法的可行性进行深入推敲，并加强其理论基础

（3）在特征选择的内容中，改进的 BEMPA 算法主要针对高维度数据有较好的效果，但不论改进算法还是其他算法的特征选择的稳定性较差，在后期的工作中，可以对如何提高基于元启发算法的特征选择结果的稳定性做出更深入的研究和探讨。

参考文献

[1] 史志龙,卢永强,沈铭清,等.基于数据挖掘的藏药防治新发急性呼吸道传染病组方规律探讨[J].世界科学技术-中医药现代化,2022,24(12):4842-4848.

[2] Chen K L, Abtahi F, Carrero J J, et al. Process mining and data mining applications in the domain of chronic diseases: A systematic review[J]. Artificial Intelligence in Medicine, 2023, 144:102645.

[3] 李泽宇,郝二伟,曹瑞,等.基于数据挖掘和网络药理学分析中药治疗痛风的用药规律及作用机制[J].科学技术与工程,2022,22(36):15967-15979.

[4] Kong D P, Zhao L B, Huang X Y, et al. Self-supervised knowledge mining from unlabeled data for bearing fault diagnosis under limited annotations[J]. Measurement, 2023,220:113387.

[5] 盛少军.基于聚类分析和人工智能的个性化推荐算法研究[J].自动化应用, 2023, 64 (17): 43-45.

[6] Tang C, Tang Y, Zeng Z L, et al. Customer characteristics analysis method based on the selection of electricity consumption characteristics and behavioral portraits of different groups of people[J]. Journal of Intelligent & Fuzzy Systems, 2023,44(3):4273-4283.

[7] 袁欣,俞卫琴,王国强.基于希尔伯特相似度的高维面板数据聚类方法及应用[J].统计与决策,2022,38(17):52-54.

[8] 李永豪.基于多标记学习理论的特征选择方法研究[D].吉林大学,2023.

[9] 刘越.基于对抗训练的特征选择与公平性对抗防御研究[D].南京邮电大学,2023.

[10] Mohamed A W, Sallam K M, Agrawal P, et al. Evaluating the performance of meta-heuristic algorithms on CEC 2021 benchmark problems[J]. Neural Computing & Applications, 2023,35(2):1493-1517.

[11] Faramarzi A, Heidarinejad M, Mirjalili S, et al. Marine predators algorithm: a nature-inspired metaheuristic[J]. Expert Systems with Applications, 2020,152:113377.

[12] 吴涛,商慧丽,张煜葵,等.基于黑洞多目标进化算法的永磁直线同步电机优化设计[J].控制与决策,2022,37(06):1567-1572.

[13] John H H. Adaptation in natural and artificial systems: an introductory analysis with application to biology[M]. Cambridge: MIT press, 1992.

[14] Pillay N. The impact of genetic programming in education[J]. Genetic Programming and

Evolvable Machines, 2020, 21(1-2):87-97.

[15] Rainer S, Kenneth P. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11: 341-359.

[16] 沈焱萍,郑康锋,伍淳华,等.智能启发算法在机器学习中的应用研究综述[J].通信学报,2019,40(12):124-137.

[17] Mahdi A. Atomic orbital search: a novel metaheuristic algorithm[J]. Applied Mathematical Modelling, 2021, 93:657-683.

[18] Seyedali M, Seyed M M, Abdolreza H. Multi-Verse optimizer: a nature-inspired algorithm for global optimization[J]. Neural Computing and Applications, 2016, 27: 495-513.

[19] Beni G， Wang J. Swarm intelligence[C]. NATO Advanced Workshop on Robots and Biological Systems, 1989:425-428.

[20] Kennedy J, Eberhart R. Particle swarm optimization [C]. Proceedings of the 4th IEEE International Conference on Neural Networks，Piscataway: IEEE Service Center, 1995:1942-1948.

[21] Li S M, Chen H L, Wang M J, et al. Slime mould algorithm: a new method for stochastic optimization[J]. Future Generation Computer Systems, 2020,111:300-323.

[22] Seyedali M, Andrew L. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95:51-67.

[23] Mohamed A B, Reda M, Mohammed J, et al. Nutcracker optimizer: a novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems[J]. Knowledge-Based Systems, 2023,262:110248.

[24] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: algorithm and applications[J]. Future Generation Computer Systems, 2019, 97:849-872.

[25] 张磊,刘升,高文欣,等.多子群改进的海洋捕食者算法[J].微电子学与计算机,2022,39(02):51-59.

[26] 付华,刘尚霖,管智峰,等.阶段化改进的海洋捕食者算法及其应用[J].控制与决策,2023,38(04):902-910.

[27] Xing Z K, He Y G. Many-objective multilevel thresholding image segmentation for infrared images of power equipment with boost marine predators algorithm[J]. Applied Soft Computing, 2021,113:107905.

[28] Jia A Q, Liu H Y, Yun Y X, et al. Energy efficiency measures in existing buildings by a multiple-objective optimization with a solar panel system using marine predators optimization algorithm[J]. Solar Energy, 2024,267:112208.

[29] Mohamed A E, Davood M, Diego O, et al. Quantum marine predators algorithm for addressing multilevel image segmentation[J]. Applied Soft Computing, 2021,110:107598.

[30] Sowmya R, Sankaranarayanan V. Optimal vehicle-to-grid and grid-to-vehicle scheduling strategy with uncertainty management using improved marine predator algorithm[J]. Computers and Electrical Engineering, 2022,100:107949.

[31] Ajayi O, Heymann R, Okampo E J. Marine predators algorithm and tunicate swarm algorithm for power system economic load dispatch[C]. The 11th Annual International Conference on Industrial Engineering and Operations Management Singapore, 2021.

[32] 黄志敏,梁承东.基于 K-means 聚类算法的等级测评数据分析[J].电子质量, 2023,(12): 40-44.

[33] 姚雨佳.中国与“一带一路”沿线国家的货币合作研究[D].云南财经大学,2023.

[34] 崔文浩,郑胜,杨森权,等.基于密度峰值聚类的高斯混合模型核电运行工况划分[J].科学技术与工程,2023,23(20):8670-8676.

[35] Abolfazl D T, Mohammad H F Z. Alpha-plane based automatic general type-2 fuzzy clustering based on simulated annealing meta-heuristic algorithm for analyzing gene expression data[J]. Computers in Biology and Medicine,2015,64(1):347-359.

[36] Alokananda D, Siddhartha B, Sandip D, et al. Automatic clustering of colour images using quantum inspired meta-heuristic algorithms[J]. Applied Intelligence, 2023,53:9823-9845.

[37] 李飞,乐强,潘紫微,等.基于自动快速密度峰值聚类的粒子群动态优化算法[J].计算机应用,2023,43(S1):154-162.

[38] Zhang Y, Martínez G M, Kalawsky R S. Grey-box modelling of the swirl characteristics in gas turbine combustion system[J]. Measurement, 2020, 51:1-24.

[39] 黄启萌,吴苗苗,李云.对抗逃避攻击的过滤式对抗特征选择研究[J].电信科学, 2023, 39(07):46-58.

[40] Got A, Moussaoui A, Zouache D. Hybrid filter-wrapper feature selection using whale optimization algorithm: a multi-objective approach[J]. Expert Systems with Applications,

2021,183:115312.

[41] Naik A K, Kuppili V. An embedded feature selection method based on generalized classifier neural network for cancer classification[J]. Computers in Biology and Medicine, 2023,168:107677.

[42] Somnath C, Shreya B, Arindam M, et al. Breast cancer detection from thermal images using a grunwald-letnikov-aided dragonfly algorithm-based deep feature selection method[J]. Computers in Biology and Medicine, 2022,141:105027.

[43] Zhang C, Zhou J Z, Li C S. A compound structure of ELM based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting[J]. Energy Conversion and Management, 2017, 143:360-376.

[44] Arpan B, Khalid H S, Erik C. COVID-19 detection from CT scans using a two-stage framework[J]. Expert Systems with Applications, 2022,193:116377.

[45] Kaur T, Saini B S, Gupta S. A novel feature selection method for brain tumor MR image classification based on the Fisher criterion and parameter-free bat optimization[J]. Neural Computing and Applications, 2018, 29(8): 193-206.

[46] Zhao Y X, Dong J W, Chen H, et al. A binary dandelion algorithm using seeding and chaos population strategies for feature selection[J]. Applied Soft Computing, 2022,125:109166.

[47] 拓云天,崔洁,王津沓,等.基于数字孪生的滚动轴承健康状态预测[J].制造技术与机床,2022(11):156-162.

[48] 丁瑞成,周玉成.引入莱维飞行与动态权重的改进灰狼算法[J].计算机工程与应用,2022,58(23):74-82.

[49] 杨顺令.基于冗余策略的可靠性优化问题[D].南昌大学,2023.

[50] Xue J K, Shen B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization[J]. Journal of Supercomputing, 2023, 79(7): 7305-7336.

[51] Seyedali M, Seyed M M, Andrew L. Grey wolf optimizer[J]. Advances in Engineering Software, 2014,69:46-61.

[52] Xue J K, Shen B. A novel swarm intelligence optimization approach： sparrow search algorithm[J]. Systems Science & Control Engineering，2020，8(1): 22-34.

[53] 赵光权. 基于贪婪策略的微分进化算法及其应用研究[D]. 哈尔滨工业大学，2007.

[54] Tizhoosh H R. Opposition-based learning: a new scheme for machine intelligence[C]. International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), 2005:695–701.

[55] Shao P, Wu Z, Zhou X, et al, FIR digital filter design using improved particle swarm optimization based on refraction principle[J]. Soft Computing, 2017,21: 2631–2642.

[56] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-based systems, 2016, 96:120–133.

[57] 刘勇,马良.转换参数非线性递减的正弦余弦算法[J].计算机工程与应用, 2017, 53(02):1-5+46.

[58] Chen X, Shen A. Self-adaptive differential evolution with Gaussian–Cauchy mutation for large-scale CHP economic dispatch problem[J]. Neural Computing & Applications,2022, 34:11769–11787.

[59] Zhao S, Wu Y, Tan S, et al. QQLMPA: A quasi-opposition learning and Q-learning based marine predators algorithm[J]. Expert Systems with Applications, 2023, 213:119246.

[60] Sadiq A S, Dehkordi A A, Mirjalili S, et al. Nonlinear marine predator algorithm: a cost-effective optimizer for fair power allocation in NOMA-VLC-B5G networks[J]. Expert Systems with Applications, 2022, 203:119246.

[61] Trojovský P, Dehghani M. Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications[J]. Sensors, 2022, 22(3):855.

[62] Ouyang Y Y, Liu J M, Tong T J, et al. A rank-based high-dimensional test for equality of mean vectors[J]. Computational Statistics & Data Analysis, 2022,173:107495.

[63] Davies D L, Bouldin D W. A cluster separation measure[J]. IEEE transactions on pattern analysis and machine intelligence, 1979, 1(2).

[64] Dunn J C. Well-separated clusters and optimal fuzzy partitions[J]. Cybern, 1974, 4:95-104.

[65] Pakhira M K, Bandyopadhyay S, Maulik U. Validity index for crisp and fuzzy clusters[J]. Pattern Recognition, 2004, 37:487-501.

[66] Zhou K. Enhanced feature extraction for machinery condition monitoring using recurrence plot and quantification measure[J]. International journal of advanced

manufacturing technology, 2022, 123(9-10):3421-3436.

[67] Chou C, Su M, Lai E. A new cluster validity measure and its application to image compression[J]. Pattern analysis and applications, 2021, 24(4):1489-1500.

[68] Wang H, Wang J, Wang G. A survey of fuzzy clustering validity evaluation methods[J]. Information Sciences, 2022, 618:270–297.

[69] 雷亚国,韩天宇,王彪,等.XJTU-SY 滚动轴承加速寿命试验数据集解读[J].机械工程学报,2019,55(16):1-6.

[70] Khurma R A, Aljarah I, Sharieh A, et al. A review of the modification strategies of the nature inspired algorithms for feature selection problem[J]. Mathematics, 2022, 10:1-45.

[71] Ge Y, Zhai J F, Su1 P C. Traffic flow prediction based on multi-spatiotemporal attention gate graph convolution network[J]. Journal of Advanced Transportation, 2022:1-9.

[72] 雷秀娟. 群智能优化算法及其应用[M]. 北京: 科学出版社, 2012.

[73] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems[J]. Neural Computing & Applications, 2016, .27:1053-1073.

[74] 江南大学数据集: http://www.52phm.cn/datasets/bear/Bearing-data-set-of-Jiangnan-University.html

[75] 张宇.蚁狮算法的改进及在机械故障诊断中的应用[D].安徽工程大学,2020.

攻读学位期间发表的学术成果目录

一、学术论文

[1]（导师一作，本人二作） A modified shuffled frog leaping algorithm with inertia weight[J]. Scientific Reports, 2024,14:1-14.（本篇论文对应文中第 2 章）；

[2] （导师一作，本人二作）Adaptive clustering algorithm based on improved marine predation algorithm and its application in bearing fault diagnosis[J]. Electronic Research Archive, 2023, 31(11):7078-7103.（本篇论文对应文中第 3）；

[3]本人一作，海洋捕食者算法的改进及其应用[J].佳木斯大学学报(自然科学版),2024,42(01):26-29+100.（本篇论文对应文中第 3 章）；

[4] （导师一作，本人二作）Binary enhanced Marine predator algorithm for feature selection[J].（已投 Neural Computing and Applications）（本篇论文对应文中第 4 章）

二、参与科研项目

[1]2022 年度安徽省教育厅科学研究重点项目“RV 减速器隐性故障诊断应用性能提升技术研究”（2022AH050995）

致谢

在我的论文完成之际，我深感感激与欣喜交织。首先，我要向我的导师表达最诚挚的感谢。在您的悉心指导下，我得以深入理解课题，掌握研究方法，克服研究中的种种困难。您的严谨治学、勤奋工作、无私奉献的精神深深影响了我，使我受益终身。

同时，我要感谢我的家人，是你们的支持与理解让我能够专注于学业，无惧前行。你们的爱是我前进的动力，也是我面对困难时的坚强后盾。感谢你们一直以来对我的包容与鼓励，让我在成长的道路上更加坚定。

此外，我还要感谢我 808 教室的所有同门和同学们。我们共同度过的时光充满了欢笑与汗水，是你们陪伴我度过了这段难忘的学习生涯。在论文撰写过程中，我们相互帮助、共同进步，这种真挚的友谊将永远铭刻在我的心中。

最后，我要感谢学校为我提供的良好学习环境和丰富资源，让我能够充分利用这些资源，完成这篇论文。同时，也要感谢所有为我提供过帮助和支持的老师、同学和朋友们，是你们的关心与鼓励让我更加坚定地走向未来。

在此，我再次向所有关心、支持、帮助过我的人表示衷心的感谢！在未来的道路上，我将继续努力，不负韶华，不负众望，为社会做出更大的贡献。

尚德教学　唯实惟新