

分类号: TP18

密 级: 公开

学校代码: 10363

学 号: 2220320162



安徽工程大学

Anhui Polytechnic University

# 硕士学位论文

题 目     具有合作竞争机制的群智能  
            优化算法研究及其应用

论 文 作 者     汪启梁

校 内 导 师     张平改、魏利胜

校 外 导 师     刘小琿

专业学位类别     电子信息

研究方向(领域)     控制工程与自动化技术

论文提交日期:     2025年 05月16日

分类号: TP18

单位代码: 10363

密 级: 公开

学 号: 2220320162

题 目 具有合作竞争机制的群智能优化算法研究及其应用

题 目 Research on Swarm Intelligence Optimization Algorithm with  
Cooperative-Competitive and Its Applications

学 生 姓 名: 汪启梁

校 内 导 师: 张平改、魏利胜

校 外 导 师: 刘小瑋

专业学位类别 : 电子信息

研究方向(领域): 控制工程与自动化技术

论文答辩日期: 2025 年 5 月 9 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： 汪应梁

日期： 2025 年 05 月 09 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。

保密 ☐，在 \_\_\_\_\_ 年解密后适用本版权书。

本学位论文属于

不保密 ☒。

学位论文作者签名： 汪启梁

指导教师签名： 张平改

日期： 2025 年 5 月 9 日

日期： 2025 年 5 月 9 日

## 具有合作竞争机制的群智能优化算法研究及其应用

### 摘 要

随着工业系统复杂度的不断提高,传统优化算法在实际应用中面临诸多挑战,如收敛速度过快、易陷入局部最优、以及高维复杂问题难以实现全局优化等问题。因此,亟需发展全局搜索能力更强、稳定性更高的智能优化算法,以应对日益复杂的工程优化问题。为此,本文将合作竞争机制融合到算法中,研究了具有合作竞争机制的粒子群优化算法和具有合作竞争机制的人类学习优化算法,有效地增强了算法的全局优化性能,并将所提出的算法应用于求解复杂工程优化问题,以验证其优化性能和求解能力。主要研究内容如下:

(1) 针对基本粒子群优化算法在高维复杂问题中探索能力不足、易陷入局部最优的问题,研究了一种具有合作竞争机制的粒子群优化算法(Particle Swarm Optimization with Cooperative and Competitive, PSOCC),并通过数值实验分析了 PSOCC 算法在搜索最优解过程中的动态收敛特性。在此基础上,进一步考虑搜索过程中种群对信息的利用效率差异特征,设计了一种信息交互式合作竞争策略,研究了一种信息交互式的合作竞争粒子群优化算法(Interactive Particle Swarm Optimization with Cooperative and Competitive, IPSOCC),并通过数值实验深入地分析了不同参数对算法性能的影响,从而有效地增强了算法的优化性能。

(2) 受人类的合作竞争学习机制启发,设计了一种合作竞争学习算子,研究了具有合作竞争机制的人类学习优化算法(Cooperative-Competitive Human Learning Optimization, CCHLO),并通过数值实验分析了所设计合作竞争学习算子的作用与功能,增强了算法的开采能力。同时, CCHLO 算法中引入重新学习机制,确保了种群的多样

性，提升了算法的探索能力。在此基础上，进一步考虑合作竞争学习算子与社会学习算子对种群信息的利用效率差异，设计了一种增强探索与开采平衡的合作竞争模型，研究了一种增强探索与开采平衡的合作竞争人类学习优化算法（Enhanced Cooperative-Competitive Human Learning Optimization, ECCHLO），并通过数值实验分析了增强探索与开采平衡的合作竞争模型可有效地帮助算法在探索与开采之间实现理想的平衡，从而进一步增强算法的优化性能。

（3）将具有合作竞争机制的粒子群优化算法和具有合作竞争机制的人类学习优化算法应用于复杂工程优化问题中，如拉压弦设计、三杆桁架设计、悬梁架设计、0-1 背包、步进圆锥滑轮、行星轮系设计优化等问题，为复杂工程优化设计问题提供了新的解决方案。同时，以炉膛燃烧场火焰识别问题为典型应用案例，研究了基于 HSV 颜色模型的炉膛火焰识别方法和基于新型神经网络的炉膛火焰识别方法，验证了所提出 IPSOCC 算法与 ECCHLO 算法的求解能力，进而显著提升炉膛火焰识别的效果与精确度，进一步促进了算法的应用与发展。

**关键词：**智能优化算法；合作竞争机制；粒子群优化算法；人类学习优化；复杂工程优化问题

# Research on Swarm Intelligence Optimization Algorithm with Cooperative-Competitive and Its Applications

## ABSTRACT

With the continuous increase in the complexity of industrial systems, traditional optimization algorithms face numerous challenges in practical applications, such as excessively fast convergence, susceptibility to local optima, and difficulties in achieving global optimization for high-dimensional complex problems. Therefore, it is imperative to develop intelligent optimization algorithms with stronger global search capabilities and higher stability to address increasingly complex engineering optimization problems. To this end, this paper integrates the cooperative-competitive mechanism into optimization algorithms and investigates the Particle Swarm Optimization algorithm with Cooperative and Competitive mechanisms (PSOCC) and the Human Learning Optimization algorithm with Cooperative and Competitive mechanisms (CCHLO), effectively enhancing the global optimization performance of the algorithms. The proposed algorithms are applied to complex engineering optimization problems to verify their optimization performance and solving ability. The main research contents are as follows:

1) To address the issues of insufficient exploration ability and susceptibility to local optima in high-dimensional complex problems of the basic particle swarm optimization algorithm, this paper studies a Particle Swarm Optimization algorithm with Cooperative and Competitive mechanisms (PSOCC), and analyzes the dynamic convergence characteristics of the PSOCC algorithm in the process of searching for optimal solutions through numerical experiments. On this basis, considering the differences in the efficiency of population information utilization during the search process, an interactive information exchange cooperative-competitive strategy is designed, and an Interactive Particle

Swarm Optimization algorithm with Cooperative and Competitive mechanisms (IPSOCC) is studied. The effects of different parameters on the performance of the algorithm are thoroughly analyzed through numerical experiments, effectively enhancing the optimization performance of the algorithm.

2) Inspired by the human cooperative-competitive learning mechanism, a cooperative-competitive learning operator is designed, and a Cooperative-Competitive Human Learning Optimization algorithm (CCHLO) is studied. Through numerical experiments, the role and function of the designed cooperative-competitive learning operator are analyzed, effectively enhancing the exploitation ability of the algorithm. Meanwhile, a relearning mechanism is introduced in the CCHLO algorithm to ensure population diversity and improve the exploration ability of the algorithm. On this basis, considering the differences in the efficiency of population information utilization between the cooperative-competitive learning operator and the social learning operator, an enhanced cooperative-competitive model for balancing exploration and exploitation is designed, and an Enhanced Cooperative-Competitive Human Learning Optimization algorithm (ECCHLO) is studied. Numerical experiments show that the enhanced cooperative-competitive model can effectively help the algorithm achieve an ideal balance between exploration and exploitation, thereby further enhancing the optimization performance of the algorithm.

3) Particle Swarm Optimization algorithm with a cooperation-competition mechanism and the Human Learning Optimization algorithm with a cooperation-competition mechanism have been applied to complex engineering optimization problems such as tension-compression string design, three-bar truss design, cantilever frame design, 0-1 knapsack problem, stepped cone pulley, and planetary gear system design optimization. These applications provide new solutions for complex

engineering design problems. Additionally, using the furnace combustion field flame recognition problem as a typical case, methods based on the HSV color model and a novel neural network for furnace flame recognition have been studied. The proposed IPSOCC and ECCHLO algorithms' solving capabilities were verified, significantly improving the flame recognition effect and accuracy, and further promoting the application and development of these algorithms.

**Key words:** Intelligent optimization algorithm; Cooperative-competition mechanism; Particle swarm optimization; Human learning optimization; Complex engineering optimization problems

## 目 录

|                                |    |
|--------------------------------|----|
| 摘 要 .....                      | IV |
| ABSTRACT.....                  | VI |
| 第 1 章 绪论 .....                 | 1  |
| 1.1 选题背景与意义.....               | 1  |
| 1.2 国内外研究现状.....               | 2  |
| 1.2.1 粒子群优化算法.....             | 2  |
| 1.2.2 人类学习优化算法.....            | 4  |
| 1.2.3 复杂工程优化问题应用.....          | 6  |
| 1.3 主要研究内容.....                | 7  |
| 第 2 章 具有合作竞争机制的粒子群优化算法研究 ..... | 10 |
| 2.1 基本粒子群优化算法.....             | 10 |
| 2.2 具有合作竞争机制的粒子群优化算法.....      | 11 |
| 2.2.1 初始化.....                 | 12 |
| 2.2.2 粒子协同共享搜索策略.....          | 12 |
| 2.2.3 粒子竞争部署策略.....            | 14 |
| 2.2.4 粒子停滞处理策略.....            | 15 |
| 2.2.5 PSOCC 算法实现.....          | 15 |
| 2.2.6 PSOCC 算法参数分析 .....       | 16 |
| 2.2.7 PSOCC 算法收敛性分析 .....      | 19 |
| 2.3 信息交互式合作竞争粒子群优化算法.....      | 20 |
| 2.3.1 信息交互式合作竞争策略.....         | 20 |
| 2.3.2 IPSOCC 算法实现.....         | 21 |
| 2.3.3 IPSOCC 算法参数分析.....       | 22 |
| 2.4 仿真实验与结果分析.....             | 24 |
| 2.4.1 低维函数测试结果.....            | 26 |
| 2.4.2 高维函数测试结果.....            | 28 |
| 2.4.3 收敛曲线对比结果.....            | 30 |
| 2.5 本章小结.....                  | 38 |

|                                  |    |
|----------------------------------|----|
| 第 3 章 具有合作竞争机制的人类学习优化算法研究 .....  | 39 |
| 3.1 人类学习优化算法.....                | 39 |
| 3.1.1 随机学习算子.....                | 39 |
| 3.1.2 个体学习算子.....                | 39 |
| 3.1.3 社会学习算子.....                | 40 |
| 3.2 具有合作竞争机制的人类学习优化算法.....       | 41 |
| 3.2.1 合作竞争学习算子.....              | 41 |
| 3.2.2 重新学习策略.....                | 42 |
| 3.2.3 CCHLO 算法实现 .....           | 43 |
| 3.2.4 CCHLO 算法参数分析 .....         | 44 |
| 3.2.5 合作竞争学习算子功能分析.....          | 48 |
| 3.3 增强探索与开采平衡的合作竞争人类学习优化算法.....  | 50 |
| 3.3.1 增强探索与开采平衡的合作竞争模型.....      | 50 |
| 3.3.2 ECCHLO 算法实现.....           | 51 |
| 3.3.3 ECCHLO 算法参数分析.....         | 52 |
| 3.4 仿真实验与结果分析.....               | 54 |
| 3.4.1 低维函数测试结果.....              | 55 |
| 3.4.2 高维函数测试结果.....              | 56 |
| 3.5 本章小结.....                    | 58 |
| 第 4 章 具有合作竞争机制的群智能优化算法应用研究 ..... | 59 |
| 4.1 连续编码复杂工程优化问题应用研究.....        | 59 |
| 4.1.1 拉压弦设计问题.....               | 59 |
| 4.1.2 三杆桁架设计问题.....              | 61 |
| 4.1.3 悬梁架设计问题.....               | 62 |
| 4.2 二进制编码复杂工程优化问题应用研究.....       | 63 |
| 4.2.1 0-1 背包问题 .....             | 63 |
| 4.2.2 步进圆锥滑轮问题.....              | 67 |
| 4.2.3 行星轮系设计优化问题.....            | 69 |
| 4.3 基于 HSV 颜色模型的炉膛火焰识别方法研究.....  | 71 |
| 4.3.1 HSV 颜色模型原理.....            | 71 |

|                              |    |
|------------------------------|----|
| 4.3.2 HSV 颜色模型构建.....        | 72 |
| 4.3.3 HSV 颜色模型的目标函数.....     | 74 |
| 4.3.4 HSV 颜色模型的实现流程.....     | 75 |
| 4.3.5 仿真实验与结果分析.....         | 76 |
| 4.4 基于新型神经网络的炉膛火焰识别方法研究..... | 78 |
| 4.4.1 新型神经网络模型构建.....        | 79 |
| 4.4.2 新型神经网络模型的目标函数.....     | 81 |
| 4.4.3 新型神经网络模型的实现流程.....     | 82 |
| 4.4.4 仿真实验与结果分析.....         | 83 |
| 4.5 本章小结.....                | 84 |
| 第 5 章 总结与展望 .....            | 86 |
| 5.1 总结.....                  | 86 |
| 5.2 展望.....                  | 87 |
| 参考文献 .....                   | 88 |
| 攻读学位期间发表的学术成果 .....          | 93 |
| 攻读学位期间获得的奖励 .....            | 93 |
| 致 谢 .....                    | 94 |

## 插图清单

|                                                |    |
|------------------------------------------------|----|
| 图 1-1 研究内容框架图 .....                            | 8  |
| 图 2-1 粒子协同共享搜索策略示意图 .....                      | 13 |
| 图 2-2 PSOCC 算法流程图.....                         | 16 |
| 图 2-3 PSOCC 与 PSO 在 F1 函数中的 AD 曲线与适应度曲线 .....  | 19 |
| 图 2-4 PSOCC 与 PSO 在 F10 函数中的 AD 曲线与适应度曲线 ..... | 20 |
| 图 2-5 IPSOCC 流程图 .....                         | 21 |
| 图 2-6 30 维度下单峰函数收敛曲线 .....                     | 32 |
| 图 2-7 30 维度下多峰函数收敛曲线 .....                     | 35 |
| 图 2-8 固定维度下复合函数收敛曲线 .....                      | 38 |
| 图 3-1 CCHLO 流程图.....                           | 44 |
| 图 3-2 30 维 F6 的平均距离曲线与适应度曲线 .....              | 49 |
| 图 3-3 30 维 F12 的平均距离曲线与适应度曲线 .....             | 49 |
| 图 3-4 50 维 F6 的平均距离曲线与适应度曲线 .....              | 49 |
| 图 3-5 50 维 F12 的平均距离曲线与适应度曲线 .....             | 50 |
| 图 3-6 增强探索与开采平衡合作竞争模型的 $pic$ 变化曲线 .....        | 51 |
| 图 3-7 ECCHLO 流程图 .....                         | 52 |
| 图 4-1 燃烧火焰 HSV 分量的三维信息分布效果图 .....              | 72 |
| 图 4-2 样本取样区域 .....                             | 75 |
| 图 4-3 火焰识别流程图 .....                            | 76 |
| 图 4-4 炉膛火焰识别实验结果 .....                         | 77 |
| 图 4-5 新型神经网络模型 .....                           | 79 |
| 图 4-6 ECCHLO 算法优化新型神经网络用于火焰识别流程图 .....         | 82 |
| 图 4-7 ECCHLO 优化新型神经网络用于火焰识别实验结果 .....          | 83 |

## 插 表 清 单

|                                             |    |
|---------------------------------------------|----|
| 表 2-1 PSOCC 分析参数 $L$ 和 $K$ 的取值情况 .....      | 17 |
| 表 2-2 参数正交表 .....                           | 18 |
| 表 2-3 IPSOCC 分析参数 $s_i$ 和 $s_j$ 的取值情况 ..... | 22 |
| 表 2-4 IPSOCC 参数正交表 .....                    | 23 |
| 表 2-5 CEC2005 测试函数参数信息 .....                | 24 |
| 表 2-6 各算法参数信息 .....                         | 26 |
| 表 2-7 30 维空间维度中各算法收敛精度统计 .....              | 27 |
| 表 2-8 50 维空间维度中各算法收敛精度统计 .....              | 29 |
| 表 3-1 CCHLO 参数取值情况 .....                    | 45 |
| 表 3-2 $G=4$ 各维度参数正交表 .....                  | 45 |
| 表 3-3 $G=5$ 各维度参数正交表 .....                  | 46 |
| 表 3-4 $G=6$ 各维度参数正交表 .....                  | 46 |
| 表 3-5 ECCHLO 参数取值情况 .....                   | 53 |
| 表 3-6 各维度参数正交表 .....                        | 53 |
| 表 3-7 各算法参数设定值 .....                        | 54 |
| 表 3-8 30 维空间维度中各算法收敛精度统计 .....              | 55 |
| 表 3-9 100 维空间维度中各算法收敛精度统计 .....             | 57 |
| 表 4-1 各算法于拉压弦设计问题的结果比较 .....                | 60 |
| 表 4-2 各算法于三杆桁架设计问题的结果比较 .....               | 62 |
| 表 4-3 各算法于悬梁架设计问题的结果比较 .....                | 63 |
| 表 4-4 各算法于 0-1 背包问题的低维度结果比较 .....           | 65 |
| 表 4-5 各算法于 0-1 背包问题的高维度结果比较 .....           | 66 |
| 表 4-6 各算法于 0-1 背包问题排名结果 .....               | 67 |
| 表 4-7 各算法于步进圆锥滑轮问题的结果比较 .....               | 69 |
| 表 4-8 各算法于行星轮系设计优化问题的结果比较 .....             | 70 |
| 表 4-9 炉膛火焰识别实验数据 .....                      | 78 |
| 表 4-10 ECCHLO 优化新型神经网络用于火焰识别实验数据 .....      | 84 |

# 第 1 章 绪论

## 1.1 选题背景与意义

随着科学技术的快速发展，现代工程领域面临的优化问题愈加复杂。尤其是在处理大规模变量、复杂目标函数和多重约束条件的情况下，传统优化算法的局限性日益显现。经典的精确模型优化方法往往无法有效应对这些不断演化和复杂化的实际问题，导致其在求解效率和结果质量上的表现受到显著制约。具体而言，随着问题规模的增大，传统算法在处理高维、非线性、强耦合以及动态变化的优化任务时，计算负担和求解时间的增长使得其适用性和鲁棒性受到极大挑战<sup>[1]</sup>。因此，研究和开发更高效、更灵活的优化算法，特别是提升算法全局搜索能力和局部精度的创新策略，已成为优化领域中的关键研究方向。在此背景下，算法的创新与发展不仅能够提升工程领域问题求解的效率和质量，还能为智能化技术的推进提供新的动力<sup>[2]</sup>。这些技术创新具有深远的理论意义，能够推动工业、能源、环境保护等多个领域的优化和发展，具有广泛的应用前景和实际价值。

随着科技的飞速发展，在优化问题中算法在处理二进制与十进制问题时遇到的挑战和难点也日益棘手，如何在快速发展的时代背景中研究合适的解决工具，成为了优化算法研究的关键方向。粒子群优化算法（Particle Swarm Optimization, PSO）和人类学习优化算法（Human Learning Optimization, HLO）在这类问题中取得了显著成果。PSO 算法擅长处理连续问题，然而在面对多局部极值的函数时，容易陷入局部最优解，导致全局搜索能力不足；并且 HLO 算法中缺少有效策略使算法在探索与挖掘中实现平衡。针对以上问题，研究了具有合作竞争机制的群智能优化算法。通过在算法中引入合作竞争机制，提升了算法的收敛精度，使算法在探索与挖掘中实现平衡。而标准 PSO 和 HLO 算法普遍缺乏合作竞争机制，这导致其在求解复杂优化问题时的效率和解的质量受到限制。因此，如何引入合作竞争机制，增强全局搜索能力，并提高算法的稳定性，成为了优化算法研究中的重要问题。

为验证所提出算法的有效性，本文选取了涉及十进制与二进制的实际工程问题和炉膛火焰识别问题作为典型应用场景。在实际工程中，优化算法广泛应用于结构设计等领域，这些问题通常涉及复杂的设计变量、约束条件以及多目标优化，

传统优化方法在求解时往往面临高维度和局部最优解的挑战。因此，研究更高效的优化算法对于提升设计性能、降低成本和提高安全性具有重要意义。在炉膛火焰识别问题中，精确识别火焰分布信息不仅有助于提升燃料利用效率，增强安全保障，尤其是在面对非线性变化的火焰特征时，能够更准确地捕捉到复杂的火焰分布情况。因此，对具有合作竞争机制的群智能优化算法，并将其应用于复杂工程优化问题以及炉膛火焰识别问题研究具有重要的理论研究意义以及实际应用价值。

## 1.2 国内外研究现状

### 1.2.1 粒子群优化算法

粒子群优化（PSO）算法<sup>[3]</sup>作为一种智能优化算法，已在多个领域得到广泛应用。由于其简单性、高效性以及易于实现的特点，PSO 算法受到了研究者的广泛关注。在 PSO 算法中，解被表示为粒子群中的每个粒子，每个粒子具有自身的位置信息和速度信息。粒子通过与其他粒子共享信息来更新自身的状态，从而在解空间中进行搜索。

在搜索过程中，每个粒子根据其历史最佳位置（个体最优解）和整个粒子群的最佳位置（全局最优解）来调整其速度和位置。Kennedy 和 Eberhart 最初指出，PSO 群体缺乏多样性的原因在于过快的收敛过程。因此，增强群体的多样性被认为是有效跳出局部最优解的关键方法。Chang 等人<sup>[4]</sup>研究了根据不同函数类型对粒子群体进行分类的策略。对于具有两个峰值和两个谷值的函数类型，粒子群体可以被划分为四类，以探索四个极值点，从而增强算法的多样性。然而，增强多样性的同时也可能导致收敛性的减弱。尽管当前的 PSO 算法在不同领域取得了显著成果，但由于不存在一种能够在所有问题领域中表现最优的通用算法，如何进一步提升 PSO 算法的性能，尤其是在平衡收敛性与多样性方面，仍是一个重要的研究方向<sup>[5]</sup>。因此，为了平衡探索与开发的关系，Krink 等人<sup>[6]</sup>研究了一种空间扩展型 PSO 算法，该算法为每个粒子分配了一个实际可触及的半径。当粒子群中的粒子发生碰撞时，可以通过反弹机制触发多样性的增加。

另外，许多研究者尝试将各种元启发式算法与 PSO 算法相结合，以提高其性能和解决复杂问题的能力。文献<sup>[7]</sup>研究了将遗传算法（GA）与 PSO 相结合的方法，实验结果表明，这种结合比传统的连续遗传算法在优化效果上更加优越。

文献<sup>[8]</sup>通过将混合 GA 与 PSO 结合,加入了多模态全局优化策略,利用两者的优点增强了算法的种群多样性,有效提高了解决多峰优化问题的能力。文献<sup>[9]</sup>将差分进化算法(DE)与 PSO 相结合,利用 DE 不依赖历史数据的特点<sup>[10]</sup>,通过三种操作算子的引入,增强了粒子的多样性,同时在每次迭代后通过比较粒子位置的适应值优化搜索过程。这一改进显著提高了粒子群的全局搜索能力。此外,研究者还成功将 PSO 与其他优化算法结合,如蚁群算法(ACO)<sup>[11]</sup>、引力搜索算法(GSA)<sup>[12]</sup>和模拟退火算法(SA)<sup>[13]</sup>,进一步增强了 PSO 算法的局部与全局搜索能力,通过引入新的更新机制提升了种群的多样性,增强了算法摆脱局部最优解的能力,改善了全局搜索性能,并推动了算法之间的协同效应。以上研究主要通过将 PSO 与其他算法进行融合,不仅拓宽了其应用范围,并创新性地引入新的更新机制,成功提升了 PSO 的性能,尤其是在加速收敛和避免局部最优解的挑战方面,取得了显著的成果。

对于 PSO 算法的改进中,大量研究者认为一个有效策略的引入被认为是提高粒子群优化算法(PSO)性能的重要目标。文献<sup>[14]</sup>研究的综合学习粒子群优化算法(CLPSO)虽然注重避免局部最优,但往往导致收敛速度较慢,这凸显了实现探索与开发平衡的问题。为了有效地平衡收敛速度与避免局部最优,研究者们研究了多种改进策略。其中,文献<sup>[15]</sup>交叉策略的引入被证明是提高粒子群优化算法性能的有效途径。交叉策略通过模拟生物种群中基因重组的过程,在搜索空间中生成新的解,从而增加解的多样性,防止算法过早收敛。通过交叉操作,粒子群在进化过程中能够探索到更多的搜索区域,有效避免陷入局部最优解。与此同时,交叉操作还能够加速全局搜索的过程,从而提升收敛速度。此外,文献<sup>[16]</sup>引入了局部搜索策略在提升算法性能方面起到了关键作用。局部搜索策略主要通过当前粒子进行细化搜索,以增加粒子的局部探索能力,进一步提高优化精度。结合全局搜索与局部搜索,算法能够兼顾全局探索和局部精细化,从而有效避免局部最优问题,并加速收敛至全局最优解。另外,在粒子群优化算法中,拓扑结构的选择对算法性能至关重要,因此,研究不同拓扑结构对算法的影响成为了一个重要课题。

拓扑结构对粒子群优化(PSO)算法的性能具有重要影响,因为粒子间的信息传递和相互作用在很大程度上依赖于邻域拓扑的设计。文献<sup>[17],[18]</sup>表明,粒子之间的差异性以及搜索空间的探索能力与所选择的拓扑结构密切相关。文献<sup>[19]</sup>

详细探讨了 PSO 算法中的四种邻域拓扑结构, 评估了这些不同结构对算法性能的影响, 并通过实验分析确定了在特定情况下最优的拓扑结构。值得注意的是, 大多数现有研究中的拓扑结构都是静态的, 即在整个算法执行过程中, 粒子的邻域关系保持不变。然而, 静态拓扑结构在某些情况下可能限制了算法的全局搜索能力, 尤其是在复杂问题中。为了解决这一问题, Suganthan 等人<sup>[20]</sup>研究了一种动态拓扑方法, 旨在通过调整粒子之间的邻域关系, 使其在算法迭代过程中逐步影响所有粒子的相互作用。动态拓扑方法不仅提高了粒子之间的信息交流效率, 而且能够在算法运行过程中灵活地调整粒子之间的合作方式, 从而在探索过程中提供更多的多样性和灵活性, 进一步提升了 PSO 算法的全局搜索能力和收敛速度。

基本粒子群优化算法 (PSO) 虽然在许多优化问题中取得了良好的效果, 但由于其缺乏合作竞争机制, 存在一些显著的问题如易陷入局部最优解、搜索能力受限、缺乏种群多样性等缺点, 限制了其在复杂优化问题中的应用。针对以上问题需要进行进一步研究。

### 1.2.2 人类学习优化算法

近年来, Wang 等人<sup>[21]</sup>研究了基于简化人类学习模型的人类学习算法 (HLO)。人类学习具有随机性、个体性和社会性等特点, 因此, HLO 算法通过引入随机学习算子 (RLO)、个体学习算子 (ILO) 和社会学习算子 (SLO) 来进行选择性学习, 从而有效地控制算法的搜索过程, 旨在寻找到最优解。

由于 HLO 算法具有对学习算子的参数设定敏感的特性, 对此, 很多研究者致力于分析 HLO 算法中不同学习算子的变化对算法性能的影响。Yang 等人<sup>[22]</sup>在标准 HLO 算法的基础上引入了自适应策略, 能够根据实际情况动态调整参数, 从而显著提升算法的搜索能力与鲁棒性。Yang 等人<sup>[23]</sup>将正余弦函数的变化规律引入 HLO 算法的参数改进中, 使得参数能够随时间变化, 从而帮助算法避免陷入局部最优, 并具备周期性的搜索与开发能力。Wang 等人<sup>[24]</sup>研究了学习算子服从高斯分布的多样性人类学习算法, 依据人类智商的高斯分布特性进行算法改进, 进一步增强了算法的鲁棒性。由于随机学习算子 (RLO) 对算法的搜索与开发能力有着重要影响, 对此 DU 等人<sup>[25]</sup>提出对 RLO 算子进行自适应动态调整, 以满足算法在不同阶段的需求, 从而实现搜索与开发之间的平衡。文献<sup>[26]</sup>在标准 HLO 算法中随机学习算子与个体学习算子中引入了随着迭代次数变化而变化的自适

应策略,研究了一种自适应简化的人类学习优化算法(ASHLO),实验结果表明研究的算法优于对比算法,实现了算法在开发与探索间的平衡。文献<sup>[27]</sup>在传统HLO算法中引入了自适应学习策略,通过控制随机学习算子、个体学习算子和社会学习算子的线性变化使算法能够平衡探索与开发,提高了搜索效率和精度,从而在复杂优化问题中取得更好的表现。文献<sup>[28]</sup>将反馈控制思想引入到人类学习优化算法中,通过控制随机学习算子自适应变化并结合随机前向反馈神经网络,研究了基于反馈控制的自适应人类学习优化算法,该方法有效增强了算法在全局优化过程中的适应性和搜索能力,提升了其在复杂优化问题中的性能。另外,文献<sup>[29]</sup>对随机学习算子引入了一种新的自适应调整策略,研究了一种增强的探索与开发平衡的自适应人类学习优化算法,使随机学习算子在开始迭代初期进行线性递增,在迭代中期进行线性递减,有效地加快迭代初期的收敛速度,保持搜索中期的多样性,并在迭代后期进行精确的局部搜索。

随着研究的深入,研究者们认为标准HLO算法的学习算子缺乏类型的多样性,使得算法的功能相对有限,且容易陷入局部最优解,从而限制了算法的性能。针对这类问题,文献<sup>[30]</sup>结合心理学小组学习理论引入一种TBL算子,研究了融合学习心理学的人类学习优化算法(LPHLO),通过小组经验引导个体学习,并结合了动态调参策略使得个体与群体间的信息交流,提升了算法的收敛速度和计算精度。文献<sup>[31]</sup>受到人类婚配的启发设计了一种基于配对机制的人类学习优化算法,在HLO算法中引入了配对学习算子,通过双人学习过程将学习经验存于PKD知识库,提高了学习效率,有效提升算法性能。Zhang等人<sup>[32]</sup>研究了一种基于多个社会信息的社会推理学习算子(SRLO),通过引入额外的算子形式来增强算法的全局搜索能力,从而有效改善算法的性能。此外,鉴于贝叶斯推理能够帮助有效完成任务规划,Zhang等人<sup>[33]</sup>基于贝叶斯推理机制研究了一种贝叶斯推理学习算子(BILO)。该算子通过累积搜索过程中的信息,并利用推理方法确定最优位置,从而提高了学习效率并进一步提升了算法的整体性能。

人类学习优化算法作为模仿人类学习和认知过程的启发式优化方法,在解决复杂优化问题时显示出较好的性能。然而,基本人类学习优化算法由于其缺乏合作竞争机制,导致易陷入局部最优解、探索能力不足、缺乏灵活的全局搜索机制等问题,针对上述不足,需要对人类学习优化算法进行进一步研究。

### 1.2.3 复杂工程优化问题应用

优化算法研究的主要目的是为了解决实际问题,尤其是在各个工程、制造和技术领域中用于改进系统的效率和性能。对此,文献<sup>[34]</sup>将融合 Fuch 混沌映射与精英反向学习策略的海鸥算法应用于工程设计优化问题,避免了算法早期陷入局部最优解,同时结合精英反向学习策略提高了局部搜索的精度和效率。在工程设计优化过程中,算法成功地解决了多个设计参数的调优问题,包括结构设计、材料选择、尺寸优化等方面。通过对设计参数的精确调整,算法显著提高了设计的精度与效率,确保了设计方案在满足功能要求的同时,优化了成本和资源利用。文献<sup>[35]</sup>针对路面车辆优化设计的特点,研究了使用遗传算法对车辆悬架系统进行优化设计。该研究通过结合车辆悬架系统的多目标优化需求,运用遗传算法对悬架参数进行调整,以优化车辆的行驶稳定性、舒适性及耐久性。在该过程中,遗传算法通过模拟自然选择和遗传机制,探索最优的悬架设计方案,同时考虑到车辆行驶过程中的动态响应、振动控制以及对不同路面条件的适应性。针对这些复杂问题的优化求解,火焰识别作为一种实际的应用问题,也面临类似的挑战。

由于高效的火焰识别模型能够为各类工业、环保和安全监控等领域提供重要的技术支持,特别是在提高生产安全性、优化能源利用、保障环境健康等方面具有深远的影响。对此 Celik 等人<sup>[36]</sup>研究了一种基于自适应背景减除的火焰统计颜色模型,利用自适应背景减除算法来提取火焰图像的前景像素,然后通过火焰统计颜色模型来辨识初步提取像素中的火焰信息,实验结果表明研究的方法可有效地识别出炉膛燃烧火焰信息。Chen 等人研究了一种利用卷积神经网络(CNN)的多火焰检测方案(MFD)<sup>[37]</sup>,通过深度学习使用 TensorFlow 来训练火焰图像数据集,构建了火焰识别模型。为了进一步改善火焰识别的效果,多种高级的炉膛火焰识别方法被研究。文献<sup>[38]</sup>引入了一种基于 SVM 的火焰图像识别算法的研究,创新之处在于该算法克服了神经网络系统的过度性学习以及易陷入局部最优等缺点,使得在解决小样本和非线性问题的识别问题上效果显著。Li 等人研究出一种基于非均匀特征和模糊积分的回转窑烧结过程火焰图像燃烧状态识别模型<sup>[39]</sup>,运用一组异构特征和融合技术,通过表征火焰图像的不同特点,从像素中直接提取,无需分割。基于图像的燃烧状态识别系统在识别精度和对窑内烟尘干扰的鲁棒性方面优于其他方法。

上述对复杂工程优化问题以及火焰识别方法的研究虽然取得了一定成效,但仍然存在一些问题,在处理部分复杂工程优化问题中,算法的求解精度仍有提升空间,优化结果可能受参数选择和搜索策略的影响,导致局部最优解的问题未能完全避免。此外,在火焰识别方法,计算成本较高,优化过程中仍然可能出现误差较大的情况,影响优化的稳定性和鲁棒性。因此,为了进一步提升优化效果,需要开拓其他途径处理上述问题。

### 1.3 主要研究内容

根据上述存在的问题,受自然界个体合作竞争现象的启发,本论文第二章研究了具有合作竞争机制的粒子群优化算法,进一步分析搜索过程中种群对信息的利用效率差异特征,研究了具有信息交互式合作竞争机制粒子群优化算法,有效提升了算法性能。第三章中,在人类学习优化算法中引入合作竞争机制,研究了具有合作竞争机制的人类学习优化算法。进一步考虑合作竞争学习算子与社会学习算子对种群信息的利用效率差异,研究了具有增强探索与开采平衡合作竞争机制的人类学习优化算法,以提升算法的探索与开采能力。第四章中,将所提出算法应用于复杂工程优化问题中,并以炉膛燃烧场火焰识别问题为典型应用案例,研究了基于 HSV 颜色模型的炉膛火焰识别方法和基于新型神经网络的炉膛火焰识别方法,验证所提出算法的求解能力。全文的研究内容框架如图 1-1 所示,主要研究内容如下:

第 1 章:概述论文的选题背景及意义,分析了目前粒子群优化算法和人类学习算法以及复杂工程优化问题和炉膛火焰识别方法的研究现状。

第 2 章:针对粒子群优化算法的缺陷,研究了一种具有合作竞争机制的粒子群优化算法(PSOCC)。通过参数学习分析,评估了不同参数组合对算法性能的影响,并优化了参数配置以提升算法效果。同时,采用 AD 曲线分析了 PSOCC 在求解空间中的收敛性。此外,进一步考虑了搜索过程中种群信息利用效率的差异性,研究了一种具有信息交互式合作竞争机制的粒子群优化算法(IPSOCC),通过数值实验深入分析了不同参数对算法性能的影响。最后,为了验证算法性能,将改进算法与对比算法应用于 23 个测试集函数,并对实验结果进行了比较分析,以验证改进算法性能的优越性。

第3章：受自然界种群的合作竞争启发，设计了一种合作竞争学习算子，研究了具有合作竞争机制的人类学习优化算法，并通过数值实验系统分析了所提出学习算子的作用及其功能，结果表明该算子能够有效增强算法搜索能力。此外，

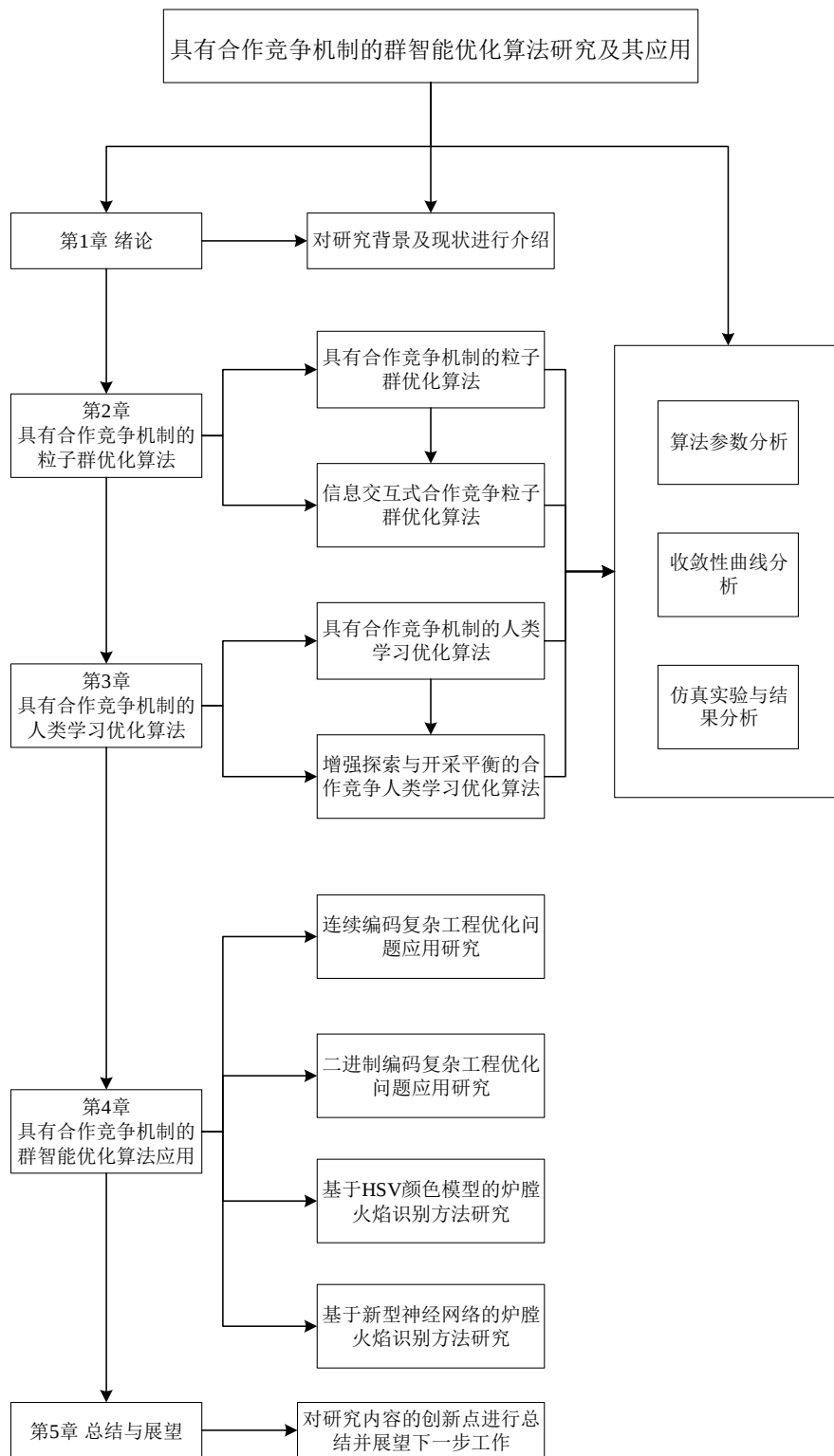


图 1-1 研究内容框架图

在 CCHLO 算法中引入重新学习机制，以维持种群的多样性，从而显著提升算法的探索能力。在此基础上，进一步探讨了合作竞争学习算子与社会学习算子在种群信息利用效率方面的差异，构建了一种增强探索与开采平衡的合作竞争模型，并研究了一种增强探索与开采平衡的合作竞争人类学习优化算法。通过数值实验，验证了该增强模型能够有效促进探索与开采的动态平衡，进而提升算法的全局优化能力。

第 4 章：将所研究的 IPSOCC 算法和 ECCHLO 算法分别应用于复杂工程问题，这为优化工程设计和提高系统性能提供了新的解决方案，具有重要的实践意义。其次，以炉膛燃烧场火焰识别问题为典型应用案例，构建了两种炉膛火焰识别模型（HSV 颜色模型、新型神经网络模型），并将 IPSOCC 算法和 ECCHLO 算法优化所提出的模型，旨在提升火焰识别的精度和鲁棒性。

第 5 章：对全文的研究内容、关键方法和主要成果进行了系统归纳与总结。总结章不仅梳理了算法在提升神经网络分类性能、优化效率以及鲁棒性等方面的贡献，还对算法在实际工程中的应用潜力进行了展望。此外，针对研究中存在的局限性与不足，研究了未来改进的方向，为进一步研究提供了思路 and 参考。

## 第 2 章 具有合作竞争机制的粒子群优化算法研究

### 2.1 基本粒子群优化算法

在标准粒子群优化（PSO）算法中，群体的运动不受限制，不断在可能的解空间中搜索最优解。粒子在搜索空间中能够保持稳定的运动，同时根据环境变化调整其运动方式。粒子群算法通过群体中个体信息的共享，促进了群体在问题求解空间中的演化过程，从而实现了最优解的搜索。自标准粒子群优化算法提出以来，经过不断演化，已经形成了多个不同版本，如新的离散粒子群优化算法（Discrete Particle Swarm Optimization, DPSO）<sup>[40]</sup>、二进制粒子群优化算法（Binary Particle Swarm Optimization Algorithm, BPSO）<sup>[41]</sup>等。这些算法的提出是为了解决一些实际问题，成为了应对特定问题求解的有力工具。

PSO 算法在处理问题时每个粒子对应的位置就是问题的解，每个粒子都有自己的速度和位置，在迭代进行时会向着自己的历史最佳位置和全局最佳位置以及原先自己的速度去更新自己下一代的速度，粒子的速度和位置更新方程如下：

$$v_i^t = w \cdot v_i^{t-1} + c_1 \cdot r_1 \cdot (pBest_i - x_i^{t-1}) + c_2 \cdot r_2 \cdot (gBest - x_i^{t-1}) \quad (2.1)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (2.2)$$

其中， $t$  为当前迭代次数， $i$  为粒子编号， $w$  为惯性权重， $c_1$ 、 $c_2$  为学习因子， $gBest$  为全局最优解， $pBest$  为当前粒子历史最优值， $r_1$  和  $r_2$  为在  $[0, 1]$  之间的随机值，可以看到粒子更新主要还是根据  $pBest$  和  $gBest$  的位置进行更新下一代粒子速度，粒子每一维的速度都会被限定在  $[-v_{max}, v_{max}]$  之间，速度太大粒子会飞离最优位置，速度太小则会容易陷入局部最优，粒子的位置也会给予限定在  $[-x_{max}, x_{max}]$ ，再以原先位置加上速度成为子代位置，以上被成为标准粒子群算法。基本 PSO 流程可以描述为：

**步骤 1：** 将所有粒子的位置和速度初始化为零，然后随机设置粒子的位置，确保粒子在问题空间中随机分布。接着，记录粒子的适应度值，并将该值作为粒子的历史局部最优解。由于后代粒子的位置更新公式需要使用这些信息更新，因此这是至关重要的一步。然后初始化粒子的速度，给粒子赋予一个初始速度，这是公式（2.1）所需的重要信息。

**步骤2:** 评估每个粒子的历史最优解的适应度值, 并将适应度最优的粒子记录为历史全局最优解。根据公式 (2.1), 根据上一代粒子的记录信息更新粒子的速度, 再通过公式 (2.2) 更新粒子的位置, 并规定速度和位置的范围限制。计算当前迭代中粒子的适应度值, 并根据该值更新粒子的历史局部最优解。根据粒子的历史局部最优解更新全局历史最优解。

**步骤3:** 判断当前迭代次数是否达到预定的最大迭代次数。如果达到, 结束迭代并记录过程中的最佳适应度值。如果未达到, 则返回步骤 2, 继续进行下一轮迭代, 直到满足终止条件。

算法中的粒子可以根据环境的变化来更新自己的位置和速度, 即满足邻近性和质量的要求。此外, PSO 算法中的群体不限制其运动, 而是在可能的解空间中不断搜索最优解。粒子群算法中的粒子能够在搜索空间中保持稳定的运动, 同时改变自身的运动方式以适应环境的变化。粒子群算法利用群体中的个体信息的共享使整个群体的运动在问题求解空间中产生的一种演化过程从而获得最优解。

基本粒子群优化算法 (PSO) 虽然因其简单性和高效性在许多优化问题中得到广泛应用, 但仍存在一些显著的局限性。具体而言, PSO 容易陷入局部最优解, 且对参数的设置高度敏感。此外, 基本 PSO 在处理高维度问题时, 其搜索效率会显著下降, 从而影响算法在复杂多目标优化任务中的表现。这些缺点限制了其在某些复杂应用场景中的有效性和适应性。

## 2.2 具有合作竞争机制的粒子群优化算法

在自然群体学习中, 个体之间可以通过相互合作与竞争来提升学习效果, 从而促进整体智慧的进化。合作使得个体能够共享知识、技能和经验, 加速信息传播, 提高学习效率; 竞争则激发个体不断优化自身策略, 提升适应能力, 从而推动整体群体的进步。通过集体智慧, 个体不仅能够快速适应环境变化, 还能在复杂、多变的环境中进行有效的学习和决策。因此, 针对基本 PSO 算法中涉及的问题, 本文引入了合作竞争机制, 研究了具有合作竞争机制的粒子群优化算法 (Particle Swarm Optimization with Cooperative and Competitive, PSOCC)。该改进策略包括粒子协同共享搜索策略、粒子竞争部署策略以及粒子停滞处理策略, 旨在增强粒子群优化算法的性能。通过合作竞争机制的引入, 算法在全局搜索能力和收敛速度等方面得到了显著提升, 从而能够更有效地解决复杂的优化问题。

### 2.2.1 初始化

在初始化阶段，基本粒子群优化（PSO）算法仅对粒子的最大速度进行了限制。然而，由于不同优化问题的求解空间大小存在差异，固定的最大速度限制可能导致算法效率的下降。在求解空间较大的情况下，粒子速度的变化范围过小，导致搜索过程缓慢，降低了算法的全局搜索能力。而在求解空间较小的情况下，粒子速度的变化范围过大，可能导致粒子越过最优解，影响算法的收敛性能。

为了应对这一问题，研究了一种自适应调整最大速度的策略，如公式(2.3)：

$$v_{max} = A \cdot ub \quad (2.3)$$

其中， $A$  为求解空间与粒子最大速度的关系权重（根据经验，这里  $A$  取为 0.04）， $ub$  为求解空间的上限。通过动态调整粒子的最大速度范围，使其能够适应不同求解空间的规模变化，从而提高算法的计算效率。该策略有效平衡了粒子的探索与开发能力，使算法能够在不同的应用环境中实现高效运行。

### 2.2.2 粒子协同共享搜索策略

基本粒子群优化算法（PSO）在速度更新过程中仅依赖于粒子自身的历史最优值以及当前种群的全局最优值进行学习，该策略可能导致粒子之间缺乏有效的协作与信息共享，从而影响优化性能。由于单个粒子在搜索过程中主要围绕自身经验和全局最优解调整速度，缺乏更广泛的信息交流，可能导致算法的多样性不足，使得粒子群过早收敛于局部最优解。当解空间复杂且包含多个局部最优点时，PSO 可能无法跳出局部陷阱，进而影响全局优化能力。这会使得 PSO 算法在高维、非线性、强耦合以及动态优化问题中表现受限，难以确保所识别的位置为整个解空间的全局最优值。

为了解决这一问题，本文于种群  $N_1$  中引入了一种粒子协同共享搜索策略，基于合作理念来增强粒子群体的协作性与信息共享能力。该策略通过计算当前粒子与其他所有粒子之间的欧几里得距离，从而选择它们信息交流的对象，促进粒子之间的相互联系。随着粒子的移动，附近的粒子会发生动态变化，这种变化促使相邻粒子之间的协作与信息交换，从而形成一个紧密合作的粒子群体。在这一过程中，粒子不仅能够独立地探索解空间，同时也依赖于周围粒子搜索的历史经验，以局部集体智慧来提升全局最优解发现的速度。具体粒子协同共享搜索策略如图 2-1 所示：

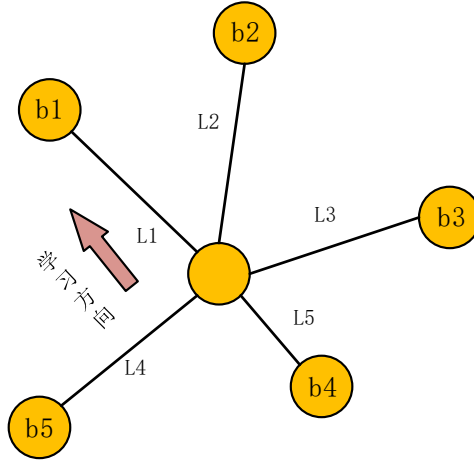


图 2-1 粒子协同共享搜索策略示意图

图中  $L1, L2, L3, L4, L5$  为当前粒子与周围欧氏距离最小的几个粒子的欧氏距离,  $b1, b2, b3, b4, b5$  为周围粒子的历史最优适应度值, 当  $b1 < b2 < b3 < b4 < b5$  时, 粒子的一个速度方向的分量将会朝着粒子历史适应度值为  $b1$  的粒子进行学习, 而不是朝着全局最优适应度的粒子学习。

通过结合公式 (2.4) 计算当前粒子与周围粒子的欧氏距离, 通过该策略, 粒子能够更全面地了解整个解空间的分布情况, 进而提升算法的全局搜索能力和收敛效果, 避免陷入局部最优。

$$d(x_i, x_j) = \sqrt{\sum_{j=1}^{N-1} (x_i - x_j)^2} \quad (2.4)$$

其中,  $d(x_i, x_j)$  是所有粒子和当前粒子之间的欧几里得距离,  $x_i$  是当前粒子的位置,  $x_j$  是其他粒子的位置。通过对当前粒子与其他粒子的欧几里得距离进行排序, 选中与当前粒子欧几里得距离最短的  $L$  个粒子进行合作。如公式 (2.5) :

$$d_1(x_i, x_{j1}) \leq d_2(x_i, x_{j2}) \leq d_3(x_i, x_{j3}) \leq \dots \leq d_L(x_i, x_{jL}) \quad (2.5)$$

其中,  $x_{jL}$  为距当前粒子  $x_i$  的欧几里得距离排序第  $L$  的粒子。在合作过程中, 需要评估每个被选中粒子的历史最优信息, 然后, 对它们各自的历史最佳值进行排名, 确定排名靠前的两个粒子。如公式 (2.6) :

$$fit(pBest_{ME}^t) \leq fit(pBest_{SO}^t) \leq \dots \leq fit(pBest_{jL}^t) \quad (2.6)$$

其中,  $fit(pBest_{ME}^t)$  为选取的合作粒子中最优的历史最优适应度,  $fit(pBest_{SO}^t)$  为选取的合作粒子中次优的历史最优适应度。根据公式 (2.7), 为选中的最优适应度粒

子与次优适应度粒子分配一个介于 0 和 1 之间的随机值,以进一步引导粒子的更新方向。

$$p_i^t = \begin{cases} pBest_{ME}^t & r_3 > K \\ pBest_{SO}^t & r_3 \leq K \end{cases} \quad (2.7)$$

其中,  $p_i^t$  被选为更新公式中的历史个体最优值,  $pBest_{ME}^t$  是当前粒子附近粒子的历史个体最佳值,  $pBest_{SO}^t$  是该粒子附近的历史个体优化值,  $r_3$  是[0,1]之间的随机值。当随机值低于阈值  $K$  时,粒子会从排名靠前的粒子的位置中选择历史最优值。相反,如果随机值超过阈值,则粒子将采用与排名第二的次优粒子的历史最优值相关的位置。速度更新公式如公式 (2.8) :

$$v_i^{t+1} = w \cdot v_i^t + c_1 r_4 (p_i^t - x_i^t) + c_2 r_5 (gBest^t - x_i^t) \quad (2.8)$$

其中,  $t$  是当前迭代次数,  $r_4$  和  $r_5$  是[0, 1]的随机值,  $c_1$  和  $c_2$  是学习因子。

粒子协同共享搜索策略的引入使得粒子能够在未知的搜索空间中扩大其感知范围,从而获取比标准更新公式所提供的更多位置信息。增强的更新机制不仅使粒子跟随最优解的引导,还引入了次优解的考虑,通过避免粒子在有限区域内聚集于单一粒子的历史最优解附近,有效降低了陷入局部最优解的风险,从而避免粒子过早收敛以增强算法整体的多样性。

### 2.2.3 粒子竞争部署策略

为了在迭代后期提高算法的局部搜索能力,在迭代  $T_K$  次时计算粒子适应度后,在迭代后的种群  $N_1$  中的粒子之间进行适应度竞争。这些粒子将竞争并相应地排名,排名较低的粒子将失去其原始位置,并根据公式 (2.10) 重新分配位置。

$$\begin{cases} R = 1 - t / T_{max} \\ lb^* = gBest^t \cdot (1 - R) \\ ub^* = gBest^t \cdot (1 + R) \end{cases} \quad (2.9)$$

$$x_i^t = lb^* + r_6 \cdot (ub^* - lb^*) \quad (2.10)$$

其中,  $T_{max}$  是最大迭代次数,  $lb^*$  是随机分布的下限,  $ub^*$  是随机分布的上限。该过程将适应度较低的粒子重新分配到更有利的位置,将资源集中在全局最优粒子附近,旨在增强算法在局部区域的搜索能力,从而提升其识别最优解的精度。在资源有限的情况下,该策略能够通过优先支持最有可能获得最优解的粒子,优化搜

索过程，避免不必要的资源浪费，并加速收敛。通过这种策略，算法能够在后续迭代中有效地利用有限的资源，精确地探索和定位最优解区域。

#### 2.2.4 粒子停滞处理策略

影响标准粒子群优化算法中速度更新公式的另一个关键因素是全局最优值。仅仅依赖全局最优值可能会使粒子在整个迭代过程中被限制在特定区域，从而限制了搜索的多样性和全局探索能力。为了解决这一问题，本文在种群  $N_2$  的粒子更新公式中引入了多策略停滞处理。

$$T = \begin{cases} T+1 & \text{if } gBest^t = gBest^{t-1} \\ 0 & \text{else} \end{cases} \quad (2.11)$$

其中， $T$  为全局最优不变的次数， $gBest^{t-1}$  为上次迭代中的全局最优值。如果算法陷入局部最优，并且  $T_1$  次迭代后全局最优值的适应度保持不变，则表明将全局粒子锁定在  $gBest$  附近。此时，有必要转换更新粒子速度的公式，并消除  $gBest$  对粒子的影响，使粒子以不受全局效应影响的方式更新其速度，如公式（2.12）：

$$v_i^{t+1} = \begin{cases} w \cdot v_i^t + c_1 r_7 (pBest_i^t - x_i^t) + c_2 r_8 (gBest^t - x_i^t) & \text{if } T_1 > T \\ w \cdot v_i^t + c_1 r_9 (pBest_i^t - x_i^t) & \text{if } T_2 > T > T_1 \end{cases} \quad (2.12)$$

多策略停滞处理的目的是防止  $gBest$  在整个迭代过程中过度依赖单一值，从而保持种群多样性避免多样性下降。通过消除粒子对陷入局部最优  $gBest$  的依赖，粒子可以根据自己的个人意图探索空间，而不受  $gBest$  影响，从而提高算法全局搜索性能。

$$x_i^t = lb + r_{10} \cdot (ub - lb) \quad (2.13)$$

当全局适应度值在  $T_2$  次迭代中保持不变时，种群  $N_2$  中的所有粒子都会在解空间内重新初始化，如公式（2.13）。该策略旨在通过迫使粒子超越局部范围，并通过迭代过程探索替代可能性，从而促进逃离局部最优解。

#### 2.2.5 PSOCC 算法实现

在具有合作竞争机制的粒子群优化算法中，首先对算法进行初始化以初始化粒子位置和速度，随后记录关键数据。并对相应种群中的粒子执行粒子协同共享搜索策略、粒子竞争部署策略以及粒子停滞处理策略以寻找求解空间中的最优适应度值，直至达到终止条件。具体流程图如图 2-2 所示：

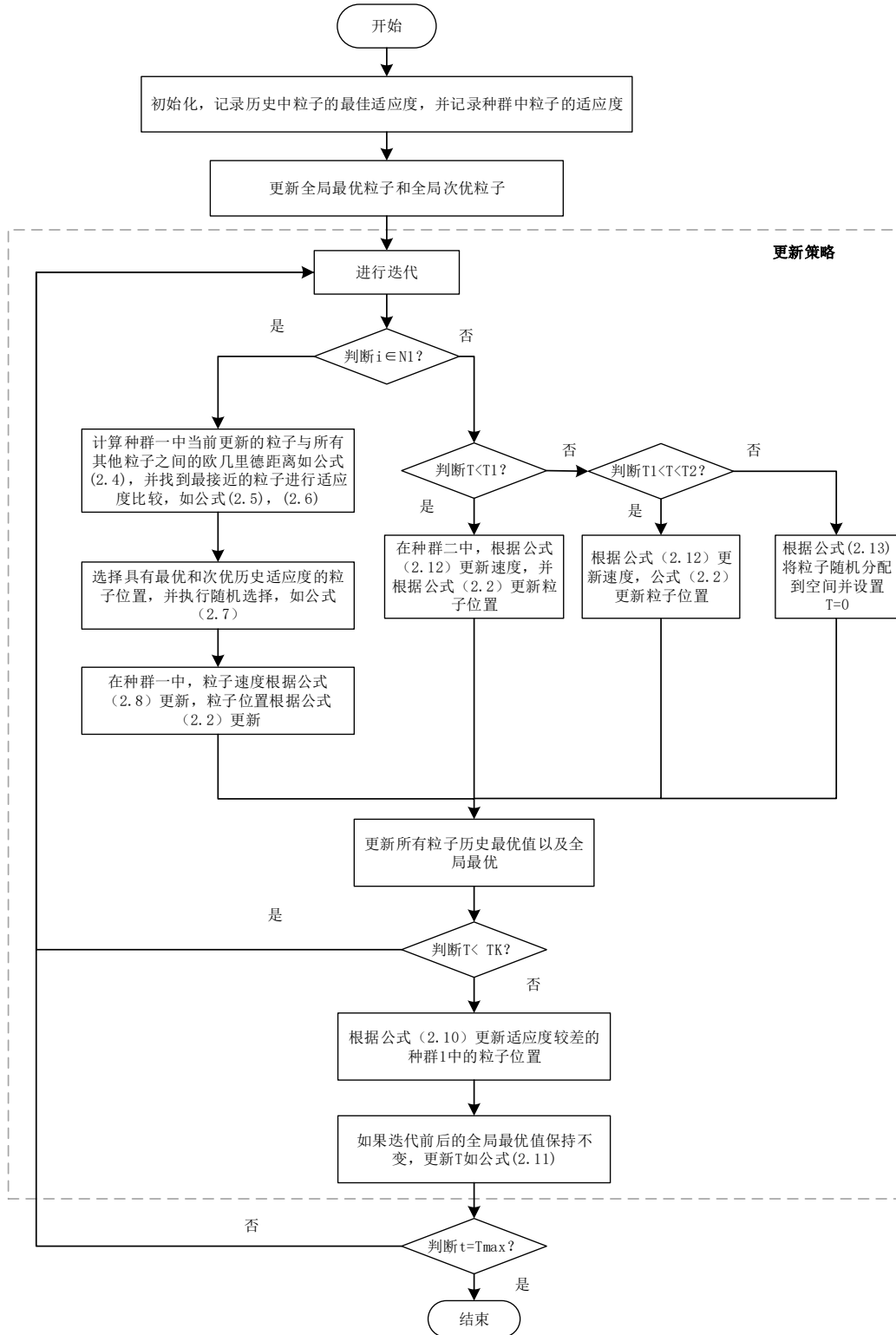


图 2-2 PSOCC 算法流程图

### 2.2.6 PSOCC 算法参数分析

在具有合作竞争机制的粒子群优化算法中, 考虑到惯性权重与学习因子在 PSOCC 算法中的功能与 PSO 算法一致, 因此, 实验的该参数设定与 PSO 算法保

持一致（即  $c_1 = c_2 = 2, w = 0.8$ ）。合作竞争机制中其他参数  $A$ 、 $T_1$ 、 $T_2$  和  $T_k$  分别设定为 0.04、50、100 和 500。此外，由于新提出的合作竞争机制无法直接应用，有个别参数需要深入分析（即合作的粒子数量  $L$ ，以及阈值  $K$ ）。两个参数所对应的不同数值组合都会影响 PSOCC 算法的优化性能，如  $L$  值较大会选择更多的粒子进行协作，粒子可以获得来自更多个体的信息，有助于提高收敛速度，避免早期收敛于局部最优解。但较小的  $L$  会使粒子只依赖于少数几个邻居的信息，导致对全局信息的忽视，影响整体搜索效果。针对阈值  $K$  的取值，直接影响当前粒子的学习对象是合作粒子中最优的历史最优值还是次优的粒子历史最优值，这一选择对优化过程的动态平衡具有重要影响。因此，合理调节  $L$  和  $K$  的值是提升算法性能的关键因素。

针对这一问题将不同的  $L$  与  $K$  值的组合构成了正交表 2-2，参数  $L$  和  $K$  的取值如表 2-1。选择 CEC2005 测试集中的 F1 和 F10 基准函数，由于 F1 和 F10 基准函数具有不同的优化特性，能够全面评估算法在不同类型问题上的表现。F1 为单峰函数，适用于测试算法在处理较简单优化问题时的收敛性和精度；而 F10 是一个多峰函数，具有多个局部最优解，适合评估算法在面对复杂、非线性、多峰问题时的全局搜索能力以及避免陷入局部最优解的能力。因此，选择这两个函数能够综合考察算法在不同复杂度问题上的性能。对于参数测试，本研究在 30 维度和 50 维度上分别进行了算法参数的评估。低维度问题（如 30 维）中，算法的搜索空间相对较小，更易找到全局最优解，主要考察算法在不同参数设置下的收敛性和稳定性。较高维度问题（如 50 维）则具有更为复杂的搜索空间，算法需要具备更强的全局探索能力和局部搜索能力。因此，选择这两个不同维度进行测试，能够充分评估算法在不同参数配置下的优化效率、鲁棒性和收敛性能。

表 2-1 PSOCC 分析参数  $L$  和  $K$  的取值情况

| 参数  | 取值情况 |      |      |      |      |      |
|-----|------|------|------|------|------|------|
|     | 取值 1 | 取值 2 | 取值 3 | 取值 4 | 取值 5 | 取值 6 |
| $L$ | 4    | 5    | 6    | 7    | 8    | 9    |
| $K$ | 0.3  | 0.4  | 0.5  | 0.6  | 0.7  | 0.8  |

在表 2-2 中，通过对每个基准函数在固定维度下的数据进行排名，并将排名前十的数据加粗，以突出算法在不同参数设置下的优越表现，便于直观比较各参数配置的效果。通过加粗排名前十的数据，便于清晰地观察不同参数组合在多个测试函数中表现出不同的优化性能，以筛选出最优参数组合。分析结果可知，当

参数  $L$  和  $K$  分别取 9 和 0.4 时, 对应的加粗项最多, 说明该参数组合在算法优化过程中表现出更强的全局搜索能力和更高的收敛效率, 表明 PSOCC 算法在  $L$  和  $K$  分别取 9 和 0.4 时算法的综合性能最优。

表 2-2 参数正交表

| 序号 | 参数  |     | F1              | F10             | F1              | F10             |
|----|-----|-----|-----------------|-----------------|-----------------|-----------------|
|    | $L$ | $K$ | D=30            |                 | D=50            |                 |
| 1  | 4   | 0.3 | 7.29E-01        | 4.12E-04        | 3.35E-02        | 2.17E+00        |
| 2  | 4   | 0.4 | <b>4.80E-01</b> | 7.09E-04        | 5.61E-02        | 2.40E+00        |
| 3  | 4   | 0.5 | <b>4.62E-01</b> | 1.11E-03        | 6.15E-02        | 2.12E+00        |
| 4  | 4   | 0.6 | 7.21E-01        | 1.57E-03        | 1.14E-01        | <b>1.74E+00</b> |
| 5  | 4   | 0.7 | 5.57E-01        | 2.27E-03        | 1.40E-01        | 1.90E+00        |
| 6  | 4   | 0.8 | 6.12E-01        | 3.61E-03        | 2.38E-01        | <b>1.69E+00</b> |
| 7  | 5   | 0.3 | 5.70E-01        | 9.15E-05        | 1.32E-02        | <b>1.84E+00</b> |
| 8  | 5   | 0.4 | <b>5.33E-01</b> | 1.83E-04        | 1.75E-02        | 2.07E+00        |
| 9  | 5   | 0.5 | <b>4.94E-01</b> | 2.90E-04        | 3.28E-02        | 2.46E+00        |
| 10 | 5   | 0.6 | <b>4.53E-01</b> | 4.87E-04        | 3.45E-02        | 2.20E+00        |
| 11 | 5   | 0.7 | 6.68E-01        | 6.93E-04        | 7.17E-02        | <b>1.43E+00</b> |
| 12 | 5   | 0.8 | 7.37E-01        | 8.64E-04        | 8.77E-02        | 2.13E+00        |
| 13 | 6   | 0.3 | 7.64E-01        | 2.44E-05        | <b>6.03E-03</b> | <b>1.81E+00</b> |
| 14 | 6   | 0.4 | 8.79E-01        | 3.62E-05        | 7.99E-03        | 2.14E+00        |
| 15 | 6   | 0.5 | 6.80E-01        | 9.46E-05        | 1.55E-02        | 2.03E+00        |
| 16 | 6   | 0.6 | 6.79E-01        | 1.42E-04        | 1.62E-02        | 2.02E+00        |
| 17 | 6   | 0.7 | 7.36E-01        | 3.63E-04        | 3.47E-02        | <b>1.74E+00</b> |
| 18 | 6   | 0.8 | 5.39E-01        | 2.82E-04        | 4.49E-02        | 2.05E+00        |
| 19 | 7   | 0.3 | 7.07E-01        | <b>1.39E-05</b> | <b>3.17E-03</b> | 2.00E+00        |
| 20 | 7   | 0.4 | 6.75E-01        | <b>1.87E-05</b> | <b>4.35E-03</b> | 2.35E+00        |
| 21 | 7   | 0.5 | <b>4.76E-01</b> | 3.48E-05        | 6.14E-03        | <b>1.88E+00</b> |
| 22 | 7   | 0.6 | 7.71E-01        | 5.51E-05        | 9.56E-03        | 2.00E+00        |
| 23 | 7   | 0.7 | 5.47E-01        | 1.16E-04        | 1.46E-02        | <b>1.87E+00</b> |
| 24 | 7   | 0.8 | 6.74E-01        | 1.28E-04        | 1.69E-02        | 2.06E+00        |
| 25 | 8   | 0.3 | 7.13E-01        | <b>6.35E-06</b> | <b>1.67E-03</b> | <b>1.67E+00</b> |
| 26 | 8   | 0.4 | 5.71E-01        | <b>1.08E-05</b> | <b>2.18E-03</b> | 2.15E+00        |
| 27 | 8   | 0.5 | <b>4.93E-01</b> | <b>1.41E-05</b> | <b>4.71E-03</b> | 2.09E+00        |
| 28 | 8   | 0.6 | 6.60E-01        | <b>2.37E-05</b> | 6.36E-03        | 2.00E+00        |
| 29 | 8   | 0.7 | 5.72E-01        | 5.04E-05        | 9.17E-03        | 2.09E+00        |
| 30 | 8   | 0.8 | <b>4.70E-01</b> | 7.52E-05        | 1.45E-02        | 2.03E+00        |
| 31 | 9   | 0.3 | 6.44E-01        | <b>2.86E-06</b> | <b>1.26E-03</b> | 2.37E+00        |
| 32 | 9   | 0.4 | <b>5.38E-01</b> | <b>3.33E-06</b> | <b>1.55E-03</b> | <b>1.75E+00</b> |
| 33 | 9   | 0.5 | 7.51E-01        | <b>8.89E-06</b> | <b>2.56E-03</b> | 1.93E+00        |
| 34 | 9   | 0.6 | 5.75E-01        | <b>1.63E-05</b> | <b>3.20E-03</b> | 2.04E+00        |
| 35 | 9   | 0.7 | <b>4.58E-01</b> | 2.40E-05        | 6.38E-03        | 2.27E+00        |
| 36 | 9   | 0.8 | 5.71E-01        | 4.33E-05        | 7.42E-03        | 2.22E+00        |

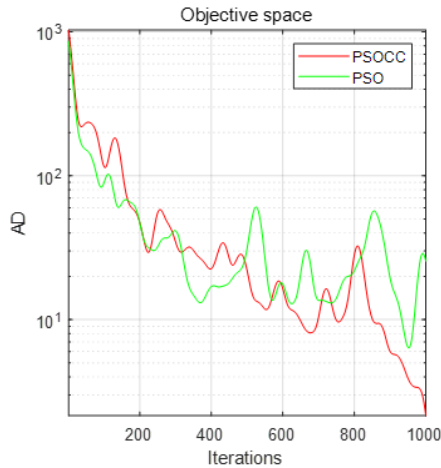
### 2.2.7 PSOCC 算法收敛性分析

使用平均距离曲线（AD 曲线）对算法进行收敛性分析，AD 曲线是使其他粒子与全局最优粒子的平均距离随着迭代的变化而变化的曲线。平均距离曲线可以反映数据点之间的平均距离随着类别之间的距离变化的情况，如，种群中个体之间的分离度、数据集的聚类情况以及进行算法的性能对比。因此，选择用 AD 曲线与及适应度曲线来检测算法对于空间的探索与开发能力。如公式（2.14）。

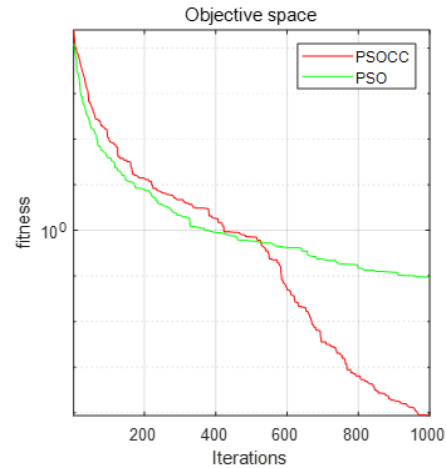
$$AD = \frac{\sum_{i=1}^{N-1} |x_i - gBest|}{N-1}, 1 \leq i \leq N-1 \quad (2.14)$$

当算法处于早期阶段，粒子的位置相对分散，平均距离通常较大。这一阶段，粒子探索空间较广，以寻找潜在的优质解。随着迭代次数的增加，粒子逐渐向全局最优粒子聚集，平均距离开始减小。这表明粒子群正在有效地利用全局最优粒子的信息来调整自身的位置。因此，通过 AD 曲线可以清晰反映算法中的粒子在迭代过程中的收敛情况。

通过图 2-3、图 2-4 可以看出与基本 PSO 算法相比 PSOCC 在搜索效率和收敛速度上有显著提升。PSOCC 算法有着比 PSO 算法更大的平均距离递减斜率，意味着在 PSOCC 中粒子群更迅速地聚集到全局最优附近，而适应度曲线的加速下降则表明算法能更快速找到优质解，从而提升了整体优化性能并减少了陷入局部最优的可能性。

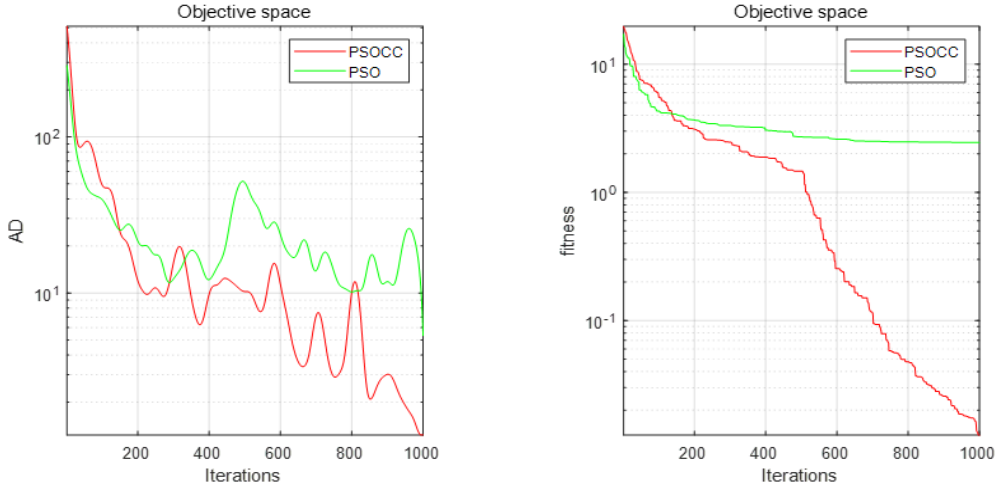


(a) F1 函数中的 AD 曲线



(b) F1 函数中的适应度曲线

图 2-3 PSOCC 与 PSO 在 F1 函数中的 AD 曲线与适应度曲线



(a) F10 函数中的 AD 曲线

(b) F10 函数中的适应度曲线

图 2-4 PSOCC 与 PSO 在 F10 函数中的 AD 曲线与适应度曲线

## 2.3 信息交互式合作竞争粒子群优化算法

### 2.3.1 信息交互式合作竞争策略

在 PSOCC 中, 虽然原有的算法已具备局部挖掘和空间探索的分化策略, 但考虑搜索过程中种群对信息的利用效率差异特征。因此, 为了进一步提升算法的优化效率, 本研究在 PSOCC 算法的基础上引入了信息交互式策略, 研究了一种信息交互式合作竞争粒子群优化算法 (Interactive Particle Swarm Optimization with Cooperative and Competition, IPSOCC)。

该机制通过种群间的资源抢夺机制来增强算法的适应性。当某一种群的全局最优解频次在一定时间内较高时, 将会为该种群分配更多的粒子资源, 以增强其局部搜索能力。IPSOCC 算法将粒子群划分为两个子种群, 其中种群  $N_1$  偏向于局部挖掘, 种群  $N_2$  专注于空间探索。通过该分化策略, 使算法根据当前的搜索需求灵活调整策略, 在不同阶段实现探索与利用之间的动态平衡, 提升整体优化效率。

为了动态调整资源, 将种群间的全局最优解成功次数  $s_k$  作为资源分配依据。在时间区间  $[t, t+\Delta t]$  内, 这里  $\Delta t$  取值设为 100, 种群  $N_1$  获得全局最优解的次数为  $s_k$  反映了该种群在该时间段内的优化效果和搜索能力。因此, 通过跟踪每个种群在该时间区间内达到全局最优解的次数, 可以合理评估其搜索质量, 并据此调整资源的分配, 满足公式 (2.15)。

$$\begin{cases} S_k = S_k + 1 & i \in N_1 \\ S_k = S_k & \text{esle} \end{cases} \quad (2.15)$$

若  $s_k$  满足公式 (2.16) 和公式 (2.17)，则相应的增加或减少粒子资源，

$$\begin{cases} N_1 = N_1 + \Delta N \\ N_2 = N_2 - \Delta N \end{cases} \quad \text{if } S_k \geq S_t \quad (2.16)$$

$$\begin{cases} N_1 = N_1 - \Delta N \\ N_2 = N_2 + \Delta N \end{cases} \quad \text{if } S_k \leq S_l \quad (2.17)$$

其中， $s_t$  为种群  $N_1$  获取奖励的阈值， $s_l$  为种群  $N_2$  获取奖励的阈值。在调整种群数量的同时，需保持总粒子数恒定。

$$N = N_1 + N_2 \quad (2.18)$$

在算法的不同阶段，通过动态调整种群间资源的分配策略，以实现探索与利用的平衡。当种群的竞争成功率较高时，系统会分配更多的资源，以增强其局部搜索和精细化挖掘能力。相反，当某一种群在较长时间内未能找到全局最优解时，则会减少该种群的资源，并将其转移至其他种群，从而增强整体搜索过程的全局探索能力。通过这种资源动态调整机制，算法能够更有效地在全局搜索与局部开发之间取得平衡，提升优化效率。

### 2.3.2 IPSOCC 算法实现

IPSOCC 算法实现流程如图 2-5 所示：

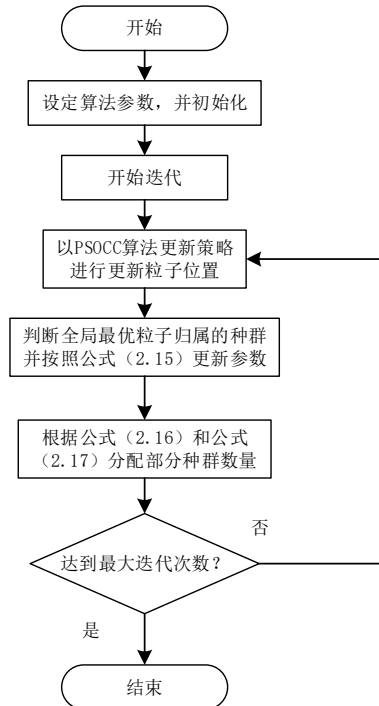


图 2-5 IPSOCC 流程图

### 2.3.3 IPSOCC 算法参数分析

所提出的 IPSOCC 新引入了增强探索与开采平衡机制, 因此, 需对该机制内的参数进行讨论 (即  $s_t$  和  $s_l$ ), 其他参数取值为 PSOCC 研究所确立的数值。由于,  $s_t$  和  $s_l$  分别是种群  $N_1$  和种群  $N_2$  获取奖励的阈值。种群  $N_1$  偏向于局部挖掘因此  $s_t$  的设置决定了种群  $N_1$  何时能够获得更多资源。当种群  $N_1$  在一定时间内获得的全局最优解次数超过  $s_t$  时, 种群  $N_1$  将获得奖励, 意味着系统会增加其资源, 增强局部搜索能力。较低的  $s_t$  值可能导致种群  $N_1$  在更短时间内就能获得更多资源, 从而加速局部挖掘, 但也有可能增加陷入局部最优解的风险。相反, 较高的  $s_t$  值则要求种群  $N_1$  在获得更多最优解之前必须展示出更强的表现, 这有助于避免过度依赖局部搜索。种群  $N_2$  偏向于空间探索, 因此  $s_l$  的设置决定了种群  $N_2$  何时能够获得更多资源。如果种群  $N_2$  能够快速找到全局最优解或在搜索过程中表现出较高的多样性, 满足  $s_l$  的条件时, 系统会向其分配更多资源以增强全局探索能力。较低的  $s_l$  值会使种群  $N_2$  容易过早得到奖励, 可能导致资源分配不均衡, 影响全局搜索效率; 而较高的  $s_l$  值则使得种群  $N_2$  需要更长的时间获得资源, 从而延迟了全局搜索, 影响了算法的收敛速度。

因此,  $s_t$  和  $s_l$  的设定直接影响种群间的资源分配和优化过程。适当的  $s_t$  和  $s_l$ , 可以有效平衡局部搜索和全局搜索之间的关系, 避免算法陷入局部最优解, 提升算法的全局优化能力和收敛效率。 $s_t$  和  $s_l$  的取值如表 2-3。在求解空间维度为 30 和 50 的情况下通过 CEC2005 测试集中的 F1 与 F10 作为基准函数, 在不同的参数组合下算法性能结果如表 2-4。

类似的, 对每个基准函数在固定维度下的数据进行排名, 对排名前十的数据加粗。观察比较各参数组合的实验结果, 如表 2-4, 当  $s_t$  和  $s_l$  分别取为 90 和 25 时, IPSOCC 算法能够在单峰函数和多峰函数中的不同维度下表现出最佳的优化性能, 最大程度体现了 IPSOCC 算法在处理不同问题下的鲁棒性和求解精度以及较好的稳定性。因此, 通过参数学习将  $s_t$  和  $s_l$  设为 90 和 25, 作为后续实验的参数值。

表 2-3 IPSOCC 分析参数  $s_t$  和  $s_l$  的取值情况

| 参数    | 取值情况 |      |      |      |      |      |      |
|-------|------|------|------|------|------|------|------|
|       | 取值 1 | 取值 2 | 取值 3 | 取值 4 | 取值 5 | 取值 6 | 取值 7 |
| $s_t$ | 60   | 65   | 70   | 75   | 80   | 85   | 90   |
| $s_l$ | 10   | 15   | 20   | 25   | 30   | 35   | 40   |

表 2-4 IPSOCC 参数正交表

| 序号 | 参数    |       | F1              | F10             | F1              | F10             |
|----|-------|-------|-----------------|-----------------|-----------------|-----------------|
|    | $S_t$ | $S_l$ | D=30            |                 | D=50            |                 |
| 1  | 60    | 10    | 7.33E-09        | 1.30E-01        | 1.09E-05        | 1.23E+00        |
| 2  | 60    | 15    | 1.05E-08        | <b>2.93E-05</b> | 2.11E-05        | 8.46E-01        |
| 3  | 60    | 20    | <b>5.81E-09</b> | <b>6.20E-05</b> | 9.04E-06        | 1.16E+00        |
| 4  | 60    | 25    | <b>6.23E-09</b> | 9.57E-02        | 1.07E-05        | 7.70E-01        |
| 5  | 60    | 30    | 8.30E-09        | <b>5.67E-05</b> | 1.09E-05        | 8.65E-01        |
| 6  | 60    | 35    | 1.77E-08        | 1.03E-01        | 1.24E-05        | 1.19E+00        |
| 7  | 60    | 40    | 6.81E-09        | 9.13E-02        | 1.21E-05        | 8.51E-01        |
| 8  | 65    | 10    | 9.02E-09        | 9.36E-02        | <b>7.77E-06</b> | <b>5.11E-01</b> |
| 9  | 65    | 15    | 7.13E-09        | 4.02E-01        | <b>6.70E-06</b> | 7.82E-01        |
| 10 | 65    | 20    | 6.32E-09        | 1.55E-01        | 7.83E-06        | 9.52E-01        |
| 11 | 65    | 25    | 1.12E-08        | 1.82E-01        | 1.22E-05        | 1.26E+00        |
| 12 | 65    | 30    | 9.21E-09        | 9.02E-02        | 8.91E-06        | <b>5.72E-01</b> |
| 13 | 65    | 35    | 9.26E-09        | 9.84E-02        | 8.93E-06        | 1.14E+00        |
| 14 | 65    | 40    | 6.43E-09        | 8.35E-02        | 1.07E-05        | 8.82E-01        |
| 15 | 70    | 10    | 1.44E-08        | 1.84E-01        | 1.32E-05        | 1.28E+00        |
| 16 | 70    | 15    | 2.17E-08        | 8.28E-02        | 1.23E-05        | 7.20E-01        |
| 17 | 70    | 20    | 8.03E-09        | <b>3.68E-05</b> | 9.16E-06        | 7.13E-01        |
| 18 | 70    | 25    | 7.80E-09        | 1.12E-03        | 1.20E-05        | <b>5.62E-01</b> |
| 19 | 70    | 30    | <b>5.79E-09</b> | 8.59E-02        | 8.58E-06        | 5.74E-01        |
| 20 | 70    | 35    | 1.02E-08        | 1.05E-01        | 1.28E-05        | 9.29E-01        |
| 21 | 70    | 40    | 1.01E-08        | 1.73E-01        | <b>5.51E-06</b> | 1.21E+00        |
| 22 | 75    | 10    | 7.44E-09        | 1.82E-01        | 1.13E-05        | <b>5.14E-01</b> |
| 23 | 75    | 15    | 1.57E-08        | <b>2.05E-04</b> | 1.16E-05        | 6.63E-01        |
| 24 | 75    | 20    | 8.66E-09        | <b>1.43E-04</b> | 2.03E-05        | 9.32E-01        |
| 25 | 75    | 25    | 8.13E-09        | 2.01E-01        | <b>6.78E-06</b> | 1.03E+00        |
| 26 | 75    | 30    | 1.55E-08        | 2.04E-01        | 1.04E-05        | <b>4.45E-01</b> |
| 27 | 75    | 35    | 8.83E-09        | 2.77E-01        | 1.12E-05        | <b>4.46E-01</b> |
| 28 | 75    | 40    | 8.47E-09        | 1.81E-01        | 1.24E-05        | <b>4.07E-01</b> |
| 29 | 80    | 10    | 1.25E-08        | 2.88E-01        | 1.71E-05        | 8.41E-01        |
| 30 | 80    | 15    | 1.08E-08        | 1.09E-01        | 1.43E-05        | 1.04E+00        |
| 31 | 80    | 20    | 7.25E-09        | 2.52E-01        | 1.08E-05        | 7.21E-01        |
| 32 | 80    | 25    | 6.94E-09        | 1.24E-01        | 8.66E-06        | 1.09E+00        |
| 33 | 80    | 30    | 6.43E-09        | 2.06E-01        | <b>6.20E-06</b> | 6.67E-01        |
| 34 | 80    | 35    | 7.45E-09        | 9.84E-02        | 1.80E-05        | 1.13E+00        |
| 35 | 80    | 40    | <b>5.09E-09</b> | 8.65E-02        | 1.39E-05        | 8.28E-01        |
| 36 | 85    | 10    | <b>5.43E-09</b> | 9.94E-02        | 1.09E-05        | <b>4.95E-01</b> |
| 37 | 85    | 15    | 1.13E-08        | 1.01E-01        | 1.19E-05        | <b>4.91E-01</b> |

续表：2-4

|    |    |    |                 |                 |                 |                 |
|----|----|----|-----------------|-----------------|-----------------|-----------------|
| 38 | 85 | 20 | 1.09E-08        | 9.75E-02        | 1.31E-05        | 6.91E-01        |
| 39 | 85 | 25 | 7.23E-09        | <b>1.18E-04</b> | <b>7.50E-06</b> | 1.32E+00        |
| 40 | 85 | 30 | 8.24E-09        | <b>4.07E-05</b> | 1.10E-05        | <b>5.72E-01</b> |
| 41 | 85 | 35 | 6.69E-09        | 3.13E-01        | <b>7.31E-06</b> | 8.72E-01        |
| 42 | 85 | 40 | <b>5.89E-09</b> | 1.17E-01        | <b>6.63E-06</b> | 6.61E-01        |
| 43 | 90 | 10 | <b>5.79E-09</b> | 1.09E-01        | 1.03E-05        | 9.43E-01        |
| 44 | 90 | 15 | 8.11E-09        | 1.10E-01        | 1.00E-05        | 5.91E-01        |
| 45 | 90 | 20 | <b>4.71E-09</b> | 9.26E-02        | 1.30E-05        | 8.18E-01        |
| 46 | 90 | 25 | <b>6.15E-09</b> | <b>6.00E-05</b> | <b>7.73E-06</b> | 8.23E-01        |
| 47 | 90 | 30 | 7.67E-09        | <b>8.96E-05</b> | <b>7.41E-06</b> | 7.44E-01        |
| 48 | 90 | 35 | <b>5.63E-09</b> | 8.64E-02        | 8.56E-06        | 1.04E+00        |
| 49 | 90 | 40 | 6.55E-09        | 1.80E-01        | 2.96E-05        | 7.04E-01        |

## 2.4 仿真实验与结果分析

本节通过仿真研究和评估，对增强优化算法的效率进行了深入分析。为了比较 PSOCC、IPSOCC 和对比算法的性能，采用了 CEC2005 测试集进行算法性能实验。表 2-5 展示了测试集中各基准函数的特性，23 个测试函数均具有独特的特征及其属性，表明 CEC2005 基准函数可以复制具有复杂场景的现实世界问题，能够有效地评估算法在各种复杂条件下的性能。

表 2-5 CEC2005 测试函数参数信息

| 函数名           | 测试函数                                                               | 范围           | 维度 | 最小值 | 类型       |
|---------------|--------------------------------------------------------------------|--------------|----|-----|----------|
| Sphere        | $F_1(x) = \sum_{i=1}^N x_i^2$                                      | [-100,100]   | 30 | 0   |          |
| Schwefel 2.22 | $F_2(x) = \sum_{i=1}^N  x_i  + \prod_{i=1}^N  x_i $                | [-10,10]     | 30 | 0   |          |
| Schwefel 1.2  | $F_3(x) = \sum_{i=1}^N (\sum_{j=1}^i x_j^2)$                       | [-100,100]   | 30 | 0   |          |
| Schwefel 2.21 | $F_4(x) = \max\{ x_i , 1 \leq i \leq D\}$                          | [-100,100]   | 30 | 0   | Unimodal |
| Rosenbrock    | $F_5(x) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | [-30,30]     | 30 | 0   |          |
| Step          | $F_6(x) = \sum_{i=1}^N ([x_i + 0.5])^2$                            | [-100,100]   | 30 | 0   |          |
| Quartic       | $F_7(x) = \sum_{i=1}^N ix_i^4 + \text{random}[0,1)$                | [-1.28,1.28] | 30 | 0   |          |

续表: 2-5

|                        |                                                                                                                                                                                        |                     |    |                  |            |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|----|------------------|------------|
| Schwefel               | $F_8(x) = -\sum_{i=1}^N (x_i \sin \sqrt{ x_i })$                                                                                                                                       | [-500,500]          | 30 | -12569.5         |            |
| Rastrigin              | $F_9(x) = 10N + \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i))$                                                                                                                              | [-5.12,5.12]        | 30 | 0                |            |
| Ackley                 | $F_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2})$<br>$-\exp(\frac{1}{N} \sum_{i=1}^N \cos(2\pi x_i)) + 20 + e$                                                        | [-32,32]            | 30 | 0                |            |
| Griewank               | $F_{11}(x) = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos(\frac{x_i}{\sqrt{i}}) + 1$                                                                                         | [-600,600]          | 30 | 0                |            |
|                        | $F_{12}(x) = \frac{\pi}{N} \{10 \sin(\pi y_1) + \sum_{i=1}^{N-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_N - 1)^2\} + \sum_{i=1}^N u(x_i, 10, 100, 4)$                           |                     |    |                  | Multimodal |
| Penalized              | $where, y_i = 1 + \frac{x_i + 1}{4}, and, u(x_i, a, k, m)$<br>$= \begin{cases} k(x_i - a)^m; x_i > a \\ 0; -a < x_i < a \\ k(-x_i - a)^m; x_i < -a \end{cases}$                        | [-50,50]            | 30 | 0                |            |
| Penalize 2             | $F_{13}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^N (x_i - 1)^2 [1 + \sin^2(3\pi x_i) + 1] + \sum_{i=1}^N (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \}$<br>$+ \sum_{i=1}^N u(x_i, 5, 100, 4)$ | [-50,50]            | 30 | 0                |            |
| Foxholes               | $F_{14}(x) = [\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^N (x_i - a_{ij})^6}]^{-1}$                                                                                       | $[-65.536, 65.536]$ | 2  | 0.998003<br>8377 |            |
| Kowalik                | $F_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$                                                                                             | [-5,5]              | 4  | 0.000307<br>5    |            |
| Six-hump<br>Camel-Back | $F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$                                                                                                          | [-5,5]              | 2  | -1.03162         |            |
| Branin                 | $F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$                                                                           | [-5,5]              | 2  | 0.397887<br>3577 | Fixed      |
| Goldstein<br>-Price    | $F_{18}(x) = [1 + (x_1 + x_2 + 1)(10 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2^2)(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$                   | [-5,5]              | 2  | 3                |            |
| Hartman 3              | $F_{19}(x) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2]$                                                                                                             | [-5,5]              | 3  | -3.86278         |            |

续表：2-5

|           |                                                                            |        |   |          |
|-----------|----------------------------------------------------------------------------|--------|---|----------|
| Hartman 6 | $F_{20}(x) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2]$ | [-5,5] | 6 | -3.32236 |
| Shekel5   | $F_{21}(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$              | [-5,5] | 4 | -10.1532 |
| Shekel7   | $F_{22}(x) = -\sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$              | [-5,5] | 4 | -10.4029 |
| Shekel10  | $F_{23}(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$           | [-5,5] | 4 | -10.5364 |

Fixed

为了客观验证算法 PSOCC 算法与 IPSOCC 算法性能，本节将 PSOCC 算法和 IPSOCC 算法与 PSO、乌鸦搜索算法（Crow Search Algorithm, CSA）<sup>[42]</sup>、正弦余弦算法（Sine Cosine Algorithm, SCA）<sup>[43]</sup>和自然选择粒子群优化算法（Natural Selection Particle Swarm Optimization Algorithm, SELPSO）<sup>[44]</sup>进行实验比较，各算法种群数量均为 30，最大迭代次数为 1000 次。表 2-6 概述了各算法用于对比时的参数配置。

表 2-6 各算法参数信息

| 算法                     | 参数                                                                                           |
|------------------------|----------------------------------------------------------------------------------------------|
| PSOCC                  | $c_1 = c_2 = 2, w = 0.8, T_1 = 50, T_2 = 100, L = 9, K = 0.4, T_K = 500$                     |
| IPSOCC                 | $c_1 = c_2 = 2, w = 0.8, T_1 = 50, T_2 = 100, L = 9, K = 0.4, T_K = 500, S_i = 90, S_l = 25$ |
| PSO <sup>[3]</sup>     | $w = 0.8, c_1 = c_2 = 2, V_{\max} = 4$                                                       |
| SelPSO <sup>[44]</sup> | $w = 0.8, c_1 = c_2 = 2, V_{\max} = 4$                                                       |
| CSA <sup>[42]</sup>    | $Awareness\ probability(AP) = 0.1, Flight\ length = 2$                                       |
| SCA <sup>[43]</sup>    | $r_2 = 2\pi * rand, r_3 = 2 * rand, r_4 = rand, a = 2$                                       |

#### 2.4.1 低维函数测试结果

本节将各算法在 30 空间维度每个基准函数上独立运行 50 次以计算平均值（Ave）、中值（Med）、方差（SD）和最差结果（Worst）进行比较，通过综合指标，可以全面分析各个算法在处理 CEC2005 基准函数时的优劣性，各算法的测试结果如表 2-7。

通过表 2-7 中的实验结果能够看出，IPSOCC、PSOCC 和 SCA、PSO、CSA 和 SelPSO 对比算法对比时在 23 个测试函数中的 16 个函数中都取得了最好数值结果，并且，通过 Rank 结果可知，IPSOCC 和 PSOCC 均优于对比算法，并且 IPSOCC 的 Rank 排名为第一。因此，对比算法相比，所提出的算法在单峰函数、多峰函数以及复合函数的优化中均表现出优越的性能。并且 IPSOCC、PSOCC 在解的精度以及全局搜索能力方面相较于 PSO 算法均取得了显著的提升。实验结

果表明, IPSOCC、PSOCC 不仅能够有效避免局部最优解的困扰, 还能够复杂的优化问题中保持较好的稳定性和鲁棒性。

表 2-7 30 维空间维度中各算法收敛精度统计

| 函数  | 指标    | SCA             | PSO             | CSA             | SelPSO          | PSOCC           | IPSOCC          |
|-----|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| F1  | Ave   | 1.85E-03        | 1.17E-01        | 7.20E-02        | 3.72E-01        | 3.66E-06        | <b>2.70E-08</b> |
|     | SD    | 7.28E-06        | 2.27E-03        | 1.10E-03        | 8.23E-01        | 1.47E-11        | <b>1.31E-15</b> |
|     | Med   | 8.22E-04        | 1.03E-01        | 8.44E-02        | 5.08E-01        | 2.22E-06        | <b>1.93E-08</b> |
|     | Worst | 7.20E-03        | 1.90E-01        | 1.17E-01        | 1.37E+00        | 1.60E-05        | <b>1.86E-07</b> |
|     | Rank  | 6               | 3               | 5               | 4               | 2               | 1               |
| F2  | Ave   | 4.01E-05        | 5.29E+00        | 2.09E+00        | 7.33E+00        | 3.92E-04        | <b>3.27E-05</b> |
|     | SD    | <b>2.85E-09</b> | 5.69E+01        | 1.12E+00        | 6.31E+00        | 9.84E-08        | 3.23E-09        |
|     | Med   | <b>1.50E-05</b> | 2.18E+00        | 1.64E+00        | 1.11E+01        | 2.82E-04        | 2.50E-05        |
|     | Worst | <b>1.46E-04</b> | 2.21E+01        | 4.34E+00        | 1.14E+01        | 1.52E-03        | 2.38E-04        |
|     | Rank  | 2               | 5               | 4               | 6               | 3               | 1               |
| F3  | Ave   | 4.46E+03        | 2.55E+01        | 9.82E+01        | 5.26E+01        | 3.00E+00        | <b>1.33E+00</b> |
|     | SD    | 1.05E+07        | 4.28E+01        | 4.31E+02        | 4.66E+01        | 7.61E+00        | <b>6.22E+00</b> |
|     | Med   | 3.56E+03        | 2.94E+01        | 8.85E+01        | 3.94E+01        | <b>2.21E+00</b> | 5.01E-01        |
|     | Worst | 9.41E+03        | 3.25E+01        | 1.23E+02        | 7.96E+01        | 1.25E+01        | <b>1.03E+01</b> |
|     | Rank  | 6               | 3               | 5               | 4               | 2               | 1               |
| F4  | Ave   | 1.96E+01        | 2.89E+00        | 4.08E+00        | 2.14E+00        | 1.34E+00        | <b>4.40E-01</b> |
|     | SD    | 5.38E+01        | 3.82E+00        | 3.57E-01        | 1.74E+00        | 1.43E+00        | <b>2.47E-01</b> |
|     | Med   | 1.90E+01        | 1.63E+00        | 4.37E+00        | 1.88E+00        | 9.34E-01        | <b>2.47E-01</b> |
|     | Worst | 3.02E+01        | 6.75E+00        | 4.65E+00        | 3.76E+00        | 5.77E+00        | <b>2.28E+00</b> |
|     | Rank  | 3               | 5               | 6               | 4               | 2               | 1               |
| F5  | Ave   | 3.15E+02        | 9.63E+01        | 8.39E+01        | 1.64E+02        | 5.15E+01        | <b>4.41E+01</b> |
|     | SD    | 1.81E+05        | 2.00E+03        | 5.49E+03        | <b>1.54E+02</b> | 1.62E+03        | 2.02E+03        |
|     | Med   | 6.00E+01        | 8.73E+01        | 5.43E+01        | 1.64E+02        | 2.88E+01        | <b>2.71E+01</b> |
|     | Worst | 1.14E+03        | 1.95E+02        | 2.92E+02        | 3.25E+02        | <b>1.60E+02</b> | 1.90E+02        |
|     | Rank  | 6               | 4               | 3               | 5               | 2               | 1               |
| F6  | Ave   | 4.69E+00        | 2.22E-01        | 9.76E-02        | 1.85E-01        | <b>4.30E-02</b> | 1.28E-01        |
|     | SD    | 3.72E-01        | 6.45E-03        | 4.49E-03        | 2.07E-01        | <b>2.48E-03</b> | 1.90E-02        |
|     | Med   | 4.35E+00        | 2.16E-01        | 6.89E-02        | 2.00E-01        | <b>2.36E-02</b> | 9.26E-02        |
|     | Worst | 5.71E+00        | 3.36E-01        | <b>2.25E-01</b> | 2.35E-01        | 2.88E-01        | 5.19E-01        |
|     | Rank  | 6               | 5               | 2               | 4               | 1               | 3               |
| F7  | Ave   | 4.09E-02        | 5.63E-02        | 1.63E-02        | 2.22E-02        | 6.14E-03        | <b>6.06E-03</b> |
|     | SD    | 8.52E-05        | 5.87E-03        | 1.11E-04        | 2.26E-02        | 2.04E-05        | <b>2.02E-05</b> |
|     | Med   | 4.52E-02        | 1.50E-02        | 3.57E-02        | 2.06E-02        | <b>5.08E-03</b> | 7.14E-03        |
|     | Worst | 4.75E-02        | 2.09E-01        | 4.58E-02        | 3.12E-02        | 2.32E-02        | <b>1.60E-02</b> |
|     | Rank  | 5               | 6               | 3               | 4               | 2               | 1               |
| F8  | Ave   | -4134.72        | -6765.30        | -6444.37        | -6964.25        | <b>-8451.92</b> | -8249.82        |
|     | SD    | 42094.78        | 878015.10       | 940489.80       | <b>7157.56</b>  | 965674.35       | 549554.55       |
|     | Med   | -4089.72        | -6485.83        | -6358.08        | -6744.45        | <b>-8466.41</b> | -8124.04        |
|     | Worst | -3907.86        | -5170.48        | -5018.32        | -6191.01        | -6613.85        | <b>-6970.46</b> |
|     | Rank  | 6               | 4               | 5               | 3               | 1               | 2               |
| F9  | Ave   | <b>3.00E+01</b> | 1.04E+02        | 3.35E+01        | 9.72E+01        | 6.59E+01        | 7.96E+01        |
|     | SD    | 6.81E+02        | 7.98E+02        | <b>3.74E+01</b> | 1.03E+02        | 4.80E+02        | 1.02E+03        |
|     | Med   | <b>2.29E+01</b> | 1.08E+02        | 3.38E+01        | 1.04E+02        | 6.15E+01        | 7.38E+01        |
|     | Worst | 7.30E+01        | 1.57E+02        | <b>4.08E+01</b> | 1.19E+02        | 1.19E+02        | 1.95E+02        |
|     | Rank  | 1               | 6               | 2               | 5               | 3               | 4               |
| F10 | Ave   | 1.62E+01        | 2.58E+00        | 3.25E+00        | 3.02E+00        | 7.05E-01        | <b>7.15E-04</b> |
|     | SD    | 6.42E+01        | 2.11E-01        | 6.47E-01        | 2.79E+00        | 1.18E+00        | <b>6.38E-06</b> |
|     | Med   | 2.02E+01        | 2.86E+00        | 3.13E+00        | 3.14E+00        | 2.49E-03        | <b>6.42E-05</b> |
|     | Worst | 2.03E+01        | 3.06E+00        | 4.82E+00        | 3.93E+00        | 2.90E+00        | <b>1.23E-02</b> |
|     | Rank  | 6               | 3               | 5               | 4               | 2               | 1               |
| F11 | Ave   | 2.74E-01        | 2.03E-02        | 2.63E-01        | 2.79E-02        | 2.67E-02        | <b>1.66E-02</b> |
|     | SD    | 6.17E-02        | <b>4.91E-05</b> | 9.47E-03        | 2.96E-02        | 1.51E-03        | 3.26E-04        |
|     | Med   | 2.36E-01        | 2.22E-02        | 2.37E-01        | 2.10E-02        | <b>1.62E-02</b> | 1.65E-02        |
|     | Worst | 7.41E-01        | <b>2.78E-02</b> | 3.87E-01        | 6.99E-02        | 1.52E-01        | 6.65E-02        |
|     | Rank  | 4               | 2               | 6               | 5               | 3               | 1               |
| F12 | Ave   | <b>8.51E-01</b> | 5.00E+00        | 4.56E+00        | 6.50E+00        | 5.74E+00        | 4.45E+00        |
|     | SD    | <b>5.22E-02</b> | 6.25E+00        | 1.90E+00        | 7.17E+00        | 7.72E+00        | 9.78E+00        |
|     | Med   | <b>9.98E-01</b> | 4.69E+00        | 4.99E+00        | 4.79E+00        | 5.52E+00        | 4.39E+00        |
|     | Worst | <b>1.08E+00</b> | 1.07E+01        | 6.25E+00        | 1.13E+01        | 1.29E+01        | 1.33E+01        |
|     | Rank  | 1               | 4               | 3               | 6               | 5               | 2               |
| F13 | Ave   | 6.04E+00        | 3.68E+00        | 7.03E-01        | 4.74E-01        | <b>3.84E-02</b> | 4.36E-02        |

续表：2-7

|      |         |                 |                  |                  |                  |                  |                  |
|------|---------|-----------------|------------------|------------------|------------------|------------------|------------------|
|      | SD      | 2.14E+01        | 4.18E+01         | 4.22E-01         | 4.76E-01         | <b>1.61E-03</b>  | 1.82E-03         |
|      | Med     | 5.07E+00        | 5.58E-01         | 6.39E-01         | 4.69E-01         | <b>2.43E-02</b>  | 2.94E-02         |
|      | Worst   | 1.50E+01        | 1.66E+01         | 1.91E+00         | 6.75E-01         | <b>1.91E-01</b>  | 2.09E-01         |
|      | Rank    | 6               | 5                | 4                | 3                | 1                | 2                |
|      | Ave     | 9.99E-01        | 1.89E+00         | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  |
| F14  | SD      | 3.75E-07        | 2.23E+00         | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         |
|      | Med     | 9.99E-01        | 9.98E-01         | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  |
|      | Worst   | 1.00E+00        | 5.93E+00         | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  |
|      | Rank    | 5               | 6                | 1                | 1                | 1                | 1                |
|      | Ave     | 1.15E-03        | 7.91E-03         | <b>6.74E-04</b>  | 1.51E-02         | 1.26E-03         | 6.77E-03         |
| F15  | SD      | 9.67E-08        | 9.25E-05         | <b>2.01E-07</b>  | 1.68E-02         | 1.56E-05         | 9.06E-05         |
|      | Med     | 1.30E-03        | 1.74E-03         | <b>3.07E-04</b>  | 2.04E-02         | 3.07E-04         | 3.19E-04         |
|      | Worst   | 1.48E-03        | 2.26E-02         | <b>1.22E-03</b>  | 2.26E-02         | 2.04E-02         | 2.04E-02         |
|      | Rank    | 2               | 5                | 1                | 6                | 3                | 4                |
|      | Ave     | <b>-1.0316</b>  | <b>-1.0316</b>   | <b>-1.0316</b>   | <b>-1.0316</b>   | <b>-1.0316</b>   | <b>-1.0316</b>   |
| F16  | SD      | <b>0</b>        | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         |
|      | Med     | <b>-1.0316</b>  | <b>-1.0316</b>   | <b>-1.0316</b>   | <b>-1.0316</b>   | <b>-1.0316</b>   | <b>-1.0316</b>   |
|      | Worst   | <b>-1.0316</b>  | <b>-1.0316</b>   | <b>-1.0316</b>   | <b>-1.0316</b>   | <b>-1.0316</b>   | <b>-1.0316</b>   |
|      | Rank    | 1               | 1                | 1                | 1                | 1                | 1                |
|      | Ave     | 3.98E-01        | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  |
| F17  | SD      | 1.13E-07        | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         |
|      | Med     | 3.99E-01        | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  |
|      | Worst   | 3.99E-01        | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  |
|      | Rank    | 6               | 1                | 1                | 1                | 1                | 1                |
|      | Ave     | <b>3.00E+00</b> | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  |
| F18  | SD      | <b>0</b>        | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         |
|      | Med     | <b>3.00E+00</b> | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  |
|      | Worst   | <b>3.00E+00</b> | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  |
|      | Rank    | 1               | 1                | 1                | 1                | 1                | 1                |
|      | Ave     | -3.85E+00       | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> |
| F19  | SD      | 2.13E-07        | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         |
|      | Med     | -3.85E+00       | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> |
|      | Worst   | -3.85E+00       | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> |
|      | Rank    | 6               | 1                | 1                | 1                | 1                | 1                |
|      | Ave     | -3.03E+00       | -3.12E+00        | <b>-3.32E+00</b> | -2.70E+00        | -3.27E+00        | -3.27E+00        |
| F20  | SD      | 1.51E-03        | 1.79E-02         | <b>0</b>         | -2.59E+00        | 3.78E-03         | 2.15E-03         |
|      | Med     | -3.01E+00       | -3.13E+00        | <b>-3.32E+00</b> | -2.81E+00        | -3.32E+00        | -3.32E+00        |
|      | Worst   | -3.01E+00       | -2.84E+00        | <b>-3.32E+00</b> | -1.71E+00        | -3.14E+00        | -3.14E+00        |
|      | Rank    | 5               | 4                | 1                | 6                | 3                | 2                |
|      | Ave     | -6.98E-01       | -9.42E+00        | <b>-1.02E+01</b> | -7.63E+00        | -9.95E+00        | -9.55E+00        |
| F21  | SD      | 5.19E-02        | 3.18E+00         | <b>0</b>         | -8.27E+00        | 1.00E+00         | 2.82E+00         |
|      | Med     | -8.79E-01       | -1.02E+01        | <b>-1.02E+01</b> | -1.02E+01        | -1.02E+01        | -1.02E+01        |
|      | Worst   | -3.51E-01       | -5.06E+00        | <b>-1.02E+01</b> | -2.63E+00        | -5.10E+00        | -5.06E+00        |
|      | Rank    | 6               | 4                | 1                | 5                | 2                | 3                |
|      | Ave     | -4.03E+00       | -9.45E+00        | <b>-1.04E+01</b> | -8.88E+00        | -9.87E+00        | -9.67E+00        |
| F22  | SD      | 2.46E+00        | 6.38E+00         | <b>0</b>         | -8.49E+00        | 2.58E+00         | 4.21E+00         |
|      | Med     | -4.66E+00       | -1.04E+01        | <b>-1.04E+01</b> | -1.04E+01        | -1.04E+01        | -1.04E+01        |
|      | Worst   | -9.08E-01       | -2.77E+00        | <b>-1.04E+01</b> | -2.77E+00        | -5.09E+00        | -2.77E+00        |
|      | Rank    | 6               | 4                | 1                | 5                | 2                | 3                |
|      | Ave     | -5.99E+00       | <b>-1.05E+01</b> | <b>-1.05E+01</b> | -8.91E+00        | -1.02E+01        | -9.89E+00        |
| F23  | SD      | 1.61E+00        | <b>0</b>         | <b>0</b>         | -1.05E+01        | 1.67E+00         | 3.22E+00         |
|      | Med     | -5.41E+00       | <b>-1.05E+01</b> | <b>-1.05E+01</b> | -1.05E+01        | -1.05E+01        | -1.05E+01        |
|      | Worst   | -4.99E+00       | <b>-1.05E+01</b> | <b>-1.05E+01</b> | -2.43E+00        | -5.13E+00        | -5.13E+00        |
|      | Rank    | 6               | 1                | 1                | 5                | 3                | 4                |
|      | AvgRank | 4.4348          | 3.6087           | 2.7391           | 3.8696           | 2.0435           | 1.8261           |
| Rank |         | 6               | 4                | 3                | 5                | 2                | 1                |

#### 2.4.2 高维函数测试结果

保持种群大小设为 30，在 CEC2005 测试集的 13 个测试函数 F1 至 F13 上求解 50 的空间维度时，比较了 PSOCC、IPSOCC 与对比算法的性能差异。通过提升解的空间维度，从而全面地评估算法的性能、稳定性和鲁棒性。评估算法在不

同维度上的性能，可以全面评估其在不同场景中的通用性和优势。表 2-8 显示了维度 50 处比较算法的每种算法解的结果。

根据表 2-8 中的实验结果可知，PSOCC、IPSOCC 在求解高维度函数空间的 13 函数测试集里的 12 个函数表现均优于对比算法，根据最终 Rank 排名可知，改进算法的总成绩均要优于其他对比算法。具体而言，PSOCC、IPSOCC 在处理高维问题时展现了更高的解精度，能够有效应对高维搜索空间中的复杂性和维度灾难。与对比算法相比，所提出的算法在全局搜索能力和局部优化能力之间取得了更好的平衡，避免了过早收敛到局部最优解。

表 2-8 50 维空间维度中各算法收敛精度统计

| 函数  | 指标    | SCA       | PSO       | MFO              | SeIPSO           | PSOCC           | IPSOCC           |
|-----|-------|-----------|-----------|------------------|------------------|-----------------|------------------|
| F1  | Ave   | 1.29E+01  | 2.53E+00  | 5.54E+00         | 3.16E+00         | 1.70E-03        | <b>4.71E-05</b>  |
|     | SD    | 8.21E+01  | 1.18E-01  | 2.46E+01         | 3.54E+00         | 3.59E-06        | <b>4.39E-09</b>  |
|     | Med   | 1.94E+01  | 2.55E+00  | 3.56E+00         | 3.22E+00         | 1.24E-03        | <b>1.92E-05</b>  |
|     | Worst | 2.13E+01  | 2.98E+00  | 1.47E+01         | 4.19E+00         | 9.19E-03        | <b>2.96E-04</b>  |
|     | Rank  | 6         | 3         | 5                | 4                | 2               | 1                |
| F2  | Ave   | 1.67E-02  | 2.06E+01  | 5.37E+01         | 1.92E+01         | 1.08E-02        | <b>1.20E-03</b>  |
|     | SD    | 3.53E-04  | 3.98E+01  | 1.68E+03         | 2.11E+01         | 4.70E-05        | <b>6.11E-07</b>  |
|     | Med   | 3.64E-03  | 2.00E+01  | 4.24E+01         | 1.79E+01         | 9.26E-03        | <b>1.09E-03</b>  |
|     | Worst | 4.94E-02  | 2.94E+01  | 1.50E+02         | 3.05E+01         | 3.37E-02        | <b>3.56E-03</b>  |
|     | Rank  | 3         | 5         | 6                | 4                | 2               | 1                |
| F3  | Ave   | 2.47E+04  | 7.22E+02  | 5.47E+04         | 8.09E+02         | <b>3.71E+02</b> | 3.89E+02         |
|     | SD    | 9.55E+07  | 1.26E+05  | 7.92E+08         | 7.35E+02         | <b>8.31E+04</b> | 8.62E+04         |
|     | Med   | 2.52E+04  | 7.10E+02  | 5.08E+04         | 7.79E+02         | 3.47E+02        | <b>3.36E+02</b>  |
|     | Worst | 3.99E+04  | 1.49E+03  | 1.05E+05         | 1.10E+03         | 1.98E+03        | 1.27E+03         |
|     | Rank  | 5         | 3         | 6                | 4                | 1               | 2                |
| F4  | Ave   | 6.04E+01  | 1.26E+01  | 8.74E+01         | 8.96E+00         | 1.16E+01        | <b>6.73E+00</b>  |
|     | SD    | 4.19E+01  | 2.78E+00  | 6.04E+00         | 8.49E+00         | 3.25E+01        | 3.33E+01         |
|     | Med   | 6.09E+01  | 1.28E+01  | 8.70E+01         | 9.27E+00         | 1.24E+01        | <b>7.15E+00</b>  |
|     | Worst | 7.15E+01  | 1.47E+01  | 9.02E+01         | 1.36E+01         | 2.24E+01        | 1.71E+01         |
|     | Rank  | 5         | 4         | 6                | 2                | 3               | 1                |
| F5  | Ave   | 1.06E+05  | 6.86E+02  | 2.29E+04         | 7.08E+02         | 1.46E+02        | <b>1.45E+02</b>  |
|     | SD    | 5.46E+09  | 4.73E+04  | 1.20E+09         | 6.91E+02         | 1.92E+04        | 2.82E+04         |
|     | Med   | 1.14E+05  | 6.80E+02  | 2.43E+03         | 7.50E+02         | 7.42E+01        | <b>4.94E+01</b>  |
|     | Worst | 1.95E+05  | 1.14E+03  | 9.35E+04         | 1.03E+03         | <b>6.36E+02</b> | 6.59E+02         |
|     | Rank  | 6         | 3         | 5                | 4                | 2               | 1                |
| F6  | Ave   | 2.55E+01  | 2.38E+00  | 8.23E+01         | 3.13E+00         | <b>1.81E+00</b> | 2.28E+00         |
|     | SD    | 7.40E+02  | 5.36E-01  | 2.82E+04         | 3.25E+00         | <b>1.86E-01</b> | 5.45E-01         |
|     | Med   | 1.39E+01  | 2.27E+00  | 7.03E+00         | 2.64E+00         | <b>1.75E+00</b> | 2.42E+00         |
|     | Worst | 7.98E+01  | 4.27E+00  | 4.58E+02         | 4.99E+00         | <b>2.74E+00</b> | 4.06E+00         |
|     | Rank  | 5         | 3         | 6                | 4                | 1               | 2                |
| F7  | Ave   | 1.11E+00  | 5.41E+00  | 3.86E+01         | 9.26E-02         | <b>6.87E-03</b> | 1.47E-02         |
|     | SD    | 1.33E+00  | 1.15E+02  | 9.17E+02         | 1.10E-01         | <b>3.09E-05</b> | 1.02E-04         |
|     | Med   | 5.59E-01  | 3.72E-02  | 3.38E+01         | 6.08E-02         | <b>5.28E-03</b> | 1.35E-02         |
|     | Worst | 3.34E+00  | 2.69E+01  | 9.44E+01         | 2.23E-01         | <b>2.38E-02</b> | 3.43E-02         |
|     | Rank  | 4         | 5         | 6                | 3                | 1               | 2                |
| F8  | Ave   | -4.96E+03 | -9.93E+03 | <b>-1.22E+04</b> | -1.28E+04        | -1.29E+04       | -1.27E+04        |
|     | SD    | 7.11E+04  | 1.24E+07  | 1.76E+06         | <b>1.28E+04</b>  | 2.17E+06        | 5.25E+06         |
|     | Med   | -4.90E+03 | -1.11E+04 | -1.29E+04        | -1.31E+04        | -1.30E+04       | <b>-1.24E+04</b> |
|     | Worst | -4.53E+03 | -3.33E-03 | -9.89E+03        | <b>-1.16E+04</b> | -8.27E+03       | -8.65E+03        |
|     | Rank  | 6         | 5         | 1                | 3                | 4               | 2                |
| F9  | Ave   | 1.33E+02  | 1.90E+02  | 3.15E+02         | 2.47E+02         | <b>1.12E+02</b> | 1.22E+02         |
|     | SD    | 3.51E+03  | 1.63E+03  | 2.59E+03         | 2.57E+02         | <b>9.20E+02</b> | 8.93E+02         |
|     | Med   | 7.74E+01  | 1.94E+02  | 3.20E+02         | 2.42E+02         | 1.12E+02        | 1.20E+02         |
|     | Worst | 1.99E+02  | 2.84E+02  | 3.87E+02         | 3.03E+02         | <b>1.94E+02</b> | 1.91E+02         |
|     | Rank  | 3         | 4         | 6                | 5                | 1               | 2                |
| F10 | Ave   | 1.64E+01  | 4.65E+00  | 1.91E+01         | 4.32E+00         | 2.26E+00        | <b>6.89E-01</b>  |
|     | SD    | 6.53E+01  | 1.59E+00  | 8.51E-01         | 4.10E+00         | 1.39E+00        | 1.54E+00         |

续表：2-8

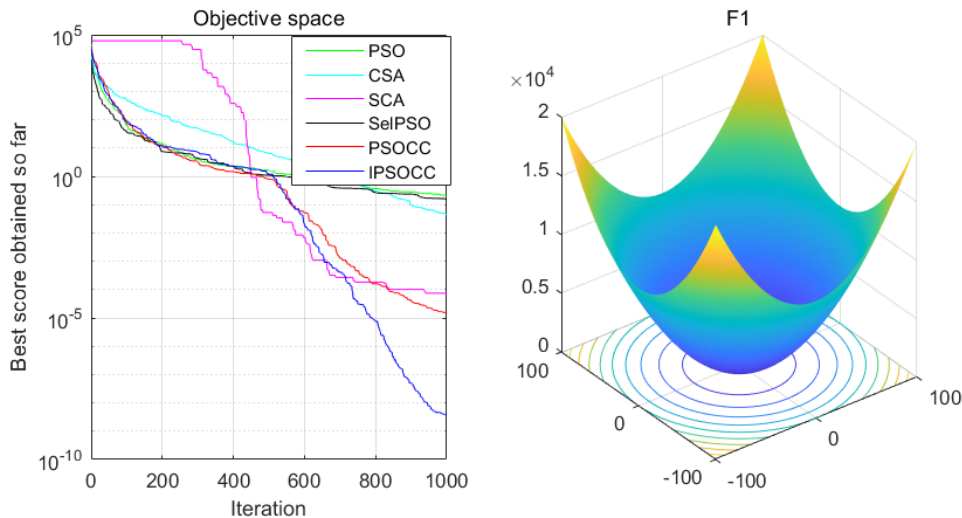
|     |         |          |          |          |          |          |                 |
|-----|---------|----------|----------|----------|----------|----------|-----------------|
|     | Med     | 2.04E+01 | 4.38E+00 | 1.92E+01 | 4.77E+00 | 2.53E+00 | <b>1.95E-03</b> |
|     | Worst   | 2.05E+01 | 7.54E+00 | 1.99E+01 | 5.21E+00 | 4.34E+00 | <b>3.50E+00</b> |
|     | Rank    | 5        | 4        | 6        | 3        | 2        | 1               |
|     | Ave     | 8.47E-01 | 1.58E-01 | 8.86E-01 | 1.87E-01 | 3.80E-02 | <b>1.73E-02</b> |
|     | SD      | 1.26E-01 | 3.28E-03 | 6.67E-02 | 1.93E-01 | 2.88E-03 | <b>8.17E-04</b> |
| F11 | Med     | 1.03E+00 | 1.34E-01 | 1.05E+00 | 1.65E-01 | 2.58E-03 | <b>3.40E-05</b> |
|     | Worst   | 1.09E+00 | 2.57E-01 | 1.10E+00 | 3.36E-01 | 2.57E-01 | <b>9.76E-02</b> |
|     | Rank    | 5        | 3        | 6        | 4        | 2        | 1               |
|     | Ave     | 7.44E+04 | 8.98E+00 | 1.75E+01 | 1.10E+01 | 9.41E+00 | <b>8.76E+00</b> |
|     | SD      | 9.37E+09 | 1.85E+01 | 1.04E+02 | 1.14E+01 | 1.74E+01 | 3.56E+01        |
| F12 | Med     | 1.82E+04 | 8.43E+00 | 2.30E+01 | 1.27E+01 | 8.39E+00 | <b>8.16E+00</b> |
|     | Worst   | 2.51E+05 | 1.73E+01 | 2.98E+01 | 1.43E+01 | 2.12E+01 | 2.96E+01        |
|     | Rank    | 6        | 2        | 5        | 4        | 3        | 1               |
|     | Ave     | 4.59E+05 | 2.41E+01 | 1.21E+02 | 1.81E+01 | 2.36E+00 | <b>9.31E-01</b> |
|     | SD      | 6.75E+11 | 4.57E+02 | 4.35E+03 | 2.11E+01 | 9.00E+01 | <b>3.17E-01</b> |
| F13 | Med     | 2.91E+04 | 1.78E+01 | 1.45E+02 | 6.18E+00 | 9.43E-01 | <b>8.29E-01</b> |
|     | Worst   | 2.27E+06 | 7.40E+01 | 2.17E+02 | 6.43E+01 | 6.80E+01 | <b>2.17E+00</b> |
|     | Rank    | 6        | 4        | 5        | 3        | 2        | 1               |
|     | AvgRank | 5        | 3.6923   | 5.3077   | 3.6154   | 2        | 1.3846          |
|     | Rank    | 5        | 4        | 6        | 3        | 2        | 1               |

### 2.4.3 收敛曲线对比结果

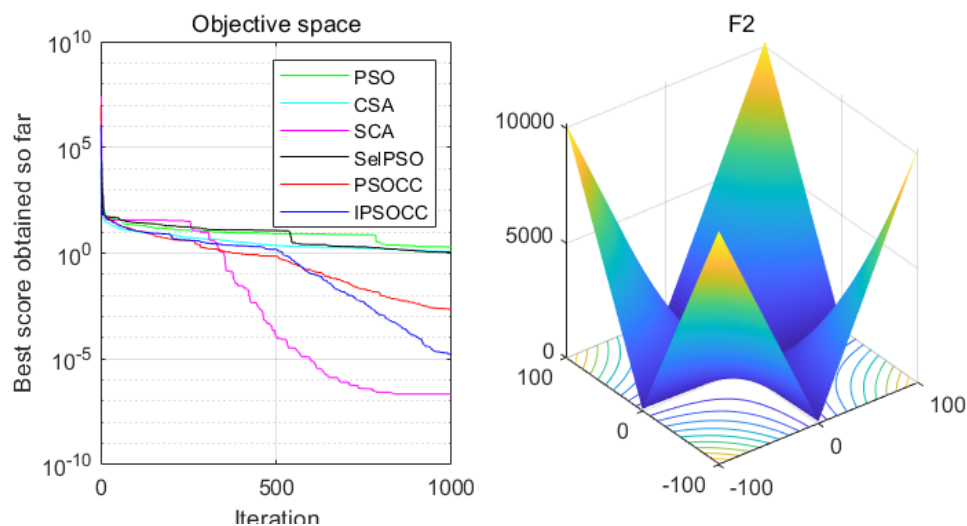
这里将 PSOCC 算法与 IPSOCC 算法和其他经典优化算法进行对比，通过收敛曲线的比较，可以明确地看到不同算法在搜索过程中的收敛趋势，以分析改进算法与其他对比算法的对比结果。由于 CEC2005 包含了一系列广泛的优化问题，能够帮助分析算法在不同问题条件下的表现，因此使用 CEC2005 测试集作为基准函数，能够更加客观地评估算法的性能。

#### (1) 单峰函数

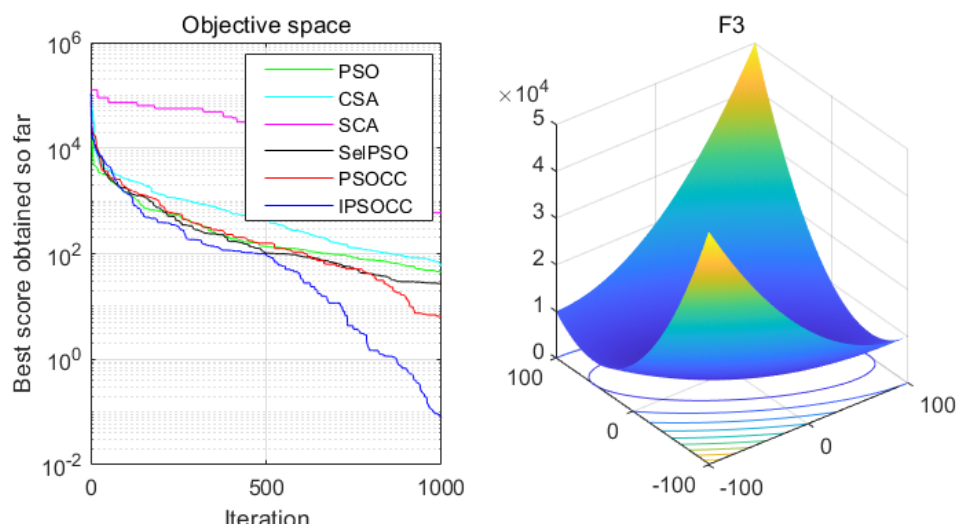
函数（F1 至 F7）为单峰函数，如图 2-6 所示，强调了测试函数的局部精细搜索能力，与较小的值相关的精确度更高。图 2-6 表明与所有对比算法相比，所研究的新算法表现出较高的收敛精度，证明了改进算法在单峰函数中的优越性能。



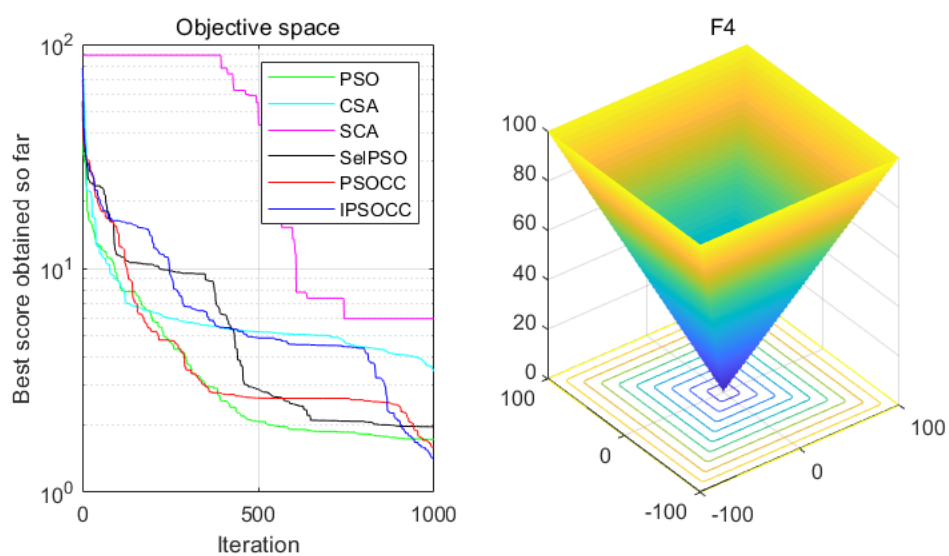
(a) 各算法函数 F<sub>1</sub> 收敛曲线



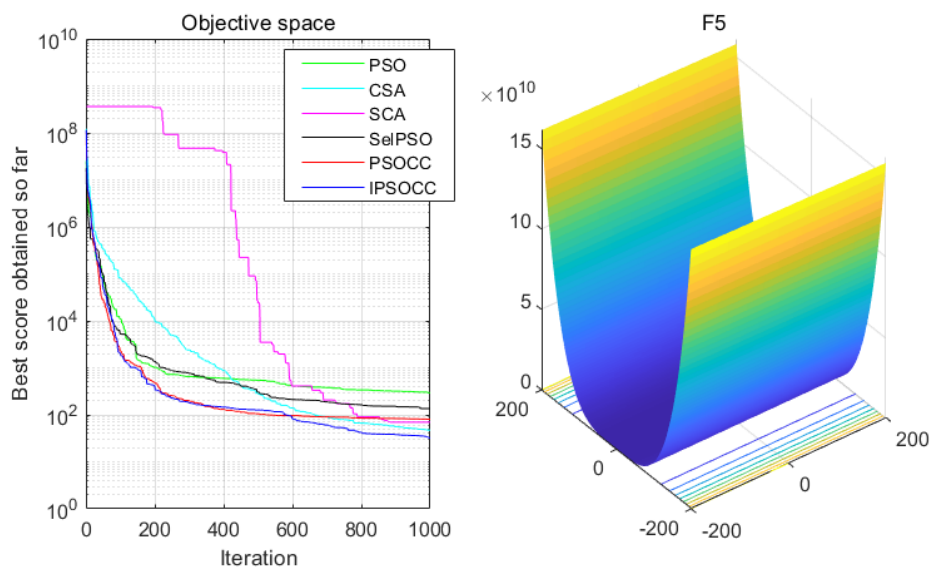
(b) 各算法函数  $F_2$  收敛曲线



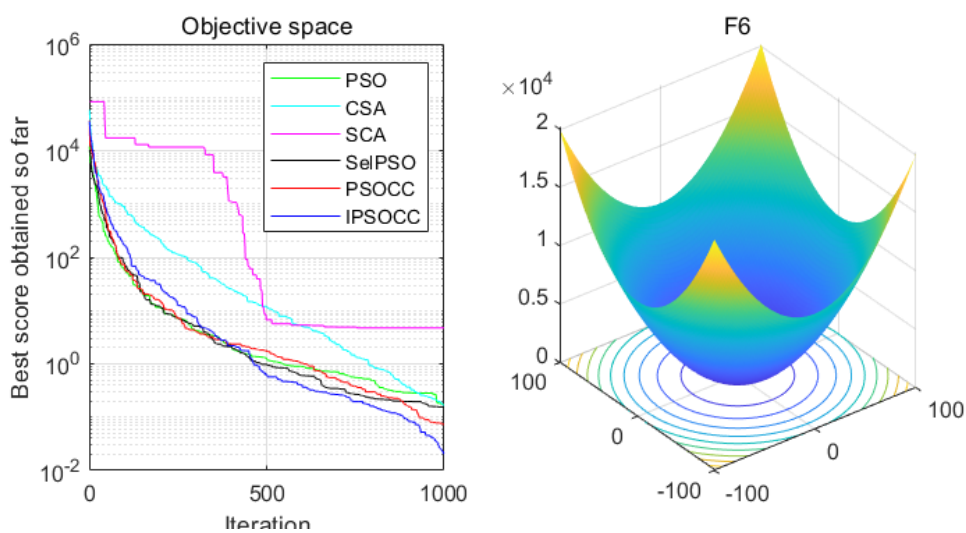
(c) 各算法函数  $F_3$  收敛曲线



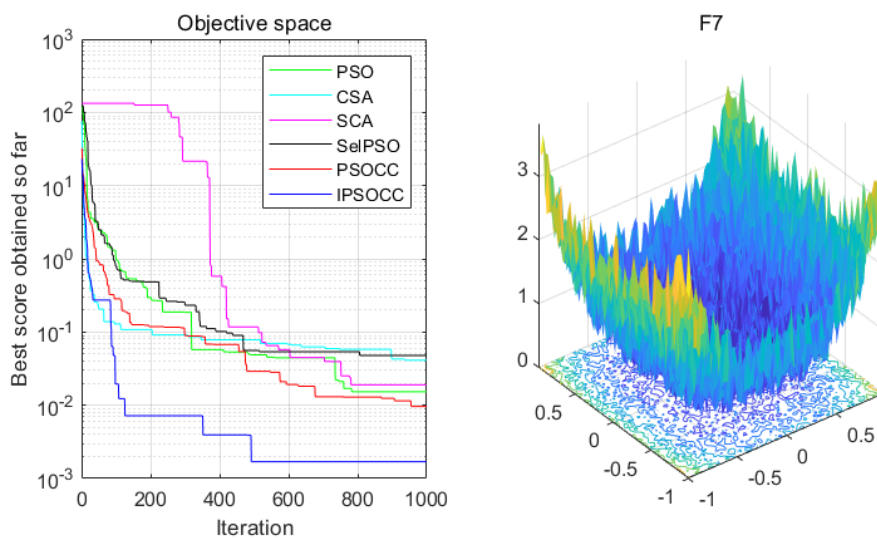
(d) 各算法函数  $F_4$  收敛曲线



(e) 各算法函数  $F_5$  收敛曲线



(f) 各算法函数  $F_6$  收敛曲线

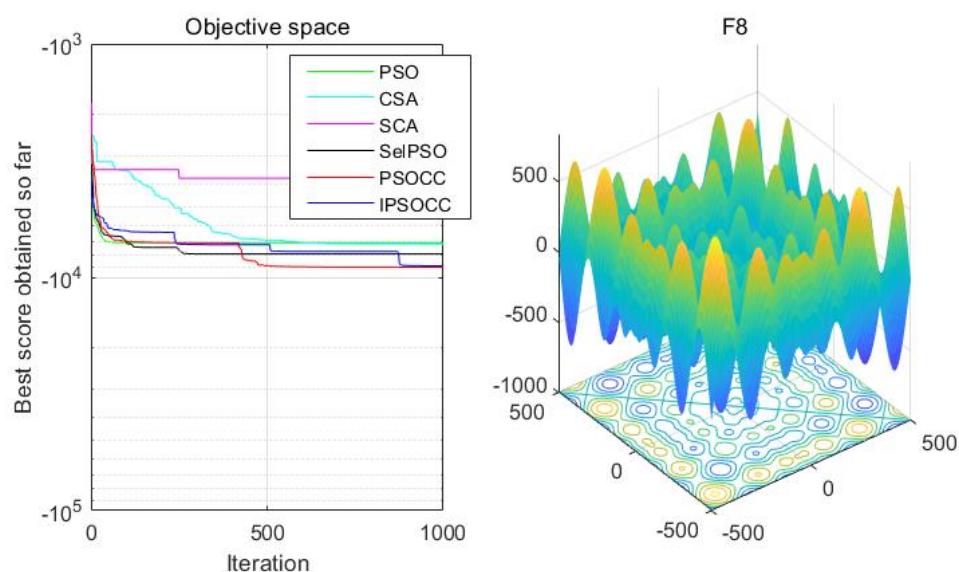


(g) 各算法函数  $F_7$  收敛曲线

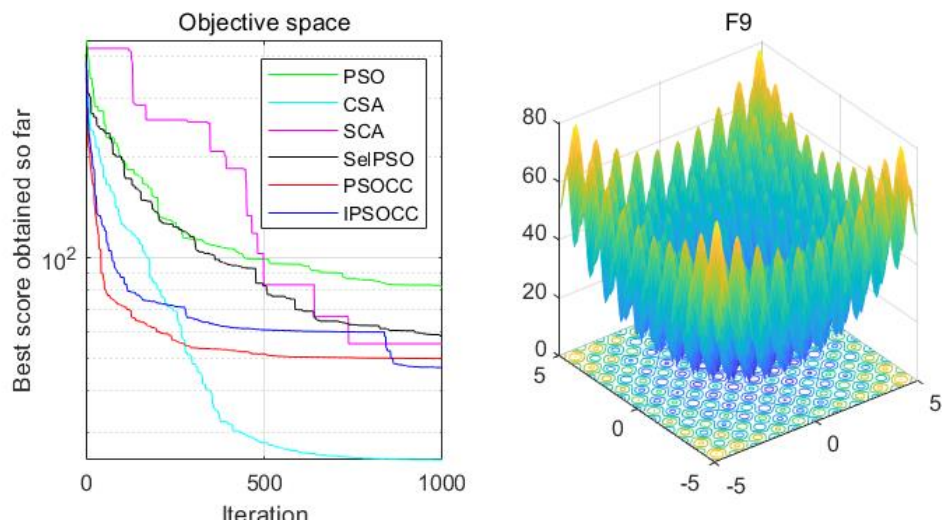
图 2-6 30 维度下单峰函数收敛曲线

## (2) 多峰函数

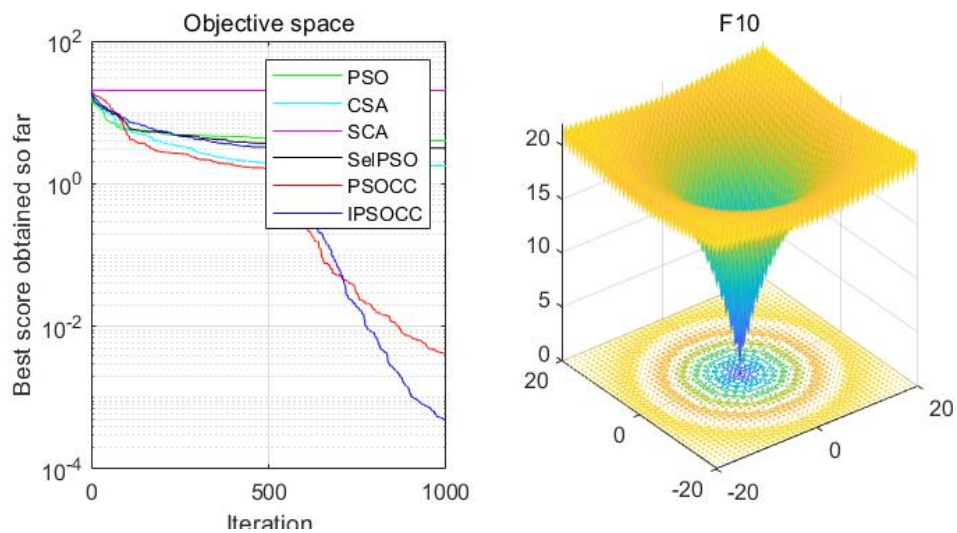
多峰函数 (F8 至 F13) 表现出复杂的函数结构, 具有多个局部最优解如图 2-7 所示。此类函数对过早收敛并陷入局部最优的算法可能会产生非全局最优的结果。因此, 在多峰函数下可以评估算法在多个局部最优值中识别全局最优值时的全局搜索能力。观察可知, PSOCC 算法与 IPSOCC 算法在大多数情况下收敛曲线的收敛速度以及其所能找到的最优值均优于对比算法。结果证明了 PSOCC 算法与 IPSOCC 算法的鲁棒性和全局搜索能力优于对比算法。



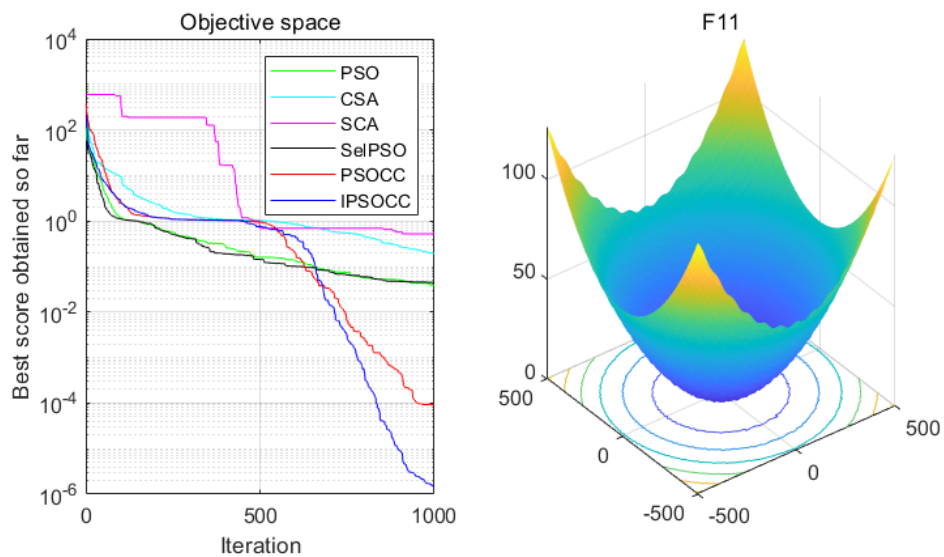
(a) 各算法函数 F<sub>8</sub> 收敛曲线



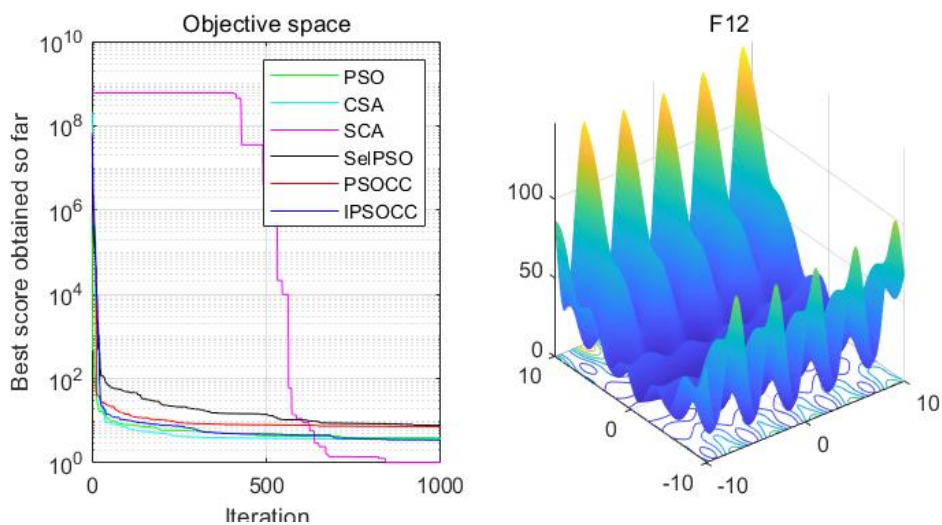
(b) 各算法函数 F<sub>9</sub> 收敛曲线



(c) 各算法函数  $F_{10}$  收敛曲线



(d) 各算法函数  $F_{11}$  收敛曲线



(e) 各算法函数  $F_{12}$  收敛曲线

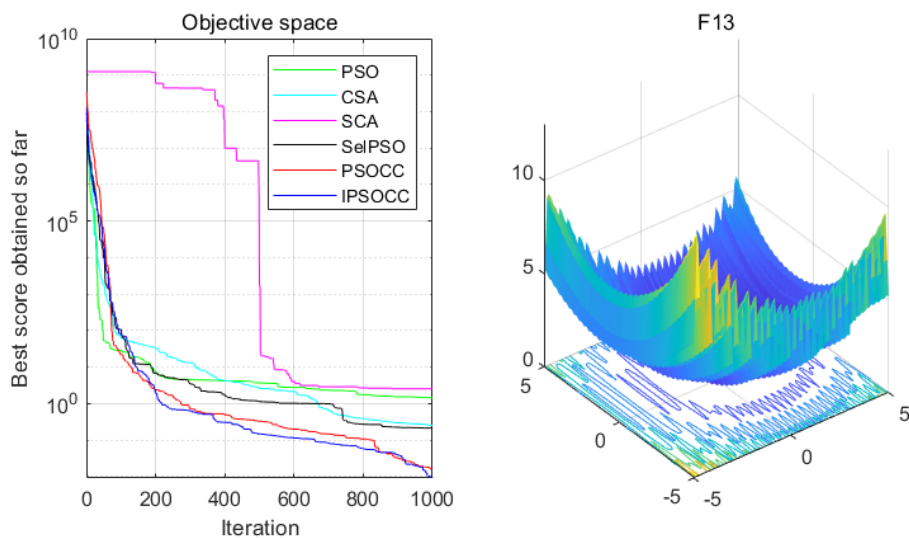
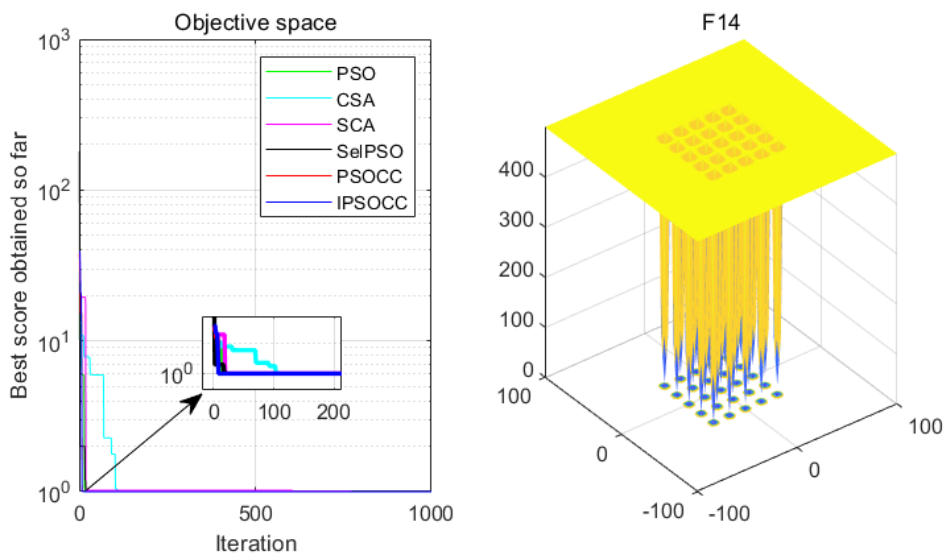
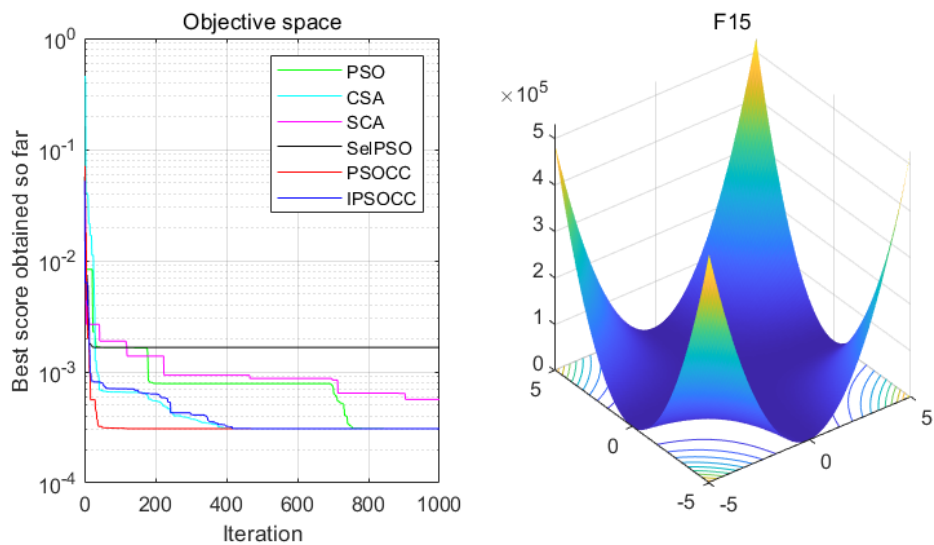
(f) 各算法函数  $F_{13}$  收敛曲线

图 2-7 30 维度下多峰函数收敛曲线

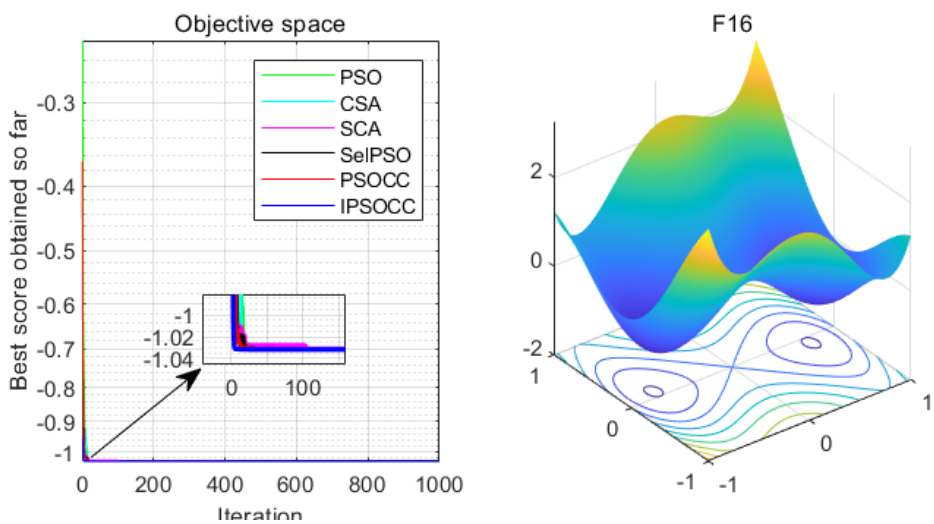
### (3) 复合函数

图 2-8 使用 CEC2005 基准函数对 PSOCC 算法和 IPSOCC 算法的评估展示了其在处理固定维度的复合函数（F14 至 F21）方面的稳健性能，复合函数通常包含多个局部最优解和复杂的空间搜索结构，要求优化算法具备强大的全局搜索能力和良好的收敛特性。实验结果表明，与其他对比算法相比，PSOCC 和 IPSOCC 具有更快的全局收敛速度，在这些复杂优化任务上的表现尤为突出，展现了较优的全局搜索能力和较强的搜索稳定性。这进一步证明了合作竞争机制在粒子群优化算法中的重要作用，提高了种群的多样性，避免陷入局部最优。

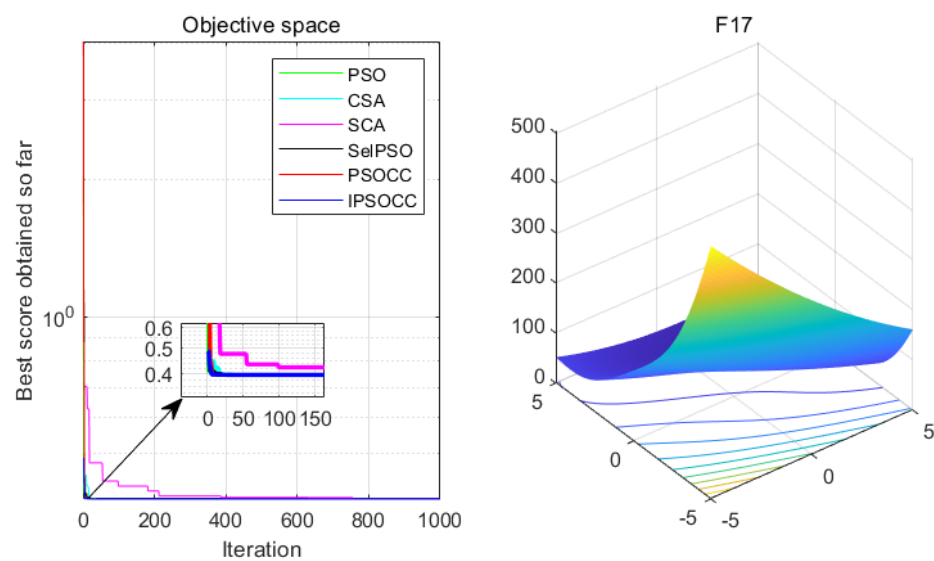
(a) 各算法函数  $F_{14}$  收敛曲线



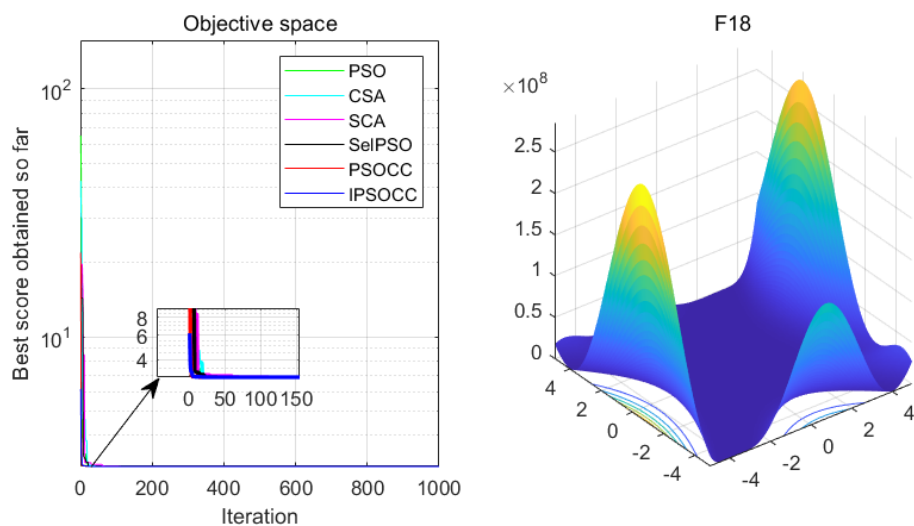
(b) 各算法函数 F<sub>15</sub> 收敛曲线



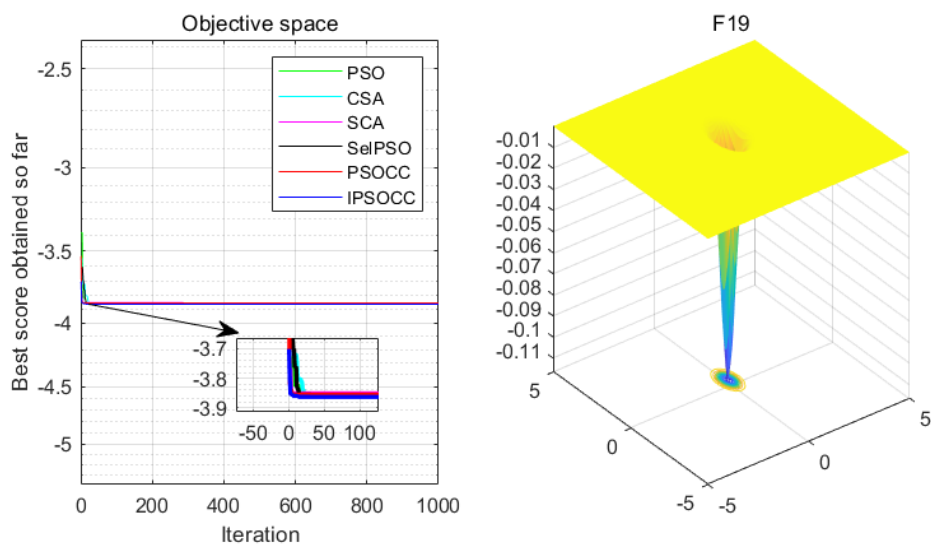
(c) 各算法函数 F<sub>16</sub> 收敛曲线



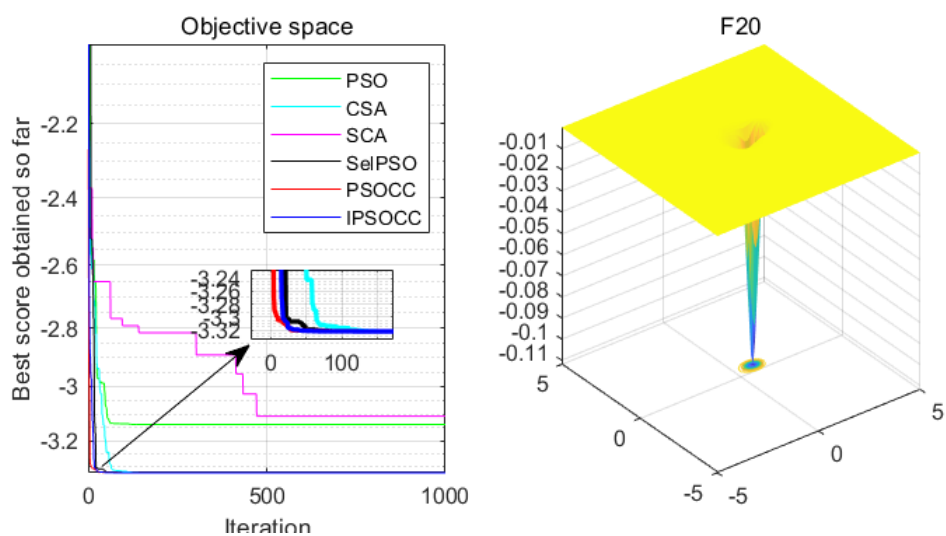
(d) 各算法函数 F<sub>17</sub> 收敛曲线



(e) 各算法函数  $F_{18}$  收敛曲线



(f) 各算法函数  $F_{19}$  收敛曲线



(g) 各算法函数  $F_{20}$  收敛曲线

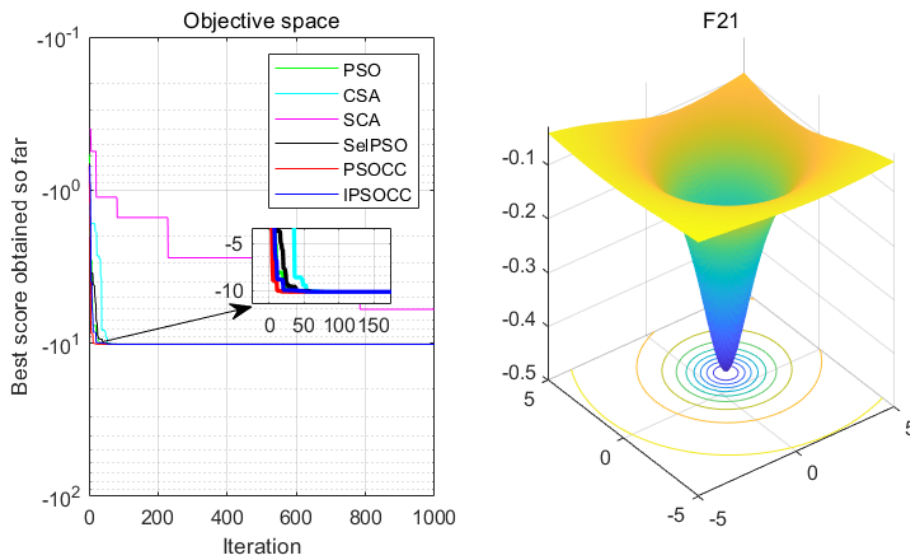
(h) 各算法函数  $F_{21}$  收敛曲线

图 2-8 固定维度下复合函数收敛曲线

## 2.5 本章小结

本章节研究了一种具有合作竞争机制的粒子群优化算法（PSOCC），通过在基本粒子群优化算法中引入粒子协同共享搜索策略以及粒子竞争部署策略，使得粒子在合作竞争中开发并挖掘求解空间。其次，通过参数学习分析了 PSOCC 算法在不同参数组合中的性能表现以及不同参数对算法的影响，从而获取算法性能最佳的参数组合。通过 AD 曲线研究并分析了 PSOCC 算法在求解空间的收敛情况。此外，进一步考虑了搜索过程中种群在信息的利用效率差异特征，研究了一种信息交互式策略，并设计了一种具有信息交互式合作竞争机制的粒子群优化算法（IPSOCC）。最后，通过 PSOCC 算法与 IPSOCC 算法应用于 CEC2005 基准函数的实验，结果表明，改进后的算法在单峰和多峰函数上均表现出良好的优化性能，展现了改进算法在高维与低维函数问题中的稳定性和较高的性能。

## 第 3 章 具有合作竞争机制的人类学习优化算法研究

### 3.1 人类学习优化算法

HLO 算法通过模拟人类的学习行为来进行寻优操作，通过模拟随机学习、个体学习和社会学习等过程，来实现问题的优化求解。HLO 使用的是二进制编码方式进行求解计算，每个个体都作为一个解，每个个体都由一串二进制字符组成，通过比特随机向三个算子学习达到更新个体以探索空间的目的。个体的二进制字符是由若干个比特组成，每个比特取 0 或 1，如公式（3.1）：

$$\begin{aligned} x_i &= [x_{i1} \ x_{i2} \ \cdots x_{ij} \ \cdots x_{iM}], \\ x_{ij} &\in \{0,1\}, 1 \leq i \leq N, 1 \leq j \leq M \end{aligned} \quad (3.1)$$

其中， $x_i$  表示第  $i$  个个体， $x_{ij}$  表示第  $i$  个个体的第  $j$  位， $N$  表示种群数量， $M$  表示解的长度。在算法开始时，需要随机初始化，生成随机的个体，个体中的比特为 0 或 1 的随机数。

#### 3.1.1 随机学习算子

人们对于最开始接触的新问题只能依靠随机尝试<sup>[45]</sup>，这种尝试就是一种随机学习，人们会在尝试中探索，在探索中获取知识。随机学习可以帮助个体探索和发现新的知识和信息，它可以引导个体去尝试新的事物和知识，从而开拓视野，拓展认知边界。因此，随机学习在人类学习的最初阶段是必不可少的过程。在人类学习算法中简单模拟出了随机学习机制如公式（3.2）：

$$x_{ij} = RLO = \begin{cases} 0, & 0 \leq r_{i1} \leq 0.5 \\ 1, & \text{else} \end{cases} \quad (3.2)$$

其中， $r_{i1}$  为 0 到 1 之间的随机值。当 HLO 的比特执行个体学习算子（Individual Learning Operator, ILO）时，当前比特则会赋予 0 或 1 的随机值。增加算法随机性，使得算法更好的探索空间，使算法具备找到全局最优的可能。在陷入局部最优时，随机学习算子使算法具备跳出局部最优的可能。

#### 3.1.2 个体学习算子

个体学习可以帮助人们获取知识、技能和经验，从而提高个人的知识和能力。个人学习是人类获取新知识和信息的主要途径<sup>[46]</sup>。通过学习，个人可以了解新的概念、原理和事实，不断扩展自己的知识体系，从而提升自己的学习效率。为了

简单模拟出这种个体学习行为,种群中的每个个体的最优解储存在个体知识数据库 (Individual Knowledge Database, IKD) 中,其被定义为公式(3.3)和公式(3.4)。

$$IKD = \begin{bmatrix} ikd_1 \\ ikd_2 \\ \vdots \\ ikd_i \\ \vdots \\ ikd_N \end{bmatrix}, 1 \leq i \leq N \quad (3.3)$$

$$ikd_i = \begin{bmatrix} ikd_{i1} \\ ikd_{i2} \\ \vdots \\ ikd_{ip} \\ \vdots \\ ikd_{iL} \end{bmatrix} = \begin{bmatrix} ik_{i1,1} & ik_{i1,2} & \cdots & ik_{i1,j} & \cdots & ik_{i1,M} \\ ik_{i2,1} & ik_{i2,2} & \cdots & ik_{i2,j} & \cdots & ik_{i2,M} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{ip,1} & ik_{ip,2} & \cdots & ik_{ip,j} & \cdots & ik_{ip,M} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{iL,1} & ik_{iL,2} & \cdots & ik_{iL,j} & \cdots & ik_{iL,M} \end{bmatrix}, 1 \leq p \leq L \quad (3.4)$$

其中,  $ikd_i$  为个体  $i$  的个体知识数据库,  $ik_{ip}$  表示的是个体  $i$  的第  $p$  个个体最优解。个体  $i$  的当前比特执行个体学习算子 (ILO) 时,将会在 IKD 中的  $ikd_i$  数据库中随机选取  $p$ , 如公式 (3.5) :

$$x_{ij} = ik_{ip,j} \quad (3.5)$$

在种群更新后,将计算个体适应度值,以更新 IKD 数据库。当  $ikd_i$  中的数据大小满足预设值  $L$ , 若当前适应度值小于数据库中最差值,则当前个体的位置取代数据库中适应度最差的位置。 $ikd_i$  中数据大小未达到预设值  $L$ ,则将当前个体位置直接记入  $ikd_i$ 。由于 HLO 是针对单目标问题设计的,因此这里的预设值  $L$  设为 1,以增强算法在单目标问题的性能和精确度。因此,当当前适应度值优于数据库 IKD 时,则将新的候选解替代原先数据库存储的解。

### 3.1.3 社会学习算子

社会学习对于人类学习有着至关重要的意义<sup>[49]</sup>,社会学习促进了个体的认知发展。通过观察和与他人讨论、交流,个体获取新知识、建立概念、形成思维模式,并学会解决问题,社会学习鼓励个体分享、合作和交流,促进了创新的产生。通过学习社会知识,个体汲取新思想、灵感和观点,从而推动学习和创新。社会知识数据库 (Social Knowledge Database, SKD) 为群体中最优适应度位置的

数据库，储存着当前为止种群最优适应度的位置。HLO 中通过社会学习算子，使个体能够向着种群适应度最优进行学习，个体们能够在社会学习中学习到更好的知识，从而达到提升自身能力的目的，SKD 如公式（3.6）：

$$SKD = \begin{bmatrix} skd_1 \\ skd_2 \\ \vdots \\ skd_q \\ \vdots \\ skd_H \end{bmatrix} = \begin{bmatrix} sk_{11} & sk_{12} & \cdots & sk_{1j} & \cdots & sk_{1M} \\ sk_{21} & sk_{22} & \cdots & sk_{2j} & \cdots & sk_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{q1} & sk_{q2} & \cdots & sk_{qj} & \cdots & sk_{qM} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{H1} & sk_{H2} & \cdots & sk_{Hj} & \cdots & sk_{HM} \end{bmatrix}, 1 \leq q \leq H \quad (3.6)$$

其中， $skd_q$  表示 SKD 的第  $q$  个解， $H$  是 SKD 数据库的大小。为了能在单目标问题中能有更好的性能，在 HLO 中 SKD 数据库大小  $H$  被设为 1。群体中出现优于 SKD 数据库中存储的适应度时，则更新 SKD。

当 HLO 执行社会学习算子后通过公式（3.7）更新候选解。

$$x_{ij} = sk_{qj} \quad (3.7)$$

## 3.2 具有合作竞争机制的人类学习优化算法

### 3.2.1 合作竞争学习算子

合作竞争在群体中具有深远的影响<sup>[47]</sup>，能够显著激发个体的学习能力<sup>[48]</sup>。这种机制不仅在教育、工作和社会交往中发挥着至关重要的作用，也是推动人类学习和进步的重要因素。竞争能够激发个体的积极性和动力，促使其挑战自我，追求更高的成就；而合作则通过整合不同个体的知识和经验，产生集体智慧，通常能够获得比个人独立工作更优的结果。因此，合作竞争机制在人类学习的过程具有重要意义，对此本文引入了一种合作竞争学习算子（Cooperative Competitive Learning Operator, CCLO），研究了具有合作竞争机制的人类学习优化算法（Cooperative-Competitive Human Learning Optimization, CCHLO）。合作竞争机制的引入使算法在更新过程中增加了更多的选择，提升了算法个体间的信息交流，并且能够更有效地获取周围空间的信息，以提升算法性能。CCLO 算子通过在个体间建立联系，使得每个个体不再各自学习。在个体位置预更新时，个体通过公式（3.8）计算其与其他个体之间的欧几里得距离，

$$d(x_i, x_o) = \sqrt{\sum_{o=1}^{N-1} (x_i - x_o)^2} \quad (3.8)$$

其中,  $x_i$  为当前个体,  $x_o$  为除当前个体外的其他个体,  $N$  为种群数量,  $d(x_i, x_o)$  代表当前个体与其他个体间的欧几里得距离。选择与其欧几里得距离较小的  $G$  个个体进行合作, 如公式 (3.9) :

$$d(x_i, x_{o1}) \leq d(x_i, x_{o2}) \leq d(x_i, x_{o3}) \leq \dots \leq d(x_i, x_{oG}) \quad (3.9)$$

其中,  $x_{o1}$  为距离当前个体欧几里得距离最近的个体。其次, 使当前个体与被选择个体对应的 IKD 的适应度值进行竞争, 竞争胜者的 IKD 将会被当前个体学习, 如公式 (3.10) :

$$fit(ikd_{oG}^1) \leq fit(ikd_{oG}^2) \leq fit(ikd_{oG}^3) \leq \dots \leq fit(ikd_{oG}^G) \quad (3.10)$$

其中,  $fit(ikd_{oG}^1)$  为个体通过竞争后优胜者的 IKD 适应度值,  $ikd_{oG}^1$  为优胜者的个体数据库, 从而选择优胜者的 IKD 进行学习。其更新如公式 (3.11) :

$$x_{ij} = CCLO = ik_{oG,j}^1 \quad (3.11)$$

因此, 合作竞争学习算子是使当前个体与距离自身较近的其他个体进行合作, 读取各自的 IKD 进行交流, 选中个体的 IKD 适应度进行竞争, 优胜者的 IKD 信息将会被识别, 被 CCLO 算子记录, 并将其作为当前个体学习目标。因此, CCLO 算子的引入能够增加算法的收敛性, 使得当前更新个体能够向着更好的 IKD 进行学习。

### 3.2.2 重新学习策略

在人类学习的过程中, 重新学习可以帮助纠正之前的错误理解或不良习惯, 促进正确知识的形成。并且, 在不断变化的环境中, 重新学习有助于个体调整自己的行为 and 策略, 以适应新的挑战。重新学习是终身学习的一个重要组成部分, 它鼓励个体持续学习和成长。这里将重新学习机制引入 CCHLO 算法中, 是否进行重新学习是需要判断标准, 首先判断群体中的第  $i$  个体中的 IKD 是否不更新, 如公式 (3.12), 将不更新次数用  $T_i$  表示。当  $T_i$  达到  $T_1$ , 则进行重新学习, 重新学习公式如 (3.13) :

$$\begin{cases} T_i = T_i + 1 & \text{if } ikd_i^t = ikd_i^{t-1} \\ T_i = 0 & \text{if } ikd_i^t \neq ikd_i^{t-1} \end{cases} \quad (3.12)$$

$$ikd_{ij} = \begin{cases} 0, & 0 \leq r_{12} \leq 0.5 \\ 1, & else \end{cases} \quad (3.13)$$

重新学习机制的引入使得优化算法具备更强的全局探索能力,能够有效跳出局部最优的陷阱。当个体在搜索过程中陷入局部最优时,重新学习机制通过重新初始化部分个体的 IKD 值以调整搜索策略,迫使个体跳出当前搜索区域,向求解空间中新的潜在最优值区域进行探索。因此,重新学习机制能够增加算法种群的多样性,避免算法陷入局部最优,提升算法在复杂优化问题中的全局搜索能力,使算法不仅能改善局部搜索的局限性,还能为全局最优解的发现提供基础,从而算法增强整体优化性能,提高解的质量和精确性。

### 3.2.3 CCHLO 算法实现

CCHLO 在标准 HLO 的基础上增加了合作竞争学习算子,标准的 HLO 算法执行随机学习算子、个体学习算子和社会学习算子。而 CCHLO 算法执行如公式 (3.14), 将执行随机学习算子、个体学习算子、竞争和合作学习算子以及社会学习算子。

$$x_{ij} = \begin{cases} RLO & 0 \leq r_{13} \leq pr \\ ik_{ip,j} & pr \leq r_{13} \leq pi \\ CCLO & pi \leq r_{13} \leq pc \\ sk_{qj} & else \end{cases} \quad (3.14)$$

其中,  $r_{13}$  为 0 和 1 之间的随机数,  $(pr-0)$  为算法执行随机学习算子的概率,  $(pi-pr)$  为算法执行个体学习算子的概率,  $(pc-pi)$  为算法执行竞争和合作学习算子的概率,  $(1-pc)$  为执行社会学习算子的概率。CCHLO 算法的具体流程图如图 3-1 所示。

通过在标准 HLO 算法中加入竞争和合作学习算子可以增强算法的学习能力和适应性,增加算法的搜索空间,提高算法的泛化能力和学习速度。通过引入新的算子,算法可以更全面地探索数据的特征和规律,从而更准确地进行学习和预测。竞争和合作学习算子可以帮助算法处理更复杂的数据和任务,提高算法的适应性和灵活性。再通过重新学习的引入,使得算法在迭代时陷入局部最优时能够跳出局部最优。通过不断扩展算法的功能和能力,算法可以更好地适应不同的环境和需求,从而取得更好的学习效果 and 性能。

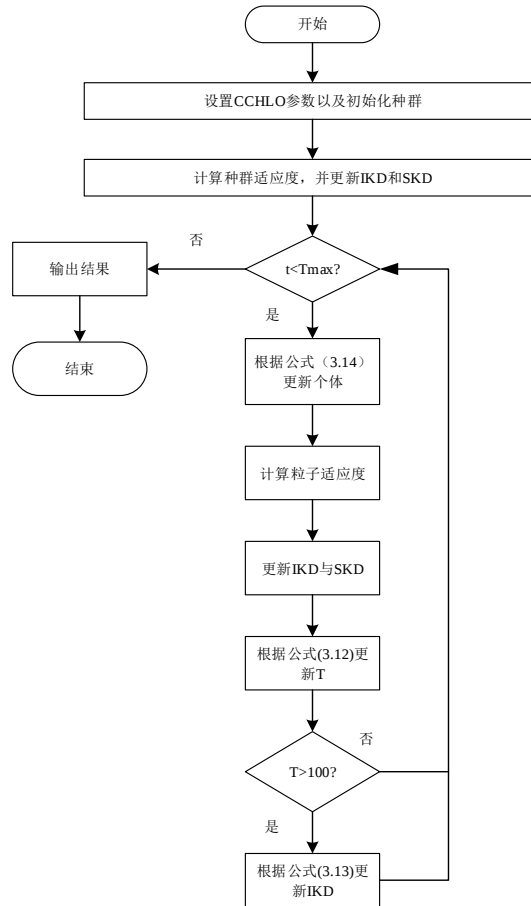


图 3-1 CCHLO 流程图

### 3.2.4 CCHLO 算法参数分析

本节叙述了对 CCHLO 的参数研究以及分析，分析以及选择合理的控制参数。这里的  $p_r$  为标准 HLO 的默认值， $p_r$  为  $5/M$ 。考虑到引入的新算子参数的设定对个体学习算子的影响，这里使用交叉组合法，对  $p_i$  参数以及  $p_c$  参数的设定进行研究，以及选择与其欧氏距离较小的个体数  $G$  对整个算法参数选择的影响，各参数取值如表 3-1。由于，F6 函数作为单峰函数，具有明确的全局最优解，可以有效考察算法在全局优化中的收敛性和搜索能力。通过对 F6 函数的优化，可以评估 CCLO 算子在单一目标下的搜索性能，尤其是其在避免过早收敛和进行广泛搜索的能力。F12 函数作为多峰函数，包含多个局部最优解，是测试算法局部搜索能力和避免陷入局部最优解的重要场景。通过选择 F12 作为多峰函数，能够测试 CCLO 算子在复杂、非线性问题空间中的表现，特别是在算法逐步收敛时，如何平衡局部搜索与全局搜索的效率，从而检验算法在面对多解问题时的适应性和准确性。因此，选择基准函数 F6 和 F12 能够全面评估自适应变化的 CCLO 算子在不同优化环境下的表现，既能测试其全局优化能力，也能考察其在多峰问题

表 3-1 CCHLO 参数取值情况

| 参数   | 取值情况 |      |      |      |      |      |
|------|------|------|------|------|------|------|
|      | 取值 1 | 取值 2 | 取值 3 | 取值 4 | 取值 5 | 取值 6 |
| $G$  | 4    | 5    | 6    |      |      |      |
| $pi$ | 0.76 | 0.78 | 0.80 | 0.82 | 0.84 |      |
| $pc$ | 0.85 | 0.87 | 0.89 | 0.91 | 0.93 | 0.95 |

表 3-2  $G=4$  各维度参数正交表

| 序号 | 参数   |      | F6              | F12             | F6              | F12             |
|----|------|------|-----------------|-----------------|-----------------|-----------------|
|    | $pi$ | $pc$ | $D=30$          |                 | $D=50$          |                 |
| 1  | 0.76 | 0.85 | 3.83E+00        | 6.68E-01        | 6.59E+00        | <b>4.76E-01</b> |
| 2  | 0.76 | 0.87 | 3.86E+00        | 5.94E-01        | <b>6.42E+00</b> | 6.18E-01        |
| 3  | 0.76 | 0.89 | <b>3.63E+00</b> | 5.59E-01        | 6.54E+00        | 5.69E-01        |
| 4  | 0.76 | 0.91 | 3.96E+00        | 6.03E-01        | 6.60E+00        | <b>4.22E-01</b> |
| 5  | 0.76 | 0.93 | 3.83E+00        | 7.30E-01        | 6.46E+00        | 6.44E-01        |
| 6  | 0.76 | 0.95 | <b>3.67E+00</b> | 8.50E-01        | 6.79E+00        | 7.34E-01        |
| 7  | 0.78 | 0.85 | 3.88E+00        | 7.22E-01        | <b>6.28E+00</b> | 5.72E-01        |
| 8  | 0.78 | 0.87 | 3.92E+00        | 6.14E-01        | <b>6.37E+00</b> | 6.51E-01        |
| 9  | 0.78 | 0.89 | 3.84E+00        | 5.97E-01        | 6.68E+00        | 5.41E-01        |
| 10 | 0.78 | 0.91 | 4.26E+00        | 7.42E-01        | 6.64E+00        | <b>5.15E-01</b> |
| 11 | 0.78 | 0.93 | 3.92E+00        | 5.48E-01        | 6.56E+00        | 5.23E-01        |
| 12 | 0.78 | 0.95 | 3.84E+00        | 6.32E-01        | 8.71E+00        | 6.50E-01        |
| 13 | 0.8  | 0.85 | <b>3.70E+00</b> | 7.29E-01        | 6.91E+00        | 6.04E-01        |
| 14 | 0.8  | 0.87 | 4.06E+00        | 6.62E-01        | 6.73E+00        | 6.10E-01        |
| 15 | 0.8  | 0.89 | 4.09E+00        | 5.50E-01        | 6.52E+00        | <b>5.07E-01</b> |
| 16 | 0.8  | 0.91 | 3.81E+00        | 6.59E-01        | 6.50E+00        | 5.44E-01        |
| 17 | 0.8  | 0.93 | 3.99E+00        | 6.48E-01        | 6.49E+00        | 6.49E-01        |
| 18 | 0.8  | 0.95 | 3.80E+00        | 6.64E-01        | 6.79E+00        | 7.65E-01        |
| 19 | 0.82 | 0.85 | 3.99E+00        | <b>5.11E-01</b> | 6.81E+00        | <b>5.01E-01</b> |
| 20 | 0.82 | 0.87 | <b>3.71E+00</b> | 5.82E-01        | <b>6.45E+00</b> | 7.13E-01        |
| 21 | 0.82 | 0.89 | 3.82E+00        | 7.37E-01        | <b>6.43E+00</b> | 6.09E-01        |
| 22 | 0.82 | 0.91 | 4.07E+00        | 6.35E-01        | 6.56E+00        | 5.75E-01        |
| 23 | 0.82 | 0.93 | 3.92E+00        | 6.08E-01        | 6.82E+00        | 6.07E-01        |
| 24 | 0.82 | 0.95 | 3.83E+00        | 6.75E-01        | 6.54E+00        | 7.78E-01        |
| 25 | 0.84 | 0.85 | 4.10E+00        | 7.72E-01        | 6.60E+00        | 7.61E-01        |
| 26 | 0.84 | 0.87 | <b>3.79E+00</b> | 6.16E-01        | 6.79E+00        | 6.94E-01        |
| 27 | 0.84 | 0.89 | 3.83E+00        | 6.27E-01        | 6.61E+00        | 6.53E-01        |
| 28 | 0.84 | 0.91 | 4.11E+00        | <b>5.20E-01</b> | 6.54E+00        | 6.17E-01        |
| 29 | 0.84 | 0.93 | 3.84E+00        | 5.93E-01        | 6.51E+00        | 6.53E-01        |
| 30 | 0.84 | 0.95 | 3.81E+00        | 6.91E-01        | 6.71E+00        | 7.53E-01        |

表 3-3  $G=5$  各维度参数正交表

| 序号 | 参数   |      | F6              | F12             | F6              | F12             |
|----|------|------|-----------------|-----------------|-----------------|-----------------|
|    | $pi$ | $pc$ | $D = 30$        |                 | $D = 50$        |                 |
| 1  | 0.76 | 0.85 | 4.19E+00        | <b>5.41E-01</b> | 6.55E+00        | <b>5.12E-01</b> |
| 2  | 0.76 | 0.87 | <b>3.68E+00</b> | 6.14E-01        | 6.82E+00        | 5.75E-01        |
| 3  | 0.76 | 0.89 | <b>3.70E+00</b> | 6.70E-01        | 6.69E+00        | 5.99E-01        |
| 4  | 0.76 | 0.91 | 4.07E+00        | 6.55E-01        | 6.51E+00        | 6.88E-01        |
| 5  | 0.76 | 0.93 | 3.84E+00        | 6.02E-01        | 6.62E+00        | <b>4.83E-01</b> |
| 6  | 0.76 | 0.95 | 3.87E+00        | 5.52E-01        | 6.77E+00        | 7.08E-01        |
| 7  | 0.78 | 0.85 | 4.02E+00        | 6.06E-01        | 6.68E+00        | <b>5.16E-01</b> |
| 8  | 0.78 | 0.87 | 3.86E+00        | <b>5.40E-01</b> | 6.91E+00        | <b>4.81E-01</b> |
| 9  | 0.78 | 0.89 | 4.27E+00        | <b>5.41E-01</b> | 6.63E+00        | 5.37E-01        |
| 10 | 0.78 | 0.91 | 4.09E+00        | 5.75E-01        | 6.78E+00        | 6.31E-01        |
| 11 | 0.78 | 0.93 | <b>3.76E+00</b> | 6.70E-01        | 6.58E+00        | 5.69E-01        |
| 12 | 0.78 | 0.95 | <b>3.79E+00</b> | 5.75E-01        | <b>6.40E+00</b> | 6.13E-01        |
| 13 | 0.8  | 0.85 | 4.03E+00        | 5.44E-01        | 6.64E+00        | 5.56E-01        |
| 14 | 0.8  | 0.87 | 3.89E+00        | 5.70E-01        | 6.76E+00        | 6.43E-01        |
| 15 | 0.8  | 0.89 | 3.91E+00        | 6.94E-01        | 6.71E+00        | <b>5.10E-01</b> |
| 16 | 0.8  | 0.91 | 4.15E+00        | <b>5.22E-01</b> | 6.82E+00        | 5.75E-01        |
| 17 | 0.8  | 0.93 | 3.90E+00        | 6.41E-01        | 6.68E+00        | 6.36E-01        |
| 18 | 0.8  | 0.95 | 3.84E+00        | 6.11E-01        | 6.82E+00        | 6.22E-01        |
| 19 | 0.82 | 0.85 | 3.85E+00        | 6.08E-01        | 6.47E+00        | <b>4.98E-01</b> |
| 20 | 0.82 | 0.87 | 4.04E+00        | 7.58E-01        | 6.56E+00        | 6.21E-01        |
| 21 | 0.82 | 0.89 | <b>3.79E+00</b> | 6.77E-01        | 7.05E+00        | 5.83E-01        |
| 22 | 0.82 | 0.91 | <b>3.74E+00</b> | <b>5.37E-01</b> | <b>6.34E+00</b> | 5.42E-01        |
| 23 | 0.82 | 0.93 | <b>3.78E+00</b> | <b>4.84E-01</b> | <b>6.44E+00</b> | <b>5.13E-01</b> |
| 24 | 0.82 | 0.95 | 4.04E+00        | 6.17E-01        | 6.56E+00        | 7.38E-01        |
| 25 | 0.84 | 0.85 | 4.07E+00        | 6.63E-01        | <b>6.31E+00</b> | 7.24E-01        |
| 26 | 0.84 | 0.87 | 3.92E+00        | 6.02E-01        | 6.69E+00        | 6.53E-01        |
| 27 | 0.84 | 0.89 | <b>3.78E+00</b> | 7.81E-01        | 6.83E+00        | 7.37E-01        |
| 28 | 0.84 | 0.91 | 3.95E+00        | 6.54E-01        | <b>6.44E+00</b> | 7.02E-01        |
| 29 | 0.84 | 0.93 | 3.80E+00        | 6.19E-01        | 6.80E+00        | 6.39E-01        |
| 30 | 0.84 | 0.95 | 4.10E+00        | 5.92E-01        | 6.67E+00        | 7.32E-01        |

表 3-4  $G=6$  各维度参数正交表

| 序号 | 参数   |      | F6              | F12      | F6              | F12      |
|----|------|------|-----------------|----------|-----------------|----------|
|    | $pi$ | $pc$ | $D = 30$        |          | $D = 50$        |          |
| 1  | 0.76 | 0.85 | 3.86E+00        | 6.53E-01 | <b>6.44E+00</b> | 5.51E-01 |
| 2  | 0.76 | 0.87 | 3.93E+00        | 8.40E-01 | 7.01E+00        | 5.23E-01 |
| 3  | 0.76 | 0.89 | 3.84E+00        | 5.43E-01 | <b>6.41E+00</b> | 5.84E-01 |
| 4  | 0.76 | 0.91 | <b>3.79E+00</b> | 6.81E-01 | <b>6.45E+00</b> | 6.07E-01 |

续表：3-4

|    |      |      |                 |                 |                 |                 |
|----|------|------|-----------------|-----------------|-----------------|-----------------|
| 5  | 0.76 | 0.93 | 3.86E+00        | 6.74E-01        | 6.58E+00        | <b>4.68E-01</b> |
| 6  | 0.76 | 0.95 | 3.93E+00        | <b>5.20E-01</b> | 6.57E+00        | 5.83E-01        |
| 7  | 0.78 | 0.85 | 3.92E+00        | 6.66E-01        | 6.67E+00        | 5.63E-01        |
| 8  | 0.78 | 0.87 | 3.88E+00        | <b>4.57E-01</b> | 6.77E+00        | 5.51E-01        |
| 9  | 0.78 | 0.89 | 4.02E+00        | 5.49E-01        | <b>6.42E+00</b> | 5.58E-01        |
| 10 | 0.78 | 0.91 | 3.94E+00        | 5.63E-01        | 6.91E+00        | 5.73E-01        |
| 11 | 0.78 | 0.93 | 3.86E+00        | <b>5.04E-01</b> | 6.54E+00        | <b>5.12E-01</b> |
| 12 | 0.78 | 0.95 | 3.92E+00        | 5.74E-01        | 6.46E+00        | <b>5.15E-01</b> |
| 13 | 0.8  | 0.85 | 4.08E+00        | <b>4.80E-01</b> | 6.75E+00        | 5.26E-01        |
| 14 | 0.8  | 0.87 | 3.92E+00        | 7.09E-01        | <b>6.39E+00</b> | 6.20E-01        |
| 15 | 0.8  | 0.89 | 3.89E+00        | <b>5.29E-01</b> | 6.77E+00        | 6.19E-01        |
| 16 | 0.8  | 0.91 | 3.87E+00        | 6.75E-01        | <b>6.44E+00</b> | 6.16E-01        |
| 17 | 0.8  | 0.93 | <b>3.72E+00</b> | 6.76E-01        | 6.82E+00        | 5.17E-01        |
| 18 | 0.8  | 0.95 | 4.17E+00        | <b>5.08E-01</b> | 6.61E+00        | 6.56E-01        |
| 19 | 0.82 | 0.85 | 3.80E+00        | 7.61E-01        | 6.53E+00        | 6.47E-01        |
| 20 | 0.82 | 0.87 | 3.86E+00        | 6.87E-01        | 6.73E+00        | 6.56E-01        |
| 21 | 0.82 | 0.89 | 3.79E+00        | <b>5.20E-01</b> | 6.75E+00        | 6.00E-01        |
| 22 | 0.82 | 0.91 | 4.20E+00        | 6.12E-01        | 6.51E+00        | 5.49E-01        |
| 23 | 0.82 | 0.93 | 3.83E+00        | 7.52E-01        | 6.82E+00        | 5.70E-01        |
| 24 | 0.82 | 0.95 | 4.06E+00        | 6.65E-01        | 6.67E+00        | 5.52E-01        |
| 25 | 0.84 | 0.85 | 3.87E+00        | 5.52E-01        | 6.51E+00        | 7.20E-01        |
| 26 | 0.84 | 0.87 | 3.85E+00        | 5.92E-01        | 6.70E+00        | 6.85E-01        |
| 27 | 0.84 | 0.89 | 3.81E+00        | 5.66E-01        | 6.59E+00        | 7.37E-01        |
| 28 | 0.84 | 0.91 | 3.80E+00        | <b>4.80E-01</b> | 6.63E+00        | 6.18E-01        |
| 29 | 0.84 | 0.93 | 3.95E+00        | 6.22E-01        | 6.88E+00        | 7.31E-01        |
| 30 | 0.84 | 0.95 | <b>3.58E+00</b> | 7.44E-01        | 6.49E+00        | 7.72E-01        |

中的局部搜索和收敛特性，从而提供更加充分的实验依据。将求解空间维度设为 30，种群规模设为 50，最大迭代次数设为 1000，并且每个函数独立运行 50 次，以求取平均值作为评价性能的指标，其中所有参数在同一函数中的较优的数值用粗体显示。表 3-2、表 3-3、表 3-4 中结果表明，当  $G$ 、 $pi$ 、 $pc$  设定为 5、0.82 和 0.93 时，CCHLO 的综合性能为最佳。因此，将该参数设定为本次工作的固定参数值。表 3-2、表 3-3、表 3-4 的结果表明，当参数  $pi$  大于或等于 0.8 时，算法的性能优于  $pi$  小于 0.8 的情况。这是由于在人类学习算法中，个体学习算子的执行概率远高于其他算子，因此，参数  $pi$  的选定不能过小，否则会破坏算法的平衡，导致性能下降。此外，竞争与合作学习算子的独特机制使得参数  $G$  对算法的收敛性产生重要影响。该参数决定了算法向周围较优个体的 IKD 进行学习的选择，

不仅增加了个体的认知度，还激发了个体的学习能力。结合重新学习机制，该机制能够在算法陷入局部最优时，迅速提高局部搜索能力。从实验结果来看，CCLO算子的引入显著增强了算法的开发能力和个体的认知能力，而重新学习机制的加入则提高了算法在遇到局部最优时对其他搜索区域的探索能力。通过这种机制的结合，算法能够实现开发与探索之间的平衡，从而提升了整体优化性能。

### 3.2.5 合作竞争学习算子功能分析

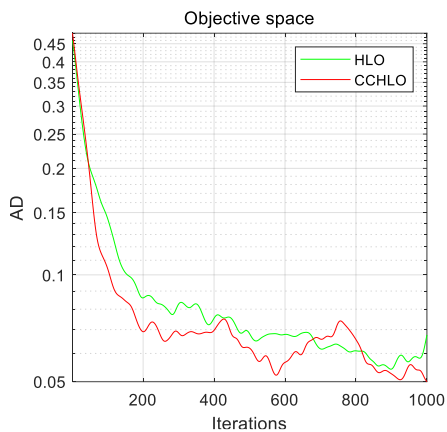
为了清晰的呈现出 CCLO 算子的引入对算法的影响，将 CCHLO 算法与 HLO 算法在 CEC2005 的 F6 和 F12 两个基准函数中进行结果比较。CCHLO 算法参数设定用前文叙述的默认值，HLO 算法参数用其默认值。将两个算法分别在 30 维和 50 维的 F6 和 F12 测试集中求得平均距离（AD）曲线以及适应度曲线，如图 3-2、图 3-3、图 3-4、图 3-5。由于，平均距离曲线可以反映数据点之间的平均距离随着类别之间的距离变化的情况，如，种群中个体之间的分离度、数据集的聚类情况以及进行算法的性能对比。因此，选择用 AD 曲线以及适应度曲线来检测算法对于空间的探索与开发能力。平均距离曲线如公式（3.15）：

$$AD = \frac{\sum_{i=1}^{N-1} \sum_{j=1}^M |ik_{ij} - sk_j|}{M \times (N-1)}, 1 \leq i \leq N-1, 1 \leq j \leq M \quad (3.15)$$

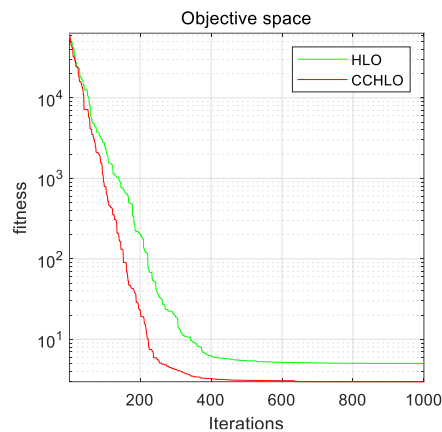
可以看出，在四个基准函数中 CCHLO 的结果相较于 HLO 的结果都要好。因此，CCLO 算子对算法的影响可以归纳为一下几点：

（1）对于适应度曲线，可以明显观察到迭代结果都要优于标准 HLO，在大多数情况下，CCHLO 算法在前期的收敛性都要大于标准 HLO。这表明，在相同的测试环境中，CCHLO 算法比 HLO 算法能够更快的找到全局最优解，意味着在较短的时间内可以得到较准确的结果，可以节省计算资源和时间，提高系统的实时性和响应速度，以达到减少系统资源消耗。

（2）对于 AD 曲线，可以明显看出，CCHLO 算法在迭代后期，种群中的个体具有比 HLO 更大的曲线波动。当种群个体分布较为发散时，表明算法有陷入局部最优趋势，从而触发重新学习机制降低了种群密度，使算法中的部分个体跳出了局部，结合竞争与合作学习算子的机制，当跳出局部的个体找到了更好的学习目标，将会带领种群跳出当前位置，达到跳出局部的作用。

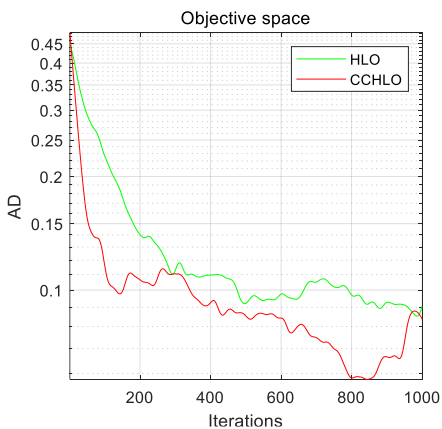


(a) F6 函数中的 AD 曲线

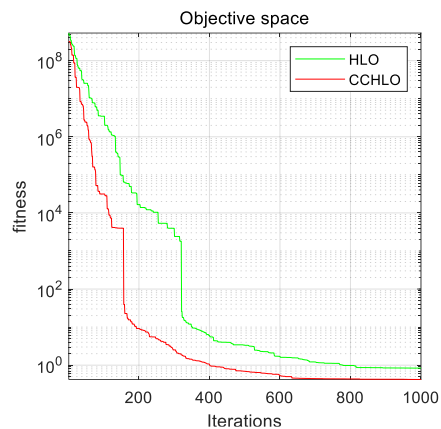


(b) F6 函数中的适应度曲线

图 3-2 30 维 F6 的平均距离曲线与适应度曲线

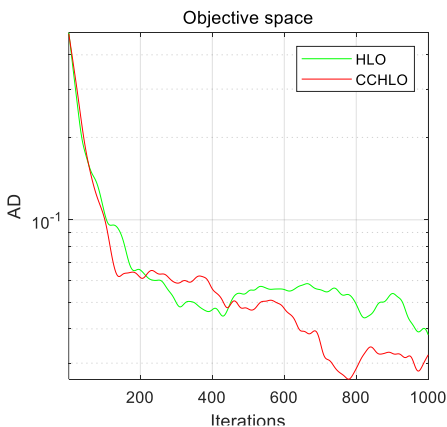


(a) F12 函数中的 AD 曲线

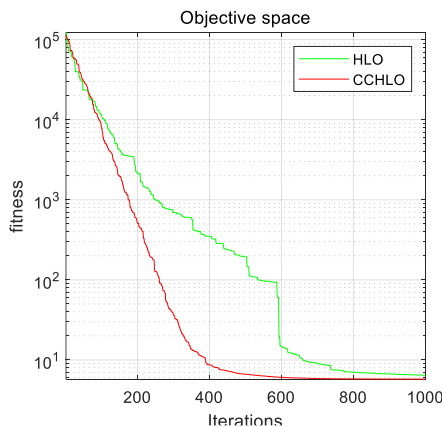


(b) F12 函数中的适应度曲线

图 3-3 30 维 F12 的平均距离曲线与适应度曲线



(a) F6 函数中的 AD 曲线



(b) F6 函数中的适应度曲线

图 3-4 50 维 F6 的平均距离曲线与适应度曲线

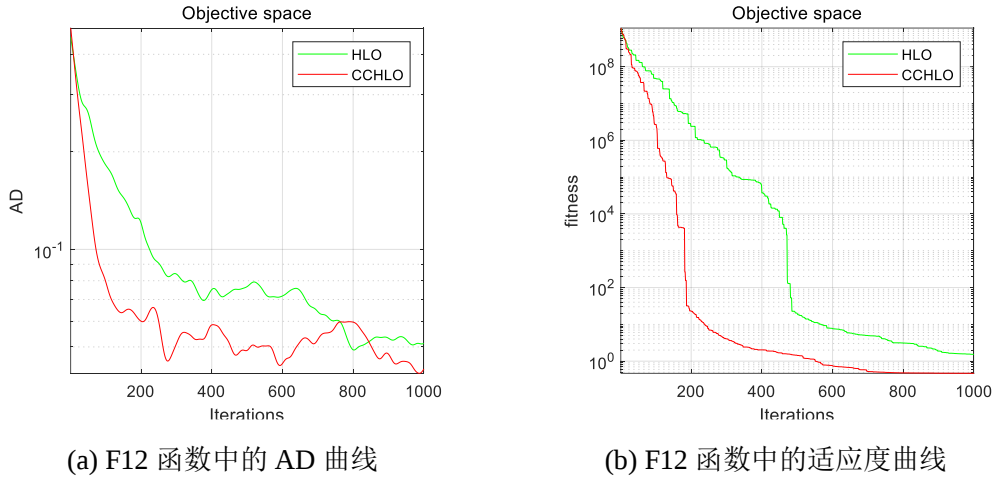


图 3-5 50 维 F12 的平均距离曲线与适应度曲线

观察图 3-2、图 3-3、图 3-4 和图 3-5 可知, CCHLO 算法在 AD 曲线中比 HLO 算法下降的速率更快, 并且 CCHLO 算法在适应度曲线中的下降速率也比 HLO 算法更快, 说明 CCHLO 算法能够更迅速地找到优化问题的优质解, 并在较短时间内取得较低的适应度值。其较快的下降速率可能反映了 CCHLO 算法更强的全局搜索能力和更高效的局部优化能力。并且, 在适应度值不变时个体间的平均距离有所提升, 表明算法在保持当前适应度值不变的同时, 通过扩展搜索空间, 避免陷入局部最优解, 从而优化性能并提高求解准确度。

### 3.3 增强探索与开采平衡的合作竞争人类学习优化算法

#### 3.3.1 增强探索与开采平衡的合作竞争模型

考虑到具有合作竞争机制的人类学习优化算法中的合作竞争学习算子与社会学习算子对种群信息的利用效率差异, 使得算法在面对不同问题的优化阶段时, 无法有效地动态调整开发与探索的平衡, 进而影响搜索效率和最优解的精度。对此, 通过合作竞争机制中引入增强探索与开采平衡策略, 研究了增强探索与开采平衡的合作竞争人类学习优化算法 (Enhanced Cooperative-Competitive Human Learning Optimization, ECCHLO), 根据问题的特性和不同优化阶段的需求, 自适应调整合作竞争机制, 增强了算法的探索与开采平衡, 避免陷入局部最优解, 从而提升整体优化性能。

增强探索与开采平衡的合作竞争模型的实现, 是通过调整参数使得算法随着迭代的进行而动态调整合作竞争算子, 如公式 (3.16):

$$pic = \begin{cases} pic_{min} & \text{if } t \leq T_a \\ pic_{min} + s \times (pic_{max} - pic_{min}) & \text{if } T_a < t \leq T_b \\ pic_{max} & \text{else} \end{cases} \quad (3.16)$$

其中,  $pic$  为所设计模型中动态调整的参数,  $pic_{max}$  为算子变化的上限,  $pic_{min}$  为算子变化的下限, 通过公式 (3.16) 使得  $pic$  的取值在  $[0, T_a]$  时取为  $pic_{min}$ , 在  $[T_a, T_b]$  时的取值为  $pic_{max}$ ,  $pic$  具体变化情况如图 3-6 所示:

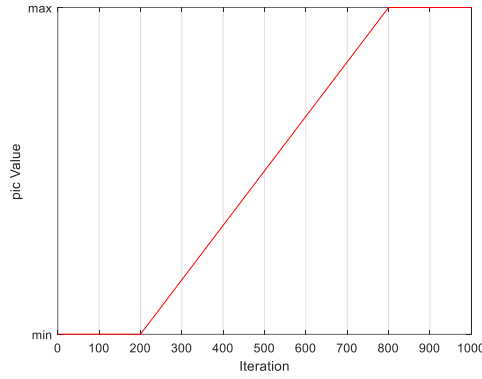


图 3-6 增强探索与开采平衡合作竞争模型的  $pic$  变化曲线

$T_a$  和  $T_b$  为迭代次数中设立的阈值设为  $0.2T_{max}$  和  $0.8T_{max}$ ,  $s$  的表达公式如下:

$$s = \frac{(t - T_a)}{(T_b - T_a)} \quad (3.17)$$

其中,  $s$  为当迭代次数达到  $T_a$  时,  $s$  将随着迭代次数的变化而变化。通过公式 (3.16) 可知, 增强探索与开采平衡合作竞争模型能够在达到一定迭代次数后, 根据时间的变化进行自适应调整。这一模型使得的合作竞争算子能够随着优化迭代过程逐渐变化, 使得改进算法能够更好地适应不同阶段的需求, 以实现算法提升探索与开采间的平衡。

### 3.3.2 ECCHLO 算法实现

ECCHLO 算法是通过在基本 HLO 算法的基础上增加了增强探索与开采平衡合作竞争模型, 如公式 (3.19):

$$x_{ij} = \begin{cases} RLO & 0 \leq r_{14} \leq pr \\ ik_{ip,j} & pr \leq r_{14} \leq pi \\ CCLO & pi \leq r_{14} \leq pic \\ sk_{qj} & \text{else} \end{cases} \quad (3.19)$$

其中,  $(pr - 0)$  为算法执行随机学习算子的概率,  $(pi - pr)$  为算法执行个体学习算子的概率,  $(pic - pi)$  为算法执行增强探索与开采平衡合作竞争模型的概率,  $(1 - pic)$  为执行社会学习算子的概率。增强探索与开采平衡合作竞争模型通过保证参数的

灵活性，确保了信息的质量和有效性。灵活的参数使得个体能够根据当前的优化阶段动态调整合作竞争，更高效地获取全局信息。通过增强信息流动和交互效率，个体能够在全局范围内快速识别潜在的优质解，避免局部最优解的困境，进而加速搜索过程并能够显著提升算法的优化性能。ECCHLO 的具体流程图如图 3-7 所示：

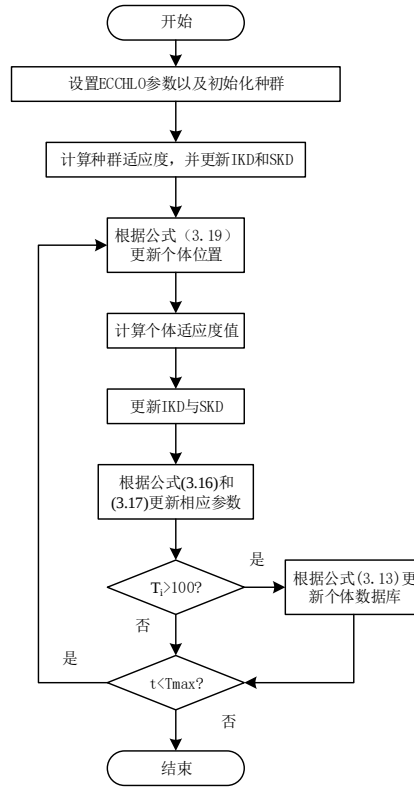


图 3-7 ECCHLO 流程图

### 3.3.3 ECCHLO 算法参数分析

ECCHLO 算法通过有效调整合作竞争学习算子和利用个体知识库中的优秀个体，实现增强算法探索与开采平衡。 $pic$  的变化情况作为一个关键参数，其自适应调整过程对算法性能的提升具有重要影响。因此，需要深入研究  $pic$  的取值范围（即  $pic_{min}$  和  $pic_{max}$  的取值），以便精确控制合作与竞争的动态变化。 $pic_{min}$  和  $pic_{max}$  的值将直接影响增强探索与开采平衡合作竞争模型的调整策略，并且在优化过程中发挥着重要作用。对于其他参数的取值，在最大迭代次数为  $T_{max}$  时  $T_a$  和  $T_b$  的取值为  $T_{max}/5$  和  $4/5 \times T_{max}$ 。对于  $T_a$  取为  $T_{max}/5$ ，在算法初期阶段， $pic$  的取值保持为  $pic_{min}$  可以确保算法探索阶段的稳定性和有效性，避免过早收敛。随着迭代次数的增加，算子值逐渐变化，可以逐步过渡到更精细的局部搜索，增强算

法的探索能力。对于  $T_b$  取为  $4/5 \times T_{max}$ ，在算法迭代后期，设置  $pic$  保持为  $pic_{max}$ ，能够确保算法在达到较高迭代次数时维持较强的收敛性，避免过度搜索造成的效率损失。在参数设置完成后， $pic_{min}$  和  $pic_{max}$  的取值情况如表 3-5，各参数组合的算法性能测试结果如表 3-6。

表 3-5 ECCHLO 参数取值情况

| 参数          | 取值情况 |      |      |      |      |      |
|-------------|------|------|------|------|------|------|
|             | 取值 1 | 取值 2 | 取值 3 | 取值 4 | 取值 5 | 取值 6 |
| $pic_{min}$ | 0.83 | 0.84 | 0.85 | 0.86 | 0.87 | 0.88 |
| $pic_{max}$ | 0.89 | 0.90 | 0.91 | 0.92 | 0.93 | 0.94 |

表 3-6 各维度参数正交表

| 序号 | 参数          |             | F6              | F12             | F6              | F12             |
|----|-------------|-------------|-----------------|-----------------|-----------------|-----------------|
|    | $pic_{min}$ | $pic_{max}$ | D=30            |                 | D=50            |                 |
| 1  | 0.83        | 0.89        | 3.85E+00        | 5.04E-01        | <b>7.18E+00</b> | <b>7.05E-01</b> |
| 2  | 0.83        | 0.90        | 3.76E+00        | 4.62E-01        | <b>7.32E+00</b> | <b>7.08E-01</b> |
| 3  | 0.83        | 0.91        | 3.78E+00        | 7.33E-01        | <b>7.21E+00</b> | <b>7.29E-01</b> |
| 4  | 0.83        | 0.92        | 4.16E+00        | 7.16E-01        | <b>7.57E+00</b> | <b>9.18E-01</b> |
| 5  | 0.83        | 0.93        | 4.16E+00        | 6.46E-01        | <b>7.70E+00</b> | <b>1.16E+00</b> |
| 6  | 0.83        | 0.94        | 3.87E+00        | 6.45E-01        | <b>7.71E+00</b> | 1.17E+00        |
| 7  | 0.84        | 0.89        | 3.31E+00        | 4.96E-01        | <b>8.02E+00</b> | 1.43E+00        |
| 8  | 0.84        | 0.90        | <b>3.16E+00</b> | <b>4.22E-01</b> | <b>8.35E+00</b> | <b>1.14E+00</b> |
| 9  | 0.84        | 0.91        | 3.30E+00        | 5.40E-01        | 9.51E+00        | <b>1.17E+00</b> |
| 10 | 0.84        | 0.92        | <b>3.22E+00</b> | 4.91E-01        | 9.04E+00        | 1.58E+00        |
| 11 | 0.84        | 0.93        | 3.28E+00        | 4.84E-01        | 9.07E+00        | 1.40E+00        |
| 12 | 0.84        | 0.94        | <b>3.18E+00</b> | <b>3.84E-01</b> | 9.01E+00        | 1.31E+00        |
| 13 | 0.85        | 0.89        | <b>3.22E+00</b> | <b>4.11E-01</b> | 9.70E+00        | 1.40E+00        |
| 14 | 0.85        | 0.90        | 3.30E+00        | 4.37E-01        | 9.72E+00        | 1.57E+00        |
| 15 | 0.85        | 0.91        | 3.42E+00        | 4.47E-01        | 9.18E+00        | 1.22E+00        |
| 16 | 0.85        | 0.92        | <b>3.16E+00</b> | <b>3.83E-01</b> | 9.26E+00        | 1.54E+00        |
| 17 | 0.85        | 0.93        | 3.31E+00        | 4.78E-01        | 9.28E+00        | <b>1.15E+00</b> |
| 18 | 0.85        | 0.94        | <b>3.24E+00</b> | 4.28E-01        | 1.00E+01        | <b>1.09E+00</b> |
| 19 | 0.86        | 0.89        | 3.40E+00        | <b>4.03E-01</b> | 9.61E+00        | 1.28E+00        |
| 20 | 0.86        | 0.90        | <b>3.17E+00</b> | 4.67E-01        | 9.59E+00        | 1.32E+00        |
| 21 | 0.86        | 0.91        | 3.42E+00        | <b>4.13E-01</b> | 9.10E+00        | 1.37E+00        |
| 22 | 0.86        | 0.92        | 3.40E+00        | <b>4.24E-01</b> | 9.09E+00        | 1.41E+00        |
| 23 | 0.86        | 0.93        | 3.50E+00        | 4.34E-01        | 8.66E+00        | 1.50E+00        |
| 24 | 0.86        | 0.94        | 3.40E+00        | 4.25E-01        | 8.66E+00        | 1.46E+00        |
| 25 | 0.87        | 0.89        | 3.39E+00        | 5.15E-01        | 9.78E+00        | 1.34E+00        |
| 26 | 0.87        | 0.90        | 3.56E+00        | 4.68E-01        | 9.08E+00        | 1.54E+00        |
| 27 | 0.87        | 0.91        | 3.52E+00        | 4.99E-01        | <b>8.35E+00</b> | 1.22E+00        |

续表：3-6

|    |      |      |                 |                 |                 |                 |
|----|------|------|-----------------|-----------------|-----------------|-----------------|
| 28 | 0.87 | 0.92 | <b>3.20E+00</b> | <b>3.64E-01</b> | 9.24E+00        | 1.27E+00        |
| 29 | 0.87 | 0.93 | <b>3.04E+00</b> | <b>3.62E-01</b> | 9.40E+00        | 1.41E+00        |
| 30 | 0.87 | 0.94 | 3.39E+00        | 4.52E-01        | <b>8.24E+00</b> | 1.29E+00        |
| 31 | 0.88 | 0.89 | 3.49E+00        | <b>3.97E-01</b> | 1.24E+01        | <b>1.17E+00</b> |
| 32 | 0.88 | 0.90 | 3.38E+00        | 4.99E-01        | 8.63E+00        | 1.50E+00        |
| 33 | 0.88 | 0.91 | 3.26E+00        | 4.77E-01        | 9.85E+00        | 1.40E+00        |
| 34 | 0.88 | 0.92 | 3.43E+00        | 5.00E-01        | 9.76E+00        | 1.50E+00        |
| 35 | 0.88 | 0.93 | 3.24E+00        | 4.50E-01        | 8.80E+00        | 1.38E+00        |
| 36 | 0.88 | 0.94 | <b>3.20E+00</b> | 4.28E-01        | 9.36E+00        | 1.37E+00        |

通过对表 3-6 中同一函数下相同求解空间维度的排名前十数据进行加粗，进一步分析加粗数据的数量，能够有效地识别出最优的参数组合。通过对表中参数的分析，结果显示，当参数  $pic_{min}$  和  $pic_{max}$  取为 0.84 和 0.90 时，算法性能达到最优，这一参数组合显著提升了算法的收敛速度和解的精度，同时增强了全局搜索能力与局部探索能力的平衡。

### 3.4 仿真实验与结果分析

为了具体验证 CCHLO 算法与 ECCHLO 算法性能，将算法应用于 CEC2005 基准函数中，与其他对比算法进行对比验证，即人类学习优化算法、二进制粒子群优化算法（Binary Particle Swarm Optimization Algorithm, BPSO）<sup>[41]</sup>、二进制差分进化算法（Binary Differential Evolution Algorithm, BDE）<sup>[50]</sup>、二进制蝙蝠算法（Binary Bat Algorithm, BBA）<sup>[51]</sup>进行对比。

算法的参数设定如表 3-7。为了防止偶然性的出现，所有算法均独立运行 50 次求取平均值中间值以及最差值进行对比，种群数量设为 50，并且对低维度 30 维以及高维度 100 维进行不同维度的结果比较。

表 3-7 各算法参数设定值

| 算法                   | 参数                                                                 |
|----------------------|--------------------------------------------------------------------|
| CCHLO                | $G = 5, pr = 5 / M, pi = 0.82, pc = 0.93$                          |
| ECCHLO               | $G = 5, pr = 5 / M, pi = 0.82, pic_{min} = 0.84, pic_{max} = 0.90$ |
| HLO <sup>[21]</sup>  | $pr = 5 / M, pi = 0.85 + 2 / M$                                    |
| BPSO <sup>[41]</sup> | $w = 0.8, c_1 = c_2 = 2, V_{max} = 4$                              |
| BDE <sup>[51]</sup>  | $F = 0.6, CR = 0.5$                                                |
| BBA <sup>[52]</sup>  | $r_t \in [0, 1], A_t \in [0, 2]$                                   |

### 3.4.1 低维函数测试结果

将算法在低维度函数中进行比较。将独立运行 50 次的平均值、中间值、最差值记录于表 3-8, 基准函数维度按照表 2-5, 在每个基准函数中的测试结果通过平均值进行排名。

表 3-8 的实验结果表明, CCHLO 算法与 ECCHLO 算法在低维函数的优化问题中表现出较其他算法更为优异的总体结果。在多个基准函数测试中, 改进算法在某些特定问题上的性能显著超越了对比算法。从 Rank 的结果来看, CCHLO 算法与 ECCHLO 算法在低维函数中的总体表现始终优于其他现有算法, 这表明该算法在低维优化问题中的全局搜索能力和精度具有较为明显的优势。结果验证了所提出算法在低维问题中高效的优化能力, 能够更好地平衡探索与开发。

表 3-8 30 维空间维度中各算法收敛精度统计

| 函数  | 指标    | BPSO            | BDE             | BBA       | HLO              | CCHLO           | ECCHLO          |
|-----|-------|-----------------|-----------------|-----------|------------------|-----------------|-----------------|
| F1  | Ave   | 8.64E-01        | 7.53E-02        | 1.53E-01  | 1.60E-05         | 4.09E-07        | <b>1.69E-07</b> |
|     | Med   | 8.53E-01        | 6.89E-02        | 2.53E-01  | 7.52E-07         | 2.00E-07        | <b>8.60E-08</b> |
|     | Worst | 9.45E-01        | 8.75E-02        | 3.56E-01  | 2.30E-04         | 4.72E-06        | <b>1.22E-06</b> |
|     | Rank  | 6               | 4               | 5         | 3                | 2               | 1               |
| F2  | Ave   | 4.85E+01        | 2.36E+01        | 1.86E+00  | 7.38E-05         | 5.57E-05        | <b>3.20E-05</b> |
|     | Med   | 8.27E-01        | 1.35E+00        | 9.42E-01  | 5.96E-05         | 5.13E-05        | <b>2.94E-05</b> |
|     | Worst | 3.05E+02        | 1.42E+02        | 1.52E+01  | 2.42E-04         | 1.75E-04        | <b>6.81E-05</b> |
|     | Rank  | 6               | 5               | 4         | 3                | 2               | 1               |
| F3  | Ave   | 6.42E+02        | <b>1.52E+02</b> | 2.41E+04  | 9.48E+03         | 9.79E+03        | 1.17E+04        |
|     | Med   | 6.53E+02        | <b>6.25E+00</b> | 2.36E+04  | 9.53E+03         | 1.03E+04        | 1.10E+04        |
|     | Worst | <b>1.62E+03</b> | 5.73E+03        | 3.72E+04  | 1.54E+04         | 1.62E+04        | 2.29E+04        |
|     | Rank  | 2               | 1               | 6         | 3                | 4               | 5               |
| F4  | Ave   | 3.52E+00        | 1.75E+00        | 2.63E+01  | 6.16E+00         | 2.93E+00        | <b>1.25E+00</b> |
|     | Med   | 1.64E+00        | 1.64E+00        | 2.36E+01  | 6.09E+00         | 1.79E+00        | <b>7.83E-01</b> |
|     | Worst | 3.53E+00        | <b>3.52E+00</b> | 4.74E+01  | 1.88E+01         | 1.12E+01        | 5.25E+00        |
|     | Rank  | 4               | 2               | 6         | 5                | 3               | 1               |
| F5  | Ave   | 4.52E+02        | <b>1.53E+02</b> | 1.35E+03  | 4.52E+02         | 3.05E+02        | 3.64E+02        |
|     | Med   | 8.32E+01        | <b>7.43E+01</b> | 1.54E+03  | 2.68E+02         | 1.17E+02        | 1.31E+02        |
|     | Worst | 6.43E+02        | <b>4.63E+02</b> | 2.41E+03  | 2.48E+03         | 1.54E+03        | 1.86E+03        |
|     | Rank  | 5               | 1               | 6         | 4                | 2               | 3               |
| F6  | Ave   | 1.28E-01        | <b>2.62E-02</b> | 1.62E+01  | 3.93E+00         | 3.85E+00        | 3.83E+00        |
|     | Med   | 1.95E-01        | <b>2.36E-02</b> | 1.32E+01  | 3.82E+00         | 3.94E+00        | 3.93E+00        |
|     | Worst | 5.57E-01        | <b>5.85E-02</b> | 4.61E+01  | 5.62E+00         | 4.77E+00        | 5.27E+00        |
|     | Rank  | 2               | 1               | 6         | 5                | 4               | 3               |
| F7  | Ave   | 1.52E-01        | 1.27E-01        | 1.36E-01  | 1.08E-01         | <b>9.44E-02</b> | 1.04E-01        |
|     | Med   | 9.37E-03        | 1.73E-02        | 1.74E-01  | 9.44E-02         | <b>8.50E-02</b> | 9.83E-02        |
|     | Worst | 2.53E+00        | 5.35E+00        | 3.73E-01  | 3.00E-01         | <b>2.12E-01</b> | 2.59E-01        |
|     | Rank  | 6               | 4               | 5         | 3                | 1               | 2               |
| F8  | Ave   | -7341.53        | -9247.35        | -10363.12 | <b>-11668.45</b> | -11467.55       | -11503.42       |
|     | Med   | -7710.25        | -7838.24        | -10263.15 | <b>-11737.96</b> | -11461.13       | -11504.42       |
|     | Worst | -6053.62        | -4204.35        | -9643.32  | <b>-10822.51</b> | -10805.97       | -10809.61       |
|     | Rank  | 6               | 5               | 4         | 1                | 3               | 2               |
| F9  | Ave   | 5.26E+01        | 6.37E+01        | 4.76E+01  | <b>3.16E+01</b>  | 3.76E+01        | 3.58E+01        |
|     | Med   | 5.63E+01        | 5.26E+01        | 4.36E+01  | <b>3.13E+01</b>  | 3.74E+01        | 3.47E+01        |
|     | Worst | 1.36E+02        | 1.72E+02        | 6.95E+01  | 5.57E+01         | <b>5.24E+01</b> | 6.28E+01        |
|     | Rank  | 5               | 6               | 4         | 1                | 3               | 2               |
| F10 | Ave   | 3.35E+00        | 2.25E+00        | 4.25E+00  | 1.70E+00         | 1.64E+00        | <b>1.64E+00</b> |
|     | Med   | 3.63E+00        | 2.52E+00        | 3.52E+00  | 2.01E+00         | 1.90E+00        | <b>1.81E+00</b> |
|     | Worst | 5.35E+00        | 4.65E+00        | 1.15E+01  | 2.75E+00         | 2.66E+00        | <b>2.50E+00</b> |
|     | Rank  | 5               | 4               | 6         | 3                | 2               | 1               |
| F11 | Ave   | <b>1.24E-02</b> | 1.53E-02        | 1.51E-01  | 5.97E-02         | 5.73E-02        | 5.57E-02        |
|     | Med   | 1.35E-02        | <b>7.34E-03</b> | 1.32E-01  | 2.95E-02         | 2.27E-02        | 2.46E-02        |
|     | Worst | <b>3.24E-02</b> | 3.32E-02        | 1.51E-01  | 7.34E-01         | 5.72E-01        | 3.74E-01        |

续表: 3-8

|         | Rank  | 1                | 2                | 6                | 5                | 4                | 3                |
|---------|-------|------------------|------------------|------------------|------------------|------------------|------------------|
| F12     | Ave   | 5.41E+00         | 3.53E+00         | 6.24E-01         | 6.63E-01         | 6.14E-01         | <b>4.34E-01</b>  |
|         | Med   | 4.67E+00         | 2.43E+00         | <b>6.62E-01</b>  | 5.06E-01         | 5.04E-01         | 5.52E-01         |
|         | Worst | 3.29E+01         | 9.26E+00         | 6.63E+00         | 3.73E+00         | 1.40E+00         | <b>1.28E+00</b>  |
|         | Rank  | 6                | 5                | 3                | 4                | 2                | 1                |
| F13     | Ave   | <b>9.15E-01</b>  | 9.35E-01         | 2.62E+00         | 1.82E+00         | 1.76E+00         | 1.72E+00         |
|         | Med   | 6.75E-01         | <b>5.74E-01</b>  | 2.47E+00         | 1.84E+00         | 1.78E+00         | 1.67E+00         |
|         | Worst | 3.36E-00         | 5.63E+00         | 4.55E+00         | <b>2.42E+00</b>  | 2.72E+00         | 2.74E+00         |
|         | Rank  | 1                | 2                | 6                | 5                | 4                | 3                |
| F14     | Ave   | 1.15E+00         | 1.05E+00         | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  |
|         | Med   | 9.98E-01         | 9.98E-01         | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  |
|         | Worst | 4.63E+00         | 1.36E+00         | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  | <b>9.98E-01</b>  |
|         | Rank  | 6                | 5                | 1                | 1                | 1                | 1                |
| F15     | Ave   | 9.67E-03         | 5.26E-03         | 9.63E-04         | 7.35E-04         | 7.43E-04         | <b>7.15E-04</b>  |
|         | Med   | 2.47E-03         | 1.62E-03         | 8.24E-04         | <b>7.15E-04</b>  | 7.42E-04         | 7.25E-04         |
|         | Worst | 4.67E-02         | 2.34E-02         | 1.36E-03         | <b>1.35E-03</b>  | 1.39E-03         | 1.49E-03         |
|         | Rank  | 5                | 6                | 4                | 2                | 3                | 1                |
| F16     | Ave   | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> |
|         | Med   | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> |
|         | Worst | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> | <b>-1.03E+00</b> |
|         | Rank  | 1                | 1                | 1                | 1                | 1                | 1                |
| F17     | Ave   | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  |
|         | Med   | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  |
|         | Worst | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  | <b>3.98E-01</b>  |
|         | Rank  | 1                | 1                | 1                | 1                | 1                | 1                |
| F18     | Ave   | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  |
|         | Med   | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  |
|         | Worst | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  | <b>3.00E+00</b>  |
|         | Rank  | 1                | 1                | 1                | 1                | 1                | 1                |
| F19     | Ave   | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> |
|         | Med   | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> |
|         | Worst | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> | <b>-3.86E+00</b> |
|         | Rank  | 1                | 1                | 1                | 1                | 1                | 1                |
| F20     | Ave   | -3.24E+00        | -3.26E+00        | -3.26E+00        | <b>-3.32E+00</b> | -3.30E+00        | -3.29E+00        |
|         | Med   | -3.25E+00        | -3.17E+00        | -3.23E+00        | <b>-3.32E+00</b> | <b>-3.32E+00</b> | <b>-3.32E+00</b> |
|         | Worst | -2.81E+00        | -2.63E+00        | <b>-3.27E+00</b> | -3.25E+00        | -3.20E+00        | -3.20E+00        |
|         | Rank  | 6                | 4                | 5                | 1                | 2                | 3                |
| F21     | Ave   | -9.19E+00        | -8.94E+00        | <b>-1.00E+01</b> | -7.91E+00        | -8.11E+00        | -8.16E+00        |
|         | Med   | <b>-1.02E+01</b> | <b>-1.02E+01</b> | <b>-1.02E+01</b> | -9.59E+00        | -9.78E+00        | -9.78E+00        |
|         | Worst | -2.63E+00        | -2.63E+00        | <b>-9.52E+00</b> | -3.86E+00        | -2.68E+00        | -4.73E+00        |
|         | Rank  | 2                | 3                | 1                | 6                | 5                | 4                |
| F22     | Ave   | -9.63E+00        | -9.02E+00        | <b>-1.04E+01</b> | -9.23E+00        | -9.03E+00        | -8.59E+00        |
|         | Med   | -1.04E+01        | -1.04E+01        | <b>-1.04E+01</b> | -1.04E+01        | -1.02E+01        | -1.00E+01        |
|         | Worst | -2.77E+00        | -1.84E+00        | <b>-1.01E+01</b> | -4.76E+00        | -4.74E+00        | -4.76E+00        |
|         | Rank  | 2                | 4                | 1                | 3                | 5                | 6                |
| F23     | Ave   | -8.62E+01        | -1.01E+01        | <b>-1.05E+01</b> | -8.23E+00        | -9.26E+00        | -9.04E+00        |
|         | Med   | -1.62E+01        | -1.05E+01        | <b>-1.05E+01</b> | -1.01E+01        | -1.03E+01        | -1.05E+01        |
|         | Worst | -5.63E+00        | -2.43E+00        | <b>-1.05E+01</b> | -3.82E+00        | -4.81E+00        | -2.87E+00        |
|         | Rank  | 5                | 2                | 1                | 6                | 3                | 4                |
| AvgRank |       | 3.6957           | 3.0435           | 3.6522           | 2.9565           | 2.5652           | 2.2174           |
| Rank    |       | 6                | 4                | 5                | 3                | 2                | 1                |

### 3.4.2 高维函数测试结果

在高维度函数环境中,数据点之间的距离会变得更加稀疏,从而使得算法在寻找最优解或者进行分类时变得更加困难。比如,数据点之间的距离变得分散,将使得传统的距离度量方法变得不再有效,导致算法性能下降。并且,高维度数据会增加计算的复杂度,使得算法的运行时间变得更长,甚至导致算法无法处理大规模高维度数据。因此,这对所有算法的性能都将是个挑战。因此,在高维度

函数环境中测试算法性能也是相对必要的。将 CCHLO 算法和 ECCHLO 与 BPSO、BDE、BBA、HLO 算法在 CEC2005 的前 13 个基准函数中进行 100 维的高维度对比。结果如表 3-9。

从测试结果可以看出，CCHLO 算法和 ECCHLO 算法在绝大多数实验中均展现出优于其他对比算法的性能。在多个基准函数测试中，两种改进算法在解的

表 3-9 100 维空间维度中各算法收敛精度统计

| 函数      | 指标    | BPSO      | BDE       | BBA              | HLO             | CCHLO           | ECCHLO          |
|---------|-------|-----------|-----------|------------------|-----------------|-----------------|-----------------|
| F1      | Ave   | 4.52E+02  | 7.35E+01  | 4.32E+03         | 8.38E-01        | 6.54E-01        | <b>4.07E-01</b> |
|         | Med   | 3.32E+02  | 7.83E+01  | 2.15E+03         | <b>6.71E-01</b> | 6.07E-01        | 3.71E-01        |
|         | Worst | 4.63E+02  | 9.26E+01  | 1.23E+04         | 2.91E+00        | 1.36E+00        | <b>9.31E-01</b> |
|         | Rank  | 5         | 4         | 6                | 3               | 2               | 1               |
| F2      | Ave   | 2.51E+02  | 2.24E+02  | 1.17E+00         | 2.43E-01        | 2.32E-01        | <b>1.80E-01</b> |
|         | Med   | 2.25E+02  | 2.22E+02  | 6.79E-01         | 2.21E-01        | 2.37E-01        | <b>1.75E-01</b> |
|         | Worst | 2.36E+02  | 3.04E+02  | 4.08E+00         | 3.27E-01        | 3.19E-01        | <b>2.62E-01</b> |
|         | Rank  | 6         | 5         | 4                | 3               | 2               | 1               |
| F3      | Ave   | 8.52E+05  | 4.11E+05  | 3.69E+05         | 1.63E+05        | 1.56E+05        | <b>1.56E+05</b> |
|         | Med   | 8.36E+05  | 4.21E+05  | 3.61E+05         | 1.64E+05        | 1.64E+05        | <b>1.62E+05</b> |
|         | Worst | 9.75E+05  | 4.59E+05  | 4.49E+05         | 2.09E+05        | 2.07E+05        | <b>2.02E+05</b> |
|         | Rank  | 6         | 5         | 4                | 3               | 2               | 1               |
| F4      | Ave   | 2.25E+02  | 2.26E+02  | 3.44E+02         | 4.38E+01        | 4.17E+01        | <b>4.04E+01</b> |
|         | Med   | 2.47E+02  | 2.23E+02  | 3.46E+02         | 4.31E+01        | 4.16E+01        | <b>3.96E+01</b> |
|         | Worst | 2.25E+02  | 2.74E+02  | 3.05E+02         | 5.05E+01        | 5.02E+01        | <b>5.00E+01</b> |
|         | Rank  | 4         | 5         | 6                | 2               | 3               | 1               |
| F5      | Ave   | 3.24E+04  | 3.26E+05  | 3.81E+06         | 5.53E+03        | <b>9.44E+02</b> | 1.30E+03        |
|         | Med   | 3.46E+04  | 4.53E+05  | 3.34E+06         | 1.33E+03        | <b>5.28E+02</b> | 1.17E+03        |
|         | Worst | 3.25E+04  | 4.44E+05  | 8.80E+06         | 7.32E+04        | <b>3.97E+03</b> | 4.87E+03        |
|         | Rank  | 4         | 5         | 6                | 3               | 1               | 2               |
| F6      | Ave   | 4.15E+02  | 5.55E+01  | 4.35E+02         | 1.55E+01        | 1.49E+01        | <b>1.44E+01</b> |
|         | Med   | 4.35E+02  | 5.31E+01  | 2.84E+02         | 1.54E+01        | 1.51E+01        | <b>1.40E+01</b> |
|         | Worst | 4.36E+02  | 6.42E+01  | 2.25E+03         | 1.92E+01        | 1.77E+01        | <b>1.74E+01</b> |
|         | Rank  | 5         | 4         | 6                | 3               | 2               | 1               |
| F7      | Ave   | 1.26E+01  | 1.25E+00  | 5.03E+00         | 7.36E-01        | 5.85E-01        | <b>5.21E-01</b> |
|         | Med   | 1.43E+01  | 1.35E+00  | 4.08E+00         | 6.57E-01        | 5.76E-01        | <b>5.19E-01</b> |
|         | Worst | 2.14E+01  | 1.46E+00  | 1.26E+01         | 1.62E+00        | 1.00E+00        | <b>7.62E-01</b> |
|         | Rank  | 6         | 4         | 5                | 3               | 2               | 1               |
| F8      | Ave   | -3.63E+04 | -5.25E+04 | <b>-7.25E+03</b> | -3.47E+04       | -3.42E+04       | -3.45E+04       |
|         | Med   | -3.35E+04 | -5.23E+04 | <b>-7.35E+03</b> | -3.45E+04       | -3.43E+04       | -3.45E+04       |
|         | Worst | -2.25E+04 | -5.57E+04 | <b>-6.46E+03</b> | -3.32E+04       | -3.27E+04       | -3.26E+04       |
|         | Rank  | 5         | 6         | 1                | 4               | 2               | 3               |
| F9      | Ave   | 4.12E+02  | 5.47E+02  | 2.29E+02         | 2.15E+02        | 2.10E+02        | <b>2.01E+02</b> |
|         | Med   | 4.52E+02  | 5.44E+02  | 1.97E+02         | 2.07E+02        | 2.08E+02        | <b>1.97E+02</b> |
|         | Worst | 4.57E+02  | 5.98E+02  | 5.30E+02         | 2.56E+02        | <b>2.47E+02</b> | 2.55E+02        |
|         | Rank  | 5         | 6         | 4                | 3               | 2               | 1               |
| F10     | Ave   | 4.40E+00  | 4.03E+00  | 1.92E+01         | 2.38E+00        | 2.32E+00        | <b>2.28E+00</b> |
|         | Med   | 4.26E+00  | 4.13E+00  | 2.06E+01         | 2.33E+00        | <b>2.30E+00</b> | 2.35E+00        |
|         | Worst | 4.79E+00  | 5.12E+00  | 2.06E+01         | 2.59E+00        | 2.61E+00        | <b>2.59E+00</b> |
|         | Rank  | 5         | 4         | 6                | 3               | 2               | 1               |
| F11     | Ave   | 8.25E-01  | 4.46E-01  | 4.35E+01         | 3.87E-01        | 3.19E-01        | <b>1.96E-01</b> |
|         | Med   | 8.30E-01  | 4.36E-01  | 3.75E+01         | 3.47E-01        | 3.10E-01        | <b>1.80E-01</b> |
|         | Worst | 8.35E-01  | 5.35E-01  | 1.24E+02         | 1.11E+00        | 5.62E-01        | <b>4.61E-01</b> |
|         | Rank  | 5         | 4         | 6                | 3               | 2               | 1               |
| F12     | Ave   | 2.31E+01  | 1.35E+01  | 1.61E+01         | 1.27E+00        | 9.55E-01        | <b>8.85E-01</b> |
|         | Med   | 2.27E+01  | 1.25E+01  | 1.59E+01         | 1.18E+00        | 8.82E-01        | <b>8.21E-01</b> |
|         | Worst | 3.27E+01  | 1.94E+01  | 3.29E+01         | 2.74E+00        | 3.92E+01        | <b>1.66E+00</b> |
|         | Rank  | 6         | 4         | 5                | 3               | 2               | 1               |
| F13     | Ave   | 4.99E+01  | 1.52E+01  | 1.34E+01         | 8.86E+00        | 7.88E+00        | <b>7.34E+00</b> |
|         | Med   | 1.57E+01  | 1.35E+01  | 1.63E+01         | 8.34E+00        | 7.78E+00        | <b>7.29E+00</b> |
|         | Worst | 1.64E+02  | 1.36E+01  | 3.36E+01         | 1.80E+01        | 1.07E+01        | <b>8.70E+00</b> |
|         | Rank  | 6         | 5         | 4                | 3               | 2               | 1               |
| AvgRank |       | 5.2308    | 4.6923    | 4.8462           | 3.0000          | 2.0000          | 1.2308          |
| Rank    |       | 6         | 4         | 5                | 3               | 2               | 1               |

质量和稳定性方面均超越了对比算法。并且,通过 Rank 数据排名情况可以看出 CCHLO 算法和 ECCHLO 算法的整体表现情况要优于其他对比算法。因此,通过 CEC2005 测试集的测试结果可以验证 CCHLO 算法和 ECCHLO 算法在处理复杂优化问题时,能够有效平衡全局搜索和局部开发,从而避免陷入局部最优解,取得更高质量的全局解。

### 3.5 本章小结

本章研究了一种具有合作竞争机制的人类学习优化算法,通过引入合作竞争机制,提升了算法的探索与开发能力。具体而言,当前个体通过计算与其他个体之间的欧几里得距离,选择适当的合作伙伴,并与其进行适应度竞争。竞争胜出的个体知识数据库作 IKD 作为当前个体的学习对象,促进了合作与竞争的有效结合,不仅增强了算法的多样性,防止因 IKD 停更而导致的局部最优困境,还通过重新学习机制确保 IKD 的持续变化,提升了算法的空间探索能力。此外,进一步探讨了合作竞争学习算子与社会学习算子在种群信息利用效率方面的差异,构建了一种增强探索与开采平衡的合作竞争模型,研究了一种增强探索与开采平衡合作竞争人类学习优化算法,根据迭代进程动态调整合作竞争算子的取值,增强了算法在优化过程中探索与开发的有效平衡,并对合作竞争算子和增强探索与开采平衡合作竞争模型的影响进行了深入分析,通过参数学习选择最优的参数组合。最后,实验将 CCHLO 算法和 ECCHLO 算法与多种对比算法应用于 CEC2005 基准函数的测试,结果显示,无论在低维还是高维空间中,所提出算法的整体性能均优于其他对比算法。

## 第4章 具有合作竞争机制的群智能优化算法应用研究

群智能优化算法是一种模仿自然界中群体行为的优化方法,广泛应用于解决复杂的优化问题。例如,在工程设计中<sup>[52]</sup>,通过群体智能算法可以优化设计参数,以达到最佳性能。在交通管理系统中<sup>[53]</sup>,利用这些算法可以优化交通流量,减少拥堵,提高通行效率。在金融领域<sup>[54]</sup>,这些算法被用于资产组合优化和风险管理,帮助投资者做出更明智的决策。这些应用不仅提高了效率,还为决策者提供了科学依据,减少了人为错误的可能性。具有合作竞争机制的群智能优化算法在优化问题的解决中具有重要的应用意义。它不仅提升了算法的性能和适应性,还在多个领域展现了广泛的应用前景,推动了科技进步和社会发展。

### 4.1 连续编码复杂工程优化问题应用研究

在工程优化问题日益复杂的今天,IPSOCC 算法在处理复杂工程优化问题中具有显著意义。该算法通过模拟群体智能,能够高效地找到最优解,显著提高设计效率。与传统优化方法相比,粒子群优化算法通过合作竞争机制的引入,使得不同粒子能在搜索空间中并行探索,加速收敛,缩短设计周期,从而提高工程效率。能有效降低工程成本,在设计过程中不仅优化性能,还考虑材料和生产成本的降低。例如,结构设计中可以优化材料使用,在确保安全性和功能性的基础上降低材料消耗。此外,具有合作竞争机制的粒子群优化算法还能提高设计方案的可靠性,推动可持续发展。随着研究的深入,该算法在更多工程领域的应用前景愈加广泛,为企业提升竞争力提供了有力支持。

#### 4.1.1 拉压弦设计问题

拉压弹簧设计问题在结构力学和材料科学中具有重要的意义。这类问题涉及到杆件、梁、桁架等结构在受力作用下的应力、应变、位移等响应,是实际工程中常见的问题。通过研究拉压弹簧设计问题,可以为实际工程提供合理的结构方案,保证结构的安全性、经济性和可靠性。拉压弹簧设计问题有助于提高结构的安全性。在实际工程中,结构的安全问题至关重要。通过对拉弹簧设计问题的研究和分析,可以预测结构在不同工况下的性能表现,从而为结构设计提供依据,确保结构在各种条件下都能满足安全要求。例如,在桥梁设计中,通过对拉压弹簧的设计问题进行研究,可以确保桥梁在承受车辆荷载和风荷载等外部作用力时,

不会出现裂缝、变形等问题，保证行人和车辆的安全通行。通过对这类问题的研究，可以为实际工程提供合理的结构设计方案，保证结构的安全性、经济性和可靠性。随着科学技术的不断发展，拉压弹簧设计问题的研究将更加深入和完善，为实际工程提供更多的技术支持。弹簧设计问题共涉及三个变量用数学表达式可表示为：

$$\begin{aligned}
 f_{\min}(x) &= (x_3 + 2) \cdot x_2 \cdot x_1^2 \\
 \text{s.t. } G_1(x) &= 1 - \frac{x_2^3 \cdot x_3}{71785 \cdot x_1^4} \leq 0 \\
 G_2(x) &= \frac{(4x_2^2 - x_1 \cdot x_2)}{12566 \cdot (x_2 \cdot x_1^3 - x_1^4)} \leq 0 \\
 G_3(x) &= 1 - \frac{140.45 \cdot x_1}{x_2^3 \cdot x_3} \leq 0 \\
 G_4(x) &= \frac{(x_1 + x_2)}{1.5} - 1 \leq 0 \\
 0.05 &\leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15
 \end{aligned} \tag{4.1}$$

这里  $f_{\min}(x)$  是关于弹簧设计问题的目标函数，通过改变  $x_1$ 、 $x_2$ 、 $x_3$  三个变量（分别对应线圈直径、平均线圈直径、活动线圈数）调整弹簧设计参数，并限制每个变量的取值上下限，通过罚函数处理约束，以满足公式（4.1）的取值约束， $G_1(x)$ 、 $G_2(x)$ 、 $G_3(x)$ 、 $G_4(x)$  为实现设计的约束条件，确保优化结果在物理和工程可行性范围内。由于设定了三个变量，因此求解空间维度设为 3。最后，在该约束条件下，将所提出的 IPSOCC 算法与乌鸦搜索优化算法（CSA）、飞蛾扑火优化算法（MFO）和粒子群优化算法（PSO）进行实验结果对比，各算法的种群数量设为 30，最大迭代次数为 1000，得到总的成本结果如表 4-1。

表 4-1 各算法于拉压弦设计问题的结果比较

| 算法     | 最优决策变量   |          |          | 最优值              |
|--------|----------|----------|----------|------------------|
|        | $X_1$    | $X_2$    | $X_3$    |                  |
| IPSOCC | 0.057145 | 0.509535 | 5.884864 | <b>0.0132257</b> |
| CSA    | 0.066506 | 0.995578 | 9.014876 | 0.0280903        |
| MFO    | 0.060484 | 0.607550 | 4.284075 | 0.0139672        |
| PSO    | 0.053882 | 0.351191 | 13.96804 | 0.0142417        |

表 4-1 实验结果表明，IPSOCC 算法针对优化拉压弦设计问题的最优值最小，为 0.0132257，说明 IPSOCC 算法和其他对比算法相比表现出最佳的模型优化性能。说明 IPSOCC 能够在考虑约束条件的情况下，找到最小的目标函数值，表明其在解决此类问题时具有较强的全局搜索能力和收敛性能。

#### 4.1.2 三杆桁架设计问题

三杆桁架设计问题是为了使桁架在满足一定约束条件下，具有最佳的性能和经济性。通过合理选择优化方法和技术，并结合数值模型和仿真分析，可以实现三杆桁架设计的高效优化，提高结构的性能和经济效益。其主要特点是三个杆件通过铰链连接在一起，形成一个稳定的空间结构。在设计三杆桁架时，需要考虑以下几个方面的问题：

(1) 杆件的截面尺寸：三杆桁架的杆件截面尺寸直接影响到整个结构的稳定性和承载能力。因此，在设计时需要根据实际需求选择合适的截面尺寸。一般来说，杆件截面尺寸越大，结构的稳定性越好，但成本也会相应增加。

(2) 杆件的连接方式：三杆桁架的杆件连接方式有多种，如焊接、螺栓连接等。不同的连接方式会影响到结构的可靠性和施工难度。在实际设计中，需要根据工程条件和施工要求选择合适的连接方式。

(3) 节点设计：三杆桁架的节点是连接三个杆件的重要部分，其设计直接影响到结构的稳定性和承载能力。在设计时，需要充分考虑节点的刚度、强度和稳定性等因素，以保证整个结构的安全性能。

三杆桁架设计问题共涉及两个变量，其数学表达式如公式（4.2）：

$$\begin{aligned}
 f_{\min}(x) &= 100 \cdot (2\sqrt{2} \cdot x_1 + x_2) \\
 \text{s.t. } G_1(x) &= 2 \cdot \frac{\sqrt{2} \cdot x_1 + x_2}{2 \cdot x_1 \cdot x_2 + \sqrt{2} \cdot x_1^2} \leq 3 \\
 G_2(x) &= 2 \cdot \frac{x_2}{2 \cdot x_1 \cdot x_2 + \sqrt{2} \cdot x_1^2} \leq 3 \\
 G_3(x) &= 2 \cdot \frac{1}{x_1 + \sqrt{2} \cdot x_2} \leq 3 \\
 0.01 &\leq x_i \leq 1, i=1, 2
 \end{aligned} \tag{4.2}$$

这里  $f_{\min}(x)$  是关于三杆桁架设计问题的目标函数，设定好变量的上下限如公式（4.2），在设定上下限的范围内调整  $x_1$ 、 $x_2$  两个变量，使得三杆桁架设计参数满足公式（4.2）的取值约束， $G_1(x)$ 、 $G_2(x)$ 、 $G_3(x)$  为实现设计的约束条件。该问题主要通过控制两个参数变量使得三杆桁架结构的总质量最小，由于设定了两个变量，因此求解空间维度设为 2，在该约束条件下 IPSOCC 算法及对比算法的结果如表 4-2。

表 4-2 各算法于三杆桁架设计问题的结果比较

| 算法     | 最优决策变量  |         | 最优值                 |
|--------|---------|---------|---------------------|
|        | $X_1$   | $X_2$   |                     |
| IPSOCC | 0.67301 | 0.01000 | <b>1.913565e+02</b> |
| CSA    | 0.71791 | 0.04984 | 2.080387e+02        |
| MFO    | 0.67301 | 0.01000 | 1.913565e+02        |
| PSO    | 0.67747 | 0.01691 | 1.933090e+02        |

表 4-2 的数据表明, IPSOCC 在三杆桁架设计问题中能得到较小的最优值, 为 1.913565e+02, 说明 IPSOCC 算法能够在约束条件下高效地找到最小的设计成本或最优解, 展示了其在全局优化和局部搜索能力方面的优势。与其他算法相比, IPSOCC 能够更好地平衡探索与开发, 避免陷入局部最优解, 从而为复杂的工程设计问题提供了更为精确和可靠的解决方案。

#### 4.1.3 悬梁架设计问题

悬梁架设计问题的意义在于保证结构的安全性能、稳定性和经济性。悬梁架是一种常见的建筑结构形式, 主要用于承受水平荷载和支撑建筑物的重量。在设计悬梁架时, 需要考虑多个方面的问题, 如材料选择、截面尺寸、连接方式、结构布局和抗震性能等。通过对这些问题的合理解决, 使悬梁架成为一种广泛应用的建筑结构形式。悬梁架设计问题的意义在于保证结构的安全性能、稳定性和经济性。悬梁架是一种常见的建筑结构形式, 主要用于承受水平荷载和支撑建筑物的重量。在设计悬梁架时, 需要考虑多个方面的问题, 如材料选择、截面尺寸、连接方式、结构布局和抗震性能等。通过对这些问题的合理解决, 可以使悬梁架成为一种广泛应用的建筑结构形式。悬梁架的设计问题具有 5 个变量, 其数学表达式如公式 (4.3):

$$\begin{aligned}
 f_{\min}(x) &= 0.0624 \cdot (x_1 + x_2 + x_3 + x_4 + x_5) \\
 s.t. \quad G_1(x) &= \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \leq 1 \\
 0.01 &\leq x_i \leq 100, i=1,2,3,4,5
 \end{aligned} \tag{4.3}$$

其中,  $f_{\min}(x)$  是关于悬梁架设计问题的目标函数, 通过  $x_1$ 、 $x_2$ 、 $x_3$ 、 $x_4$ 、 $x_5$  五个变量调整悬梁架设计参数以满足公式 (4.3) 的取值约束,  $G_1(x)$  为实现设计的约束条件。由于, 在悬梁架设计问题中设定了五个变量, 因此, 求解空间维度设为 5。在该约束条件下 IPSOCC 算法与对比算法 CSA、MFO 和 PSO 的实验结果如表 4-3。

表 4-3 各算法于悬梁架设计问题的结果比较

| 算法     | 最优决策变量    |           |           |           |           | 最优值              |
|--------|-----------|-----------|-----------|-----------|-----------|------------------|
|        | $X_1$     | $X_2$     | $X_3$     | $X_4$     | $X_5$     |                  |
| IPSOCC | 6.0025543 | 5.3332652 | 4.5228065 | 3.4719946 | 2.1442235 | <b>1.3400332</b> |
| CSA    | 6.0253321 | 5.1419699 | 4.2653314 | 3.5196020 | 2.8437245 | 1.3600679        |
| MFO    | 6.1246076 | 5.2947110 | 4.4613957 | 3.4827724 | 2.1160310 | 1.3403219        |
| PSO    | 6.3720417 | 5.1752739 | 4.5393386 | 3.2367155 | 2.2505424 | 1.3462121        |

表 4-3 的实验结果表明，IPSOCC 与 CSA、MFO 和 PSO 算法相比寻优精度最高，最优值达到 1.3400332，说明 IPSOCC 算法能够在优化悬梁架设计问题的过程中提供最精确的解决方案。验证了 IPSOCC 通过交互式合作竞争机制的引入，在全局搜索和局部搜索之间实现了良好的平衡，使得算法能够高效地探索解空间，并找到最优或接近最优的设计参数。

## 4.2 二进制编码复杂工程优化问题应用研究

在提高求解效率、优化资源配置、增强解的精确性、推动技术创新、提高设计方案准确性等问题中，ECCHLO 算法的应用具有重要意义。0-1 背包问题是一个经典的组合优化问题，广泛应用于资源分配、库存管理等领域。ECCHLO 算法通过灵活调整个体间的合作竞争机制，能够有效地探索解空间，找到最优解。在 ECCHLO 算法中，每个个体代表一种可能的解，通过学习周围其他个体的成功经验，迅速调整个体信息的策略，从而达到提升算法性能的目的。合作竞争机制使得算法能够在复杂的解空间中进行全局搜索，避免陷入局部最优，显著提高了求解效率。对于实际应用而言，ECCHLO 算法的引入意味着可以在更短的时间内得到更优的资源配置方案，从而达到提升生产效率、优化资源配置、增强解的精确性的目的。在复杂工程优化问题中，资源的有效配置和优化设计的合理性至关重要。

### 4.2.1 0-1 背包问题

0-1 背包问题是一个经典的组合优化问题，也是动态规划中常见的问题之一。在 0-1 背包问题中，给定一个背包容量（通常用一个正整数表示）和一组物品，每个物品有自己的重量和价值，在不超过背包容量的情况下，选择一些物品放入背包，使得背包中物品的总价值最大。实际上，装载问题、发电机组开关问题以及选择问题都能够归纳为 0-1 背包问题。因此用 0-1 背包问题可以作为一个标准

化的测试问题，用于评估不同算法的性能和效果。在后面的工作中，将 ECCHLO 与其他元启发式算法用来解决 0-1 背包问题（0-1KP）。

$$\begin{aligned} \text{Max } f(x) &= \sum_{j=1}^N p_j x_j \\ \text{s.t. } &\begin{cases} \sum_{j=1}^N w_j x_j \leq C \\ x_j = \{0, 1\}, j = 1, 2, \dots, N \end{cases} \end{aligned} \quad (4.4)$$

其中， $x_j$  为 0 或 1， $x_j$  为 0 表示该物品不放入背包， $x_j$  为 1 则表示放入背包。由于，在物品选择时可能会产生物品的总重量大于背包的承重限制。因此，需要引入约束条件，使得选择的物品总重量在规定的数值之内。这里引入了罚函数的方式用来处理不满足规定条件的选项，罚函数如公式（4.5）：

$$\begin{aligned} \text{Max } F(x) &= f(x) - \lambda \times \max(0, c) \\ c &= \sum_{j=1}^N w_j x_j - C \end{aligned} \quad (4.5)$$

其中， $\lambda$  为惩罚系数，为了满足要求，将不满足条件解的适应度劣于可行解的适应度，通过罚函数控制被选择的物品总重量在满足背包承重限制的前提下，总价值最大。

这里根据<sup>[55],[56]</sup>对 0-1KP 的背包承重能力以及物品数量进行设定，将背包的承重能力依次设为 50、100、250、500、800、1000、1200、1500、2000 和 2500 对应的物品数量设置为 600、1200、3000、6000、10000、12000、15000、18000、25000 和 30000。并且将物品的重量  $w_j$  设为 5 到 20 的随机数，将物品价值  $p_j$  设为 50 到 100 的随机数。将研究的 ECCHLO 算法与人类学习优化算法（HLO），二进制粒子群优化算法（Binary Particle Swarm Optimization Algorithm, BPSO），二进制差分进化算法（Binary Differential Evolution Algorithm, BDE），二进制蝙蝠算法（Binary Bat Algorithm, BBA）进行对比。各个算法种群数量设定为 50，最大迭代次数设定为 1000。为公平起见，将每个设定的值均做三次实验，每次实验的算法通过独立运行 50 次以求得平均值、标准差、中间值和最优值并记录，结果如表 4-6。

表 4-4，表 4-5 结果表明，ECCHLO 算法在类型 50.1、50.2、50.3、100.1、100.2、250.1、800.1、800.2、800.3、1000.1、1000.2、1200.1、1200.2、1200.3、1500.1、1500.2、1500.3、2000.1、2000.2、2000.3、2500.1、2500.2、2500.3 情况

下的实验结果均比 BPSO、BDE、BBA 和 HLO 算法效果要好，并且，根据表 4-6 的 Rank 排名显示 ECCHLO 算法的综合排名为第一。因此，ECCHLO 算法相较于其他对比算法性能要优越，体现出比对比算法更加优越的整体优化性能。

表 4-4 各算法于 0-1 背包问题的低维度结果比较

| 类型    | 算法     | Mean           | STD            | Median         | Best         | Rank |
|-------|--------|----------------|----------------|----------------|--------------|------|
| 50.1  | BPSO   | <b>3841</b>    | <b>0</b>       | <b>3841</b>    | <b>3841</b>  | 1    |
|       | BDE    | 3834.7         | 9.1779         | 3839           | 3841         | 4    |
|       | BBA    | 3629.6         | 62.8423        | 3620.5         | 3756         | 5    |
|       | HLO    | <b>3841</b>    | <b>0</b>       | <b>3841</b>    | <b>3841</b>  | 1    |
|       | ECCHLO | <b>3841</b>    | <b>0</b>       | <b>3841</b>    | <b>3841</b>  | 1    |
| 50.2  | BPSO   | 3729           | 2.8284         | 3730           | 3731         | 3    |
|       | BDE    | 3717.7         | 5.9638         | 3718           | 3724         | 4    |
|       | BBA    | 3566.1         | 58.0449        | 3571           | 3628         | 5    |
|       | HLO    | <b>3731</b>    | <b>0</b>       | <b>3731</b>    | <b>3731</b>  | 1    |
|       | ECCHLO | <b>3731</b>    | <b>0</b>       | <b>3731</b>    | <b>3731</b>  | 1    |
| 50.3  | BPSO   | <b>3799</b>    | <b>0</b>       | <b>3799</b>    | <b>3799</b>  | 1    |
|       | BDE    | 3794.7         | 4.1110         | 3794           | 3799         | 4    |
|       | BBA    | 3513           | 71.2398        | 3501.5         | 3650         | 5    |
|       | HLO    | <b>3799</b>    | <b>0</b>       | <b>3799</b>    | <b>3799</b>  | 1    |
|       | ECCHLO | <b>3799</b>    | <b>0</b>       | <b>3799</b>    | <b>3799</b>  | 1    |
| 100.1 | BPSO   | <b>7536</b>    | <b>0</b>       | <b>7536</b>    | <b>7536</b>  | 1    |
|       | BDE    | <b>7536</b>    | <b>0</b>       | <b>7536</b>    | <b>7536</b>  | 1    |
|       | BBA    | 6101.7         | 142.2471       | 6098.5         | 6265         | 5    |
|       | HLO    | <b>7536</b>    | <b>0</b>       | <b>7536</b>    | <b>7536</b>  | 1    |
|       | ECCHLO | <b>7536</b>    | <b>0</b>       | <b>7536</b>    | <b>7536</b>  | 1    |
| 100.2 | BPSO   | <b>7637</b>    | <b>0</b>       | <b>7637</b>    | <b>7637</b>  | 1    |
|       | BDE    | 7629.6         | 5.5418         | 7628.5         | 7637         | 4    |
|       | BBA    | 6299.4         | 143.3467       | 6285           | 6635         | 5    |
|       | HLO    | <b>7637</b>    | <b>0</b>       | <b>7637</b>    | <b>7637</b>  | 1    |
|       | ECCHLO | <b>7637</b>    | <b>0</b>       | <b>7637</b>    | <b>7637</b>  | 1    |
| 100.3 | BPSO   | 7225.9         | 5.8585         | 7225.5         | 7231         | 3    |
|       | BDE    | 7204           | 15.0555        | 7206           | 7225         | 4    |
|       | BBA    | 6034.8         | 137.4949       | 6007           | 6342         | 5    |
|       | HLO    | <b>7231</b>    | <b>0</b>       | <b>7231</b>    | <b>7231</b>  | 1    |
|       | ECCHLO | 7228.5         | 3.2745         | 7231           | 7231         | 2    |
| 250.1 | BPSO   | 18372.8        | 16.9430        | 18370.5        | 18418        | 3    |
|       | BDE    | 18331          | 36.8148        | 18338          | 18377        | 4    |
|       | BBA    | 13202.2        | 244.2612       | 13176          | 13650        | 5    |
|       | HLO    | 18402.6        | 22.5891        | 18415.5        | <b>18420</b> | 2    |
|       | ECCHLO | <b>18409</b>   | <b>8.6410</b>  | <b>18413</b>   | 18416        | 1    |
| 250.2 | BPSO   | 18423          | 2.0548         | 18423.5        | 18425        | 2    |
|       | BDE    | 18407.9        | 13.0081        | 18412.5        | 18418        | 4    |
|       | BBA    | 12976.4        | 246.6388       | 13050          | 13206        | 5    |
|       | HLO    | <b>18424.3</b> | <b>0.9487</b>  | <b>18424.5</b> | <b>18425</b> | 1    |
|       | ECCHLO | 18422.5        | 2.0138         | 18423          | 18424        | 3    |
| 250.3 | BPSO   | 17748.9        | 19.9246        | 17742.5        | 17782        | 3    |
|       | BDE    | 17608.9        | 69.3645        | 17623          | 17685        | 4    |
|       | BBA    | 13130.6        | 324.2784       | 13081.5        | 13718        | 5    |
|       | HLO    | <b>17773</b>   | <b>8.2999</b>  | <b>17771</b>   | <b>17788</b> | 1    |
|       | ECCHLO | 17751.3        | 13.6549        | 17749          | 17780        | 2    |
| 500.1 | BPSO   | 36544.4        | 34.2741        | 36547          | 36586        | 3    |
|       | BDE    | 36386.1        | 89.4010        | 36369          | 36514        | 4    |
|       | BBA    | 24186.3        | 425.8811       | 24011.5        | 24968        | 5    |
|       | HLO    | <b>36557.1</b> | <b>18.5679</b> | 36554          | <b>36601</b> | 1    |
|       | ECCHLO | 36552.9        | 28.6257        | <b>36556</b>   | 36593        | 2    |
| 500.2 | BPSO   | 36968.6        | 31.0240        | 36975          | 37001        | 3    |
|       | BDE    | 36906.7        | 29.2577        | 36904          | 36947        | 4    |
|       | BBA    | 24170.7        | 383.8573       | 24260          | 24593        | 5    |
|       | HLO    | <b>36998.5</b> | <b>4.4033</b>  | <b>36998</b>   | <b>37006</b> | 1    |
|       | ECCHLO | 36978.3        | 17.4423        | 36980.5        | 36996        | 2    |
| 500.3 | BPSO   | 36827.6        | 28.7642        | 36827.5        | 36889        | 3    |
|       | BDE    | 36664.7        | 65.4575        | 36683          | 36734        | 4    |
|       | BBA    | 24274.5        | 238.3248       | 24222.5        | 24718        | 5    |

续表：4-4

|       |        |                |                |                |              |   |
|-------|--------|----------------|----------------|----------------|--------------|---|
|       | HLO    | <b>36870.3</b> | 19.9669        | <b>36864.5</b> | <b>36904</b> | 1 |
|       | ECCHLO | 36846.4        | <b>16.3653</b> | 36840.5        | 36867        | 2 |
| 800.1 | BPSO   | <b>59992</b>   | <b>0</b>       | <b>59992</b>   | <b>59992</b> | 1 |
|       | BDE    | <b>59992</b>   | <b>0</b>       | <b>59992</b>   | <b>59992</b> | 1 |
|       | BBA    | 37057.4        | 350.2460       | 37061.5        | 37611        | 5 |
|       | HLO    | <b>59992</b>   | <b>0</b>       | <b>59992</b>   | <b>59992</b> | 1 |
|       | ECCHLO | <b>59992</b>   | <b>0</b>       | <b>59992</b>   | <b>59992</b> | 1 |
| 800.2 | BPSO   | <b>59938</b>   | <b>0</b>       | <b>59938</b>   | <b>59938</b> | 1 |
|       | BDE    | <b>59938</b>   | <b>0</b>       | <b>59938</b>   | <b>59938</b> | 1 |
|       | BBA    | 36933.9        | 375.8337       | 36940.5        | 37403        | 5 |
|       | HLO    | <b>59938</b>   | <b>0</b>       | <b>59938</b>   | <b>59938</b> | 1 |
|       | ECCHLO | <b>59938</b>   | <b>0</b>       | <b>59938</b>   | <b>59938</b> | 1 |
| 800.3 | BPSO   | <b>60081</b>   | <b>0</b>       | <b>60081</b>   | <b>60081</b> | 1 |
|       | BDE    | <b>60081</b>   | <b>0</b>       | <b>60081</b>   | <b>60081</b> | 1 |
|       | BBA    | 36998.9        | 533.7121       | 37037          | 37789        | 5 |
|       | HLO    | <b>60081</b>   | <b>0</b>       | <b>60081</b>   | <b>60081</b> | 1 |
|       | ECCHLO | <b>60081</b>   | <b>0</b>       | <b>60081</b>   | <b>60081</b> | 1 |

表 4-5 各算法于 0-1 背包问题的高维度结果比较

| 类型     | 算法     | Mean            | STD            | Median         | Best          | Rank |
|--------|--------|-----------------|----------------|----------------|---------------|------|
| 1000.1 | BPSO   | 73616.8         | 60.0607        | 73630          | 73695         | 2    |
|        | BDE    | 73372.1         | 76.5426        | 73373.5        | 73530         | 4    |
|        | BBA    | 45644.6         | 520.6812       | 45632          | 46807         | 5    |
|        | HLO    | 73596.8         | 48.0828        | 73595.5        | 73665         | 3    |
|        | ECCHLO | <b>73642</b>    | <b>27.1047</b> | <b>73645.5</b> | <b>73679</b>  | 1    |
| 1000.2 | BPSO   | 72758.8         | 49.1930        | 72770          | 72828         | 3    |
|        | BDE    | 72554           | 100.3848       | 72503.5        | 72742         | 4    |
|        | BBA    | 45084.5         | 396.3187       | 45156          | 45617         | 5    |
|        | HLO    | 72761.2         | 41.6088        | 72771.5        | <b>72815</b>  | 2    |
|        | ECCHLO | <b>72773.6</b>  | <b>31.7427</b> | <b>72781</b>   | 72813         | 1    |
| 1000.3 | BPSO   | 73107.3         | 53.5227        | 73099          | 73201         | 3    |
|        | BDE    | 72933.1         | 63.1813        | 72927          | 73051         | 4    |
|        | BBA    | 44626.1         | 524.2443       | 44697.5        | 45651         | 5    |
|        | HLO    | <b>73152.9</b>  | <b>31.9529</b> | <b>73147</b>   | <b>73223</b>  | 1    |
|        | ECCHLO | 73117.9         | 36.1492        | 73121.5        | 73164         | 2    |
| 1200.1 | BPSO   | <b>90039</b>    | <b>0</b>       | <b>90039</b>   | <b>90039</b>  | 1    |
|        | BDE    | <b>90039</b>    | <b>0</b>       | <b>90039</b>   | <b>90039</b>  | 1    |
|        | BBA    | 53458.4         | 510.9230       | 53498.5        | 54201         | 5    |
|        | HLO    | <b>90039</b>    | <b>0</b>       | <b>90039</b>   | <b>90039</b>  | 1    |
|        | ECCHLO | <b>90039</b>    | <b>0</b>       | <b>90039</b>   | <b>90039</b>  | 1    |
| 1200.2 | BPSO   | <b>90065</b>    | <b>0</b>       | <b>90065</b>   | <b>90065</b>  | 1    |
|        | BDE    | <b>90065</b>    | <b>0</b>       | <b>90065</b>   | <b>90065</b>  | 1    |
|        | BBA    | 53871.6         | 777.2037       | 53460          | 55172         | 5    |
|        | HLO    | <b>90065</b>    | <b>0</b>       | <b>90065</b>   | <b>90065</b>  | 1    |
|        | ECCHLO | <b>90065</b>    | <b>0</b>       | <b>90065</b>   | <b>90065</b>  | 1    |
| 1200.3 | BPSO   | 88447.4         | 35.6346        | 88449          | 88492         | 2    |
|        | BDE    | 88351.3         | 70.1887        | 88337.5        | 88467         | 4    |
|        | BBA    | 53135.8         | 694.8784       | 52945.5        | 54293         | 5    |
|        | HLO    | 88421.1         | 46.3188        | 88409.5        | 88508         | 3    |
|        | ECCHLO | <b>88452</b>    | <b>34.8450</b> | <b>88460</b>   | <b>88521</b>  | 1    |
| 1500.1 | BPSO   | 109699.5        | 95.5885        | 109691.5       | 109843        | 3    |
|        | BDE    | 109500.4        | 139.4005       | 109448         | 109730        | 4    |
|        | BBA    | 66207.4         | 886.0163       | 66502.5        | 67448         | 5    |
|        | HLO    | 109850.9        | 79.2667        | 109859         | 109977        | 2    |
|        | ECCHLO | <b>109870.8</b> | <b>73.5856</b> | <b>109860</b>  | 109968        | 1    |
| 1500.2 | BPSO   | 110662.9        | 87.5220        | 110671.5       | 110824        | 3    |
|        | BDE    | 110516.7        | 84.8856        | 110528         | 110660        | 4    |
|        | BBA    | 65935.1         | 493.9776       | 65877.5        | 67037         | 5    |
|        | HLO    | 110707.3        | 87.7434        | 110670.5       | 110840        | 2    |
|        | ECCHLO | <b>110732.5</b> | <b>82.0220</b> | <b>110750</b>  | <b>110865</b> | 1    |
| 1500.3 | BPSO   | 109756.5        | 117.2445       | 109741.5       | 109910        | 3    |
|        | BDE    | 109385.9        | 109.7395       | 109384.5       | 109564        | 4    |
|        | BBA    | 66314.1         | 736.7483       | 66155.5        | 67674         | 5    |
|        | HLO    | 109842.8        | 161.3897       | 109844         | <b>110083</b> | 2    |
|        | ECCHLO | <b>109931.5</b> | <b>94.5436</b> | <b>109940</b>  | 110060        | 1    |
| 2000.1 | BPSO   | 147445.2        | 378.2606       | 147429         | 148160        | 4    |

续表：4-5

|        |        |                 |                 |                 |               |   |
|--------|--------|-----------------|-----------------|-----------------|---------------|---|
|        | BDE    | 147565.5        | 398.5434        | 147534          | 148326        | 3 |
|        | BBA    | 85645.4         | 661.9314        | 85767.5         | 86739         | 5 |
|        | HLO    | 149215.8        | <b>145.0339</b> | 149222          | 149441        | 2 |
|        | ECCHLO | <b>149345.8</b> | 179.7528        | <b>149323</b>   | <b>149611</b> | 1 |
| 2000.2 | BPSO   | 147728          | 352.2988        | 147719          | 148305        | 4 |
|        | BDE    | 147932          | 302.2463        | 147931          | 148363        | 3 |
|        | BBA    | 85797.2         | 614.0005        | 85751.5         | 86802         | 5 |
|        | HLO    | 149733.1        | <b>148.9216</b> | 149743.5        | 149961        | 2 |
|        | ECCHLO | <b>149779.6</b> | 156.6108        | <b>149757.5</b> | <b>150024</b> | 1 |
| 2000.3 | BPSO   | 147840.9        | 467.8610        | 147879          | 148601        | 4 |
|        | BDE    | 148159.7        | 60.4822         | 148163          | 148247        | 3 |
|        | BBA    | 85821.3         | 812.0063        | 85671           | 87606         | 5 |
|        | HLO    | 149033.2        | 66.5128         | 149046          | <b>149172</b> | 2 |
|        | ECCHLO | <b>149080.3</b> | <b>58.2524</b>  | <b>149080.5</b> | 149153        | 1 |
| 2500.1 | BPSO   | 180562.9        | 589.5112        | 180749          | 181217        | 4 |
|        | BDE    | 181820.6        | 250.7966        | 181857.5        | 182104        | 3 |
|        | BBA    | 105627.8        | 780.8225        | 105647          | 106805        | 5 |
|        | HLO    | 183359.4        | 202.1964        | 183377          | 183610        | 2 |
|        | ECCHLO | <b>183450.9</b> | <b>100.8458</b> | <b>183453</b>   | <b>183648</b> | 1 |
| 2500.2 | BPSO   | 178808.5        | 694.6117        | 178866          | 179692        | 4 |
|        | BDE    | 179544.4        | 299.6309        | 179630          | 181819        | 3 |
|        | BBA    | 105499.7        | 790.2117        | 105407          | 106806        | 5 |
|        | HLO    | 181315.8        | <b>168.7831</b> | 181360.5        | 181512        | 2 |
|        | ECCHLO | <b>181431.5</b> | 284.5789        | <b>181415</b>   | <b>182039</b> | 1 |
| 2500.3 | BPSO   | 179891.1        | 671.3858        | 180036          | 180995        | 4 |
|        | BDE    | 179914.3        | 654.9661        | 179956.5        | 179983        | 3 |
|        | BBA    | 105736.2        | 774.2928        | 105728          | 107217        | 5 |
|        | HLO    | 181724.6        | <b>167.8182</b> | 181748.5        | 181973        | 2 |
|        | ECCHLO | <b>181812.2</b> | 212.2262        | <b>181796</b>   | <b>182115</b> | 1 |

表 4-6 各算法于 0-1 背包问题排名结果

| 算法             | BPSO | BDE | BBA | HLO | ECCHLO |
|----------------|------|-----|-----|-----|--------|
| <b>AvgRank</b> | 2.5  | 3.2 | 5   | 1.5 | 1.27   |
| <b>Rank</b>    | 3    | 4   | 5   | 2   | 1      |

#### 4.2.2 步进圆锥滑轮问题

本小结将 ECCHLO 算法应用于步进圆锥滑轮问题<sup>[57]</sup>。步进圆锥滑轮问题是一个工程力学问题，描述了绳索或带传动系统中使用的滑轮。这个问题涉及到滑轮尺寸设计和分析。解决步进圆锥滑轮问题需要考虑力学原理、材料力学、摩擦力学等知识，以便设计出合适的滑轮结构，满足传动系统的要求。在工程实践中，步进圆锥滑轮问题常常涉及到工程师需要保证滑轮的尺寸、皮带张力、皮带传输功率的相同，以确保传动系统的稳定性和效率。此外，合理的滑轮尺寸设计还需考虑力矩传递、运动精度等因素，避免因设计不当而引起过多的能量损失或不稳定的传动行为。因此，在设计过程中必须平衡各个参数，以优化滑轮的工作性能，保证传动系统的顺畅运行。

此问题来源于，在锥形皮带轮与皮带宽度相同的前提下，通过 5 个变量设计重量最小的 4 步锥滑轮。其中 4 个变量为 4 个滑轮的直径  $d_i$  ( $i=1,2,3,4$ )，另外一个

变量为滑轮的宽度。问题中共包含 8 个非线性约束条件以及 3 个线性约束条件。圆锥滑轮的设计用于传输不小于 0.75hp 的功率。当输入转轴的转速为 350rpm 时，对应的输出转轴的速度应该规定为 750、450、250 和 150rpm。皮带拉紧时的张力应该为松弛时张力的两倍以上，皮带与滑轮之间的摩擦系数为  $\mu$ ，皮带厚度为  $t$ 。皮带拉紧侧的张力在皮带中的应力用  $s$  表示。步进圆锥滑轮问题的公式如公式（4.6）：

$$\begin{aligned}
 f_{\min}(x) &= \rho\omega \left[ d_1^2 \left\{ 1 + \left( \frac{N_1}{N} \right)^2 \right\} + d_2^2 \left\{ 1 + \left( \frac{N_2}{N} \right)^2 \right\} \right. \\
 &\quad \left. + d_3^2 \left\{ 1 + \left( \frac{N_3}{N} \right)^2 \right\} + d_4^2 \left\{ 1 + \left( \frac{N_4}{N} \right)^2 \right\} \right] \\
 \text{s.t. } h_1(x) &= C_1 - C_2 = 0, \quad h_2(x) = C_1 - C_3 = 0, \\
 h_3(x) &= C_1 - C_4 = 0, \\
 g_{1,2,3,4}(x) &= R_i \geq 2, \quad g_{5,6,7,8}(x) = P_i \geq (0.75 * 745.6998)
 \end{aligned} \tag{4.6}$$

其中， $C_i$  为输出速度  $N_i$  所需的皮带长度， $C_i$  表达式为公式（4.7）：

$$C_i = \frac{\pi d_i}{2} \left( 1 + \frac{N_i}{N} \right) + \frac{\left( \frac{N_i}{N} - 1 \right)^2}{4a} + 2a \quad i = (1, 2, 3, 4). \tag{4.7}$$

其中， $N$  为输入轴的速度， $a$  为轴中心间的中心距离。

$$\frac{T_1^i}{T_2^i} = e^{\mu\theta_i} \tag{4.8}$$

公式（4.8）中  $T_1^i$  与  $T_2^i$  为第  $i$  级滑轮皮带拉紧侧的张力和松弛侧的张力， $\theta_i$  为皮带在第  $i$  级滑轮上的搭接角， $\theta_i$  公式如公式（4.9）：

$$\theta_i = \pi - 2 \sin^{-1} \left[ \frac{\left( \frac{N_i}{N} - 1 \right) d_i}{2a} \right] \tag{4.9}$$

因此，皮带张力比  $R_i$  的约束条件变成公式（4.10）：

$$R_i = \exp \left\{ \mu \left[ \pi - 2 \sin^{-1} \left\{ \left( \frac{N_i}{N} - 1 \right) \frac{d_i}{2a} \right\} \right] \right\} \geq 2, \quad i = 1, 2, 3, 4 \tag{4.10}$$

皮带最大张力的限制条件如公式（4.11）：

$$T_1^i = st\omega, \quad i = 1, 2, 3, 4 \tag{4.11}$$

其中， $s$  为皮带能够允许的最大应力。对滑轮传输功率  $P_i$  的约束条件公式可以表示为公式（4.12）：

$$P_i = \frac{(T_1^i - T_2^i) \pi d_i (350)}{33,000} \geq 0.75 \tag{4.12}$$

因此,  $P_i$  的约束条件公式可改为公式 (4.13) :

$$P_i = st\omega \left[ 1 - \exp \left[ -\mu \left\{ \pi - 2 \sin^{-1} \left\{ \left( \frac{N_i}{N} - 1 \right) \frac{d_i}{2a} \right\} \right\} \right] \right] \frac{\pi d_i N_i}{60} \geq 0.75 \quad (4.13)$$

$i=(1,2,3,4)$

$$\rho = 7200 \text{ kg} / \text{m}^3, a = 3 \text{ m}, \mu = 0.35, s = 1.75 \text{ MPa}, t = 8 \text{ mm}. \quad (4.14)$$

将 ECCHLO 与人类学习算法 (HLO)、二进制粒子群优化算法 (BPSO), 二进制差分进化算法 (BDE), 二进制蝙蝠算法 (BBA) 进行对比。对比结果如表 4-7。

表 4-7 各算法于步进圆锥滑轮问题的结果比较

| 算法     | 最优决策变量  |         |         |         |         | 最优值            |
|--------|---------|---------|---------|---------|---------|----------------|
|        | $X_1$   | $X_2$   | $X_3$   | $X_4$   | $X_5$   |                |
| ECCHLO | 38.5547 | 52.9687 | 81.9141 | 84.7774 | 89.7363 | <b>16.1511</b> |
| HLO    | 39.5576 | 55.3134 | 72.3779 | 42.0172 | 76.1353 | 16.5410        |
| BPSO   | 40.6958 | 56.0968 | 75.0745 | 89.8527 | 90.0000 | 18.3513        |
| BDE    | 41.1221 | 57.7600 | 75.3612 | 87.0431 | 90.0000 | 25.8978        |
| BBA    | 39.9290 | 54.8894 | 73.3998 | 87.1819 | 87.2451 | 17.0250        |

表 4-7 展示了 ECCHLO 与 HLO、BPSO、BDE 和 BBA 算法对于步进圆锥滑轮问题的解决方案, 将 ECCHLO 算法与其他算法的实验结果进行比较。结果表明 ECCHLO 能够找到最优值达到 16.1511, 均优于对比算法在步进圆锥滑轮问题下找到的最优值, 使得滑轮在满足约束条件的前提下得到的质量最小。该结果进一步验证了 ECCHLO 算法在复杂工程优化问题中的优越性和适用性, 表明其在结构优化、机械设计等领域具有广泛的应用潜力。

#### 4.2.3 行星轮系设计优化问题

行星轮系设计优化<sup>[58]</sup>是机械工程领域中的一个重要课题, 行星轮系, 也称为行星齿轮机构, 由一个中心的太阳齿轮、多个围绕太阳齿轮旋转的行星齿轮和一个外环(环齿轮)组成。这种配置使得行星轮系在传递扭矩和速度方面非常有效, 同时也保证了紧凑的设计和高承载能力。

行星轮系设计的优化关键在于如何选择齿轮的尺寸、齿数比和材料等, 以达到预定的性能指标, 如高效率、低噪音和低误差。此次问题主要目标是将行星轮传动比的最大误差最小化。问题共包括了 9 个变量, 其中 6 个整数变量为  $N_i (i=1,2,3,4,5,6)$  为齿轮的齿数, 行星齿轮的数量  $P$ , 以及齿轮的模数  $m_1, m_2$ 。下面

将详细描述行星轮系设计的具体公式包括：目标函数、约束条件以及求解空间的限制。如公式（4.15）：

$$\begin{aligned}
x &= (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) = (N_1, N_2, N_3, N_4, N_5, N_6, m_1, m_2, p). \\
f_{\min}(x) &= \max |i_k - i_{ok}|, k = \{1, 2, \dots, R\}. \\
s.t. \quad g_1(x) &= m_2(N_6 + 2.5) - D_{\max} \leq 0, \\
g_2(x) &= m_1(N_1 + N_2) + m_1(N_2 + 2) - D_{\max} \leq 0, \\
g_3(x) &= m_2(N_4 + N_5) + m_2(N_5 + 2) - D_{\max} \leq 0, \\
g_4(x) &= |m_1(N_1 + N_2) - m_2(N_6 + N_3)| - m_1 - m_2 \leq 0, \\
g_5(x) &= -(N_1 + N_2)\sin(\pi/p) + N_2 + 2 + \delta_{22} \leq 0, \\
g_6(x) &= -(N_6 + N_3)\sin(\pi/p) + N_3 + 2 + \delta_{33} \leq 0, \\
g_7(x) &= -(N_4 + N_5)\sin(\pi/p) + N_5 + 2 + \delta_{55} \leq 0, \\
g_8(x) &= (N_3 + N_5 + 2 + \delta_{35})^2 - (N_6 - N_3)^2 \\
&\quad - (N_4 + N_5)^2 + 2(N_6 - N_3)(N_4 + N_5)\cos\left(\frac{2\pi}{p} - \beta\right) \leq 0, \\
g_9(x) &= N_4 - N_6 + 2N_5 + 2\delta_{56} + 4 \leq 0, \\
g_{10}(x) &= 2N_3 - N_6 + N_4 + 2\delta_{34} + 4 \leq 0, \\
h_1(x) &= \frac{N_6 - N_4}{p} = \text{integer}.
\end{aligned} \tag{4.15}$$

其中， $\beta$  的表达式如公式（4.16）：

$$\beta = \frac{\cos^{-1}((N_4 + N_5)^2 + (N_6 - N_3)^2 - (N_3 + N_5)^2)}{2(N_6 - N_3)(N_4 + N_5)}, \tag{4.16}$$

变量的取值范围为：

$$\begin{aligned}
p &= (3, 4, 5), m_1, m_2 = (1.75, 2.0, 2.25, 2.5, 2.75, 3.0), 17 \leq N_1 \leq 96, \\
14 \leq N_2 \leq 54, 14 \leq N_3 \leq 51, 17 \leq N_4 \leq 46, 14 \leq N_5 \leq 51, 48 \leq N_6 \leq 124
\end{aligned}$$

ECCHLO 算法与 HLO、BPSO、BDE 和 BBA 算法的实验结果如表 4-8。

表 4-8 各算法于行星轮系设计优化问题的结果比较

| 算法     | 最优决策变量 |       |       |       |       |       |       |       |       | 最优值           |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|---------------|
|        | $X_1$  | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ |               |
| ECCHLO | 19     | 18    | 27    | 20    | 33    | 119   | 2.75  | 2     | 4     | <b>0.2355</b> |
| HLO    | 39     | 24    | 17    | 30    | 15    | 87    | 2.25  | 2     | 4     | 0.2384        |
| BPSO   | 46     | 20    | 14    | 37    | 31    | 107   | 2.5   | 1.75  | 5     | 0.2431        |
| BDE    | 47     | 35    | 25    | 33    | 21    | 105   | 1.75  | 1.75  | 3     | 0.3312        |
| BBA    | 18     | 16    | 14    | 17    | 14    | 50    | 1.75  | 1.75  | 3     | 0.2882        |

表 4-8 展示了 ECCHLO 算法与其他对比算法的结果数据。表明在相同条件下，ECCHLO 在处理行星轮系设计优化问题能够实现传动比的最大误差最小，最优值达到 0.2355，减小了行星轮的运作误差，能够准确地优化齿轮参数，提高系统的稳定性和可靠性，实现更高的传动效率，避免了算法可能出现的早熟收敛问题，从而确保优化结果的稳定性和精确性。

以上两个工程应用问题表明 ECCHLO 算法能够更好的在满足约束条件的前提下找到更优解,展现了算法在开发与探索间的平衡能力。因此,将 ECCHLO 算法应用于实际工程应用问题,能够优化结构设计,减少材料浪费并提高建筑的稳定性和安全性等,并且能够帮助解决复杂问题、优化流程。

### 4.3 基于 HSV 颜色模型的炉膛火焰识别方法研究

现代工业生产中,炉膛火焰的稳定性与效率直接影响到能源的利用效率和排放控制。因此,准确、实时地识别炉膛中的火焰状态对于优化生产过程、提高安全性和降低环境污染至关重要。近年来,随着科技的迅猛发展,基于优化算法的火焰识别方法逐渐成为研究的热点。

合作竞争机制是指在群体学习中,通过个体之间的合作竞争来优化学习效果。通过集体智慧,个体能够在多变的环境中进行有效的学习和决策,通过合作竞争的优势,引入信息交互式合作竞争机制的粒子群优化算法可以有效提升模型的鲁棒性和准确性。不仅提高了识别的准确率,还能在动态环境中保持较高的适应性,能够在复杂和动态的工业环境中实现高效、准确的火焰识别,为提高工业生产的安全性和环保性做出重要贡献。未来的研究可以进一步探索该算法在其他工业应用中的推广与发展,为智能制造领域的创新提供新的思路与方向。

#### 4.3.1 HSV 颜色模型原理

在计算机图像处理和视觉领域,RGB(红、绿、蓝)色彩空间是最常见的颜色表示方式。然而,RGB 色彩空间虽然在硬件显示和计算上广泛应用,但在一些实际应用场景中并不总是最优的选择。相较于 RGB 色彩空间,HSV(色调、饱和度、明度)色彩空间更适合在图像处理中进行颜色分割、筛选或分类任务。在 RGB 空间中,颜色分布并不线性,两个看似相似的颜色在 RGB 空间中的距离可能差异较大。而在 HSV 空间中,色调通道可以用来区分不同类型的颜色,饱和度和明度通道则可以控制颜色的纯度和亮度,从而更容易实现颜色的分割和识别。比如,在目标检测或对象识别中,通过 HSV 空间的色调范围,可以轻松提取出特定颜色的物体,这在 RGB 空间中则往往需要复杂的计算和转换。HSV 空间在亮度变化的情况下表现更加稳定。RGB 空间中,当亮度变化时,三个通道的值都会受到影响,导致颜色的表现发生较大改变。而在 HSV 空间中,明度的变化只影响 Value 通道,色调和饱和度通道仍然保持不变,这使得在火焰强度条

件不同的情况下，颜色处理更加稳定和可靠。因此，可将图 4-1 中燃烧火焰的色彩信息分成色调、饱和度、明度三种分量。

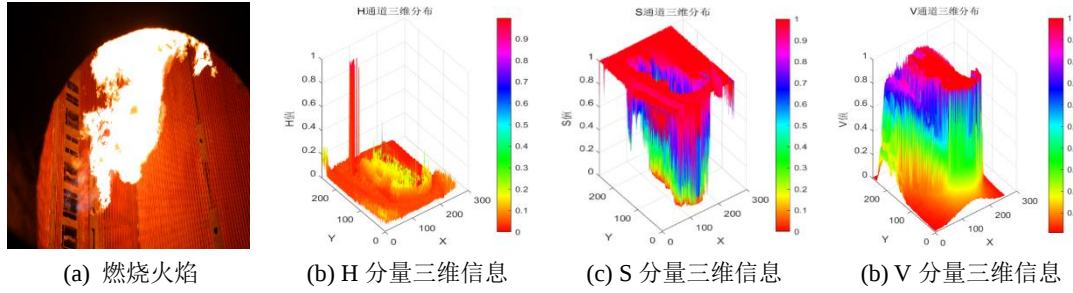


图 4-1 燃烧火焰 HSV 分量的三维信息分布效果图

通过对燃烧火焰的 HSV 分量进行三维信息分布分析，发现 H 分量在火焰特征的表现上相对不明显，其变化较为平缓，难以有效区分火焰与背景之间的差异。相反，S（饱和度）和 V（亮度）分量在火焰的颜色和亮度变化上具有更加显著的特征，通过两种分量的阈值分割进行火焰的识别工作，当 S 分量和 V 分量均满足所设定条件，即可判断该像素点为火焰像素点。因此，选择 S 分量和 V 分量作为阈值选择依据，可以更有效地提取火焰区域，从而提升火焰识别的精度和鲁棒性。

#### 4.3.2 HSV 颜色模型构建

由于现实图片均为 RGB 颜色空间图片，因此需要通过颜色空间转换，将 RGB 空间转为 HSV 空间。RGB 空间转 HSV 空间的步骤可以通过以下几个公式进行，其中 RGB 的三个参数的取值范围是[0, 255]，HSV 的取值范围是：色调（Hue）[0° , 360° ]，饱和度（Saturation）和明度（Value）[0, 1]。转换过程包括以下步骤：

将 RGB 值转换到[0, 1]范围：

$$R' = \frac{R}{255}, G' = \frac{G}{255}, B' = \frac{B}{255} \quad (4.17)$$

其中，R, G, B 为原始 RGB 值，R', G', B' 为归一化后的值。

计算归一化 RGB 值中的最大值  $C_{\max}$  和  $C_{\min}$ ，并得到差值  $\Delta$ ：

$$C_{\max} = \max(R', G', B') \quad (4.18)$$

$$C_{\min} = \min(R', G', B') \quad (4.19)$$

$$\Delta = C_{\max} - C_{\min} \quad (4.20)$$

如果  $\Delta=0$ （即所有颜色值相等），则色调为  $0^\circ$ ，呈现出灰色。否则，根据不同的  $C_{max}$ ，使用以下公式：

$$H = \begin{cases} 0 & \text{if } \Delta = 0 \\ 60^\circ \times \left( \frac{G' - B'}{\Delta} \bmod 6 \right) & \text{if } C_{max} = R' \\ 60^\circ \times \left( \frac{B' - R'}{\Delta} + 2 \right) & \text{if } C_{max} = G' \\ 60^\circ \times \left( \frac{R' - G'}{\Delta} + 4 \right) & \text{if } C_{max} = B' \end{cases} \quad (4.21)$$

并保证色调  $H$  的范围是  $[0^\circ, 360^\circ]$ 。

$$H = H + 360^\circ \quad (4.22)$$

饱和度的计算依据  $C_{max}$  的值。如果  $C_{max} = 0$ ，表示颜色为黑色，此时饱和度为 0。否则，使用以下公式：

$$S = \begin{cases} 0 & \text{if } C_{max} = 0 \\ \frac{\Delta}{C_{max}} & \text{else} \end{cases} \quad (4.23)$$

其中，饱和度  $S$  的范围是  $[0,1]$ 。

明度  $V$  直接等于最大通道值  $C_{max}$ ：

$$V = C_{max} \quad (4.24)$$

其中，明度  $V$  的范围是  $[0, 1]$ 。通过以上公式，将火焰样本像素从 RGB 空间向 HSV 空间转化。

在火焰检测和图像分割任务中，将图像转换到 HSV（色调、饱和度、亮度）颜色空间，并对不同通道进行区间阈值筛选。HSV 空间比 RGB 空间更符合人眼对颜色的感知，因此 HSV 的各通道能更有效地区分出火焰与背景，从而生成一个精准的火焰区域。

利用 IPSOCC 算法通过优化  $S$  通道和  $V$  通道的阈值，来实现火焰与非火焰像素的分类。基于火焰在 HSV 颜色空间中的显著特性，IPSOCC 算法对  $S$  通道和  $V$  通道进行逐步优化，通过调整阈值区间精确分离火焰与非火焰像素。火焰像素通常具有较高的亮度和饱和度，以及特定的色调范围，而非火焰像素则分布在不同的阈值之外。通过 IPSOCC 算法的引入能够有效降低背景噪声和环境光干扰，为火焰检测的准确性和实时性提供可靠保障，并进一步提升火焰像素分类的鲁棒性和性能。

给定火焰区域的 HSV 值范围，通过具有合作竞争机制的人类学习算法计算色调、饱和度和亮度的上下限， $[low_s, high_s]$  表示饱和度 S 的上下限， $[low_v, high_v]$  表示饱和度 V 的上下限，对于每个像素  $(i, j)$ ，S 分量和 V 分量都在对应的阈值范围内，则认为该像素属于火焰区域，否则为非火焰区域。

$$A(i, j) = \begin{cases} 1, & \text{if } low_s \leq S(i, j) \leq high_s \text{ and } low_v \leq V(i, j) \leq high_v \\ 0, & \text{else} \end{cases} \quad (4.25)$$

最终，生成一个二值化图像，其中火焰区域的像素被标记为 1，背景区域的像素标记为 0。达到通过阈值的方式对火焰样本信息与非火焰样本信息进行分类，以便后续操作。

通过改进算法迭代的最优 DA 值所对应的 S 通道和 V 通道阈值进行提取，作用于源火焰图片的 S 通道和 V 通道，对源火焰图像进行通道过滤，将符合双阈值范围的像素置为 1，否则为 0。

$$\begin{aligned} \Pi_s(i, j) &= \Pi(S(i, j) \in [low'_s, high'_s]) \\ \Pi_v(i, j) &= \Pi(V(i, j) \in [low'_v, high'_v]) \end{aligned} \quad (4.26)$$

其中， $low'_s$ 、 $high'_s$ 、 $low'_v$ 、 $high'_v$  为经过算法优化后的各颜色空间阈值。并将源火焰图像中每个像素的 HSV 通道进行过滤得出矩阵  $\Pi_s(i, j)$ 、 $\Pi_v(i, j)$ ，将矩阵中像素所对应的值进行如下操作：

$$M(i, j) = \Pi_s(i, j) \cdot \Pi_v(i, j) \quad (4.27)$$

其中， $M(i, j)$  为对源火焰图像通过 IPSOCC 算法对 HSV 颜色空间阈值筛选出的识别结果。

#### 4.3.3 HSV 颜色模型的目标函数

火焰样本信息的提取是 IPSOCC 火焰识别系统的关键步骤之一。通过对火焰源图片的人工分析与标注，提取出火焰区域内的像素信息以及非火焰区域内的像素信息，通过对火焰图像手动提取火焰区域的方法，使得系统在开发的初期阶段或处理复杂的火焰场景时，能够提供更为精准的数据基础，为后续的算法训练与优化提供高质量样本支持，以建立准确的火焰识别模型，还能对系统自动化处理提供有效的对照验证。

提取的火焰像素样本矩阵的长为  $M$ ，高为  $N$ ，因此提取的火焰样本像素总数为  $M \times N$ ，非火焰像素样本总数为  $M \times N$ ，提取的总样本数为  $2 \times M \times N$ 。取样区域如图 4-2 所示：

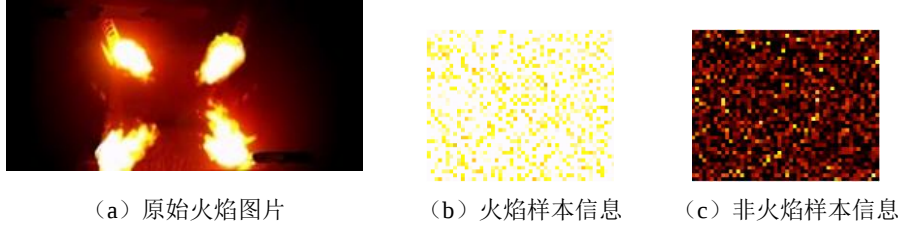


图 4-2 样本取样区域

通过 IPSOCC 算法分类的结果  $A$  与真实标签  $B$  进行对比，两个矩阵大小均为  $M \times N$ ，以计算分类结果与真实标签中的相同像素数为  $S$ ：

$$S = \sum_{i=1}^M \sum_{j=1}^N \Pi(A(i, j) = B(i, j)) \quad (4.28)$$

其中， $\Pi(A(i, j) = B(i, j))$  为指示函数，当  $A(i, j) = B(i, j)$  时取值为 1，否则为 0。

分类结果与真实标签中的不同像素数为  $D$ ：

$$D = \sum_{i=1}^M \sum_{j=1}^N \Pi(A(i, j) \neq B(i, j)) \quad (4.29)$$

其中， $\Pi(A(i, j) \neq B(i, j))$  为指示函数，当  $A(i, j) \neq B(i, j)$  时取值为 1，否则为 0。

通过计算相同像素数与不同像素数以计算分类实验结果与真实标签的相似度  $DA$ ：

$$DA = S / (S + D) * 100\% \quad (4.30)$$

通过与真实标签对比的相似度作为阈值优化的目标函数，作为优化结果的参考标准。通过构建合理的相似度目标函数，IPSOCC 算法能够在搜索过程中动态调整颜色阈值，使得最终火焰分类结果与真实标签的匹配度最大化，能够提高优化的准确性以及模型在不同数据分布下的泛化能力和鲁棒性。

#### 4.3.4 HSV 颜色模型的实现流程

本节主要基于 IPSOCC 算法，通过 HSV 颜色空间对火焰的颜色特征进行提取和分析。通过引入改进优化算法能够有效优化  $S$  和  $V$  通道的阈值，使火焰区域与非火焰区域的区分更加精准。本节首先介绍了基于 HSV 颜色空间的火焰像素样本获取方法，接着描述了改进算法的优化过程，最终通过优化的阈值对源火焰图像进行通道过滤，并完成火焰区域的检测。该方法为火焰检测提供了一种高效、精准的解决方案。具体流程如图 4-3 所示：

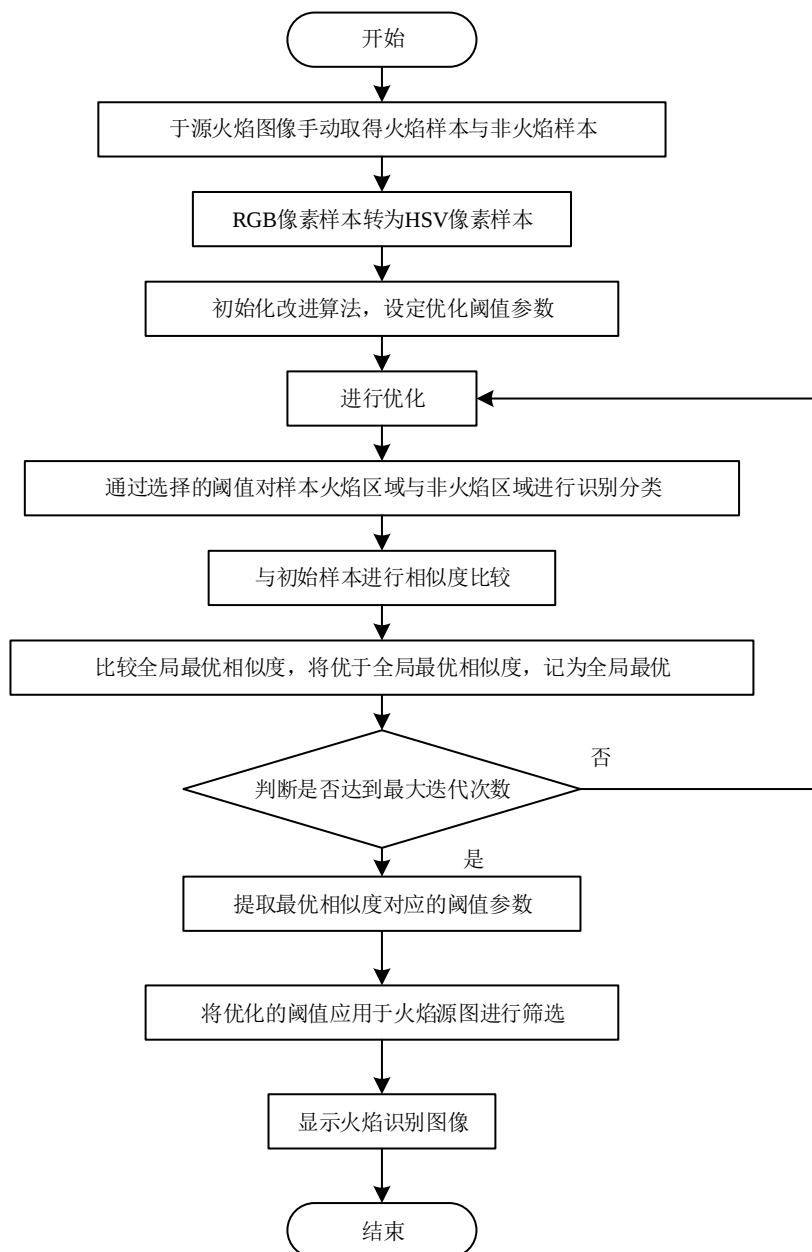


图 4-3 火焰识别流程图

#### 4.3.5 仿真实验与结果分析

为了验证基于群智能优化算法优化 HSV 阈值模型的炉膛火焰识别方法的有效性与可行性, 采用火焰图像进行仿真验证。通过 IPSOCC 算法优化 HSV 颜色模型 (命名为 IPSOCC-HSV), 计算得到最优的通道参数, 并与其他火焰识别模型进行对比, 如基于形态学操作的火焰识别方法<sup>[59]</sup> (MFD) 和基于 YCbCr 颜色分割模型<sup>错误!未找到引用源。</sup> (YCbCr CSM)。实验中, 采用关键性能指标如正确像素值、错误像素值、错误率 (ER) 和正确率 (DA) 来评估各模型的火焰识别效果。DA

代表正确识别的像素数与检测图像总像素数的比值，ER 则表示错误识别的像素数与检测总像素数的比值。

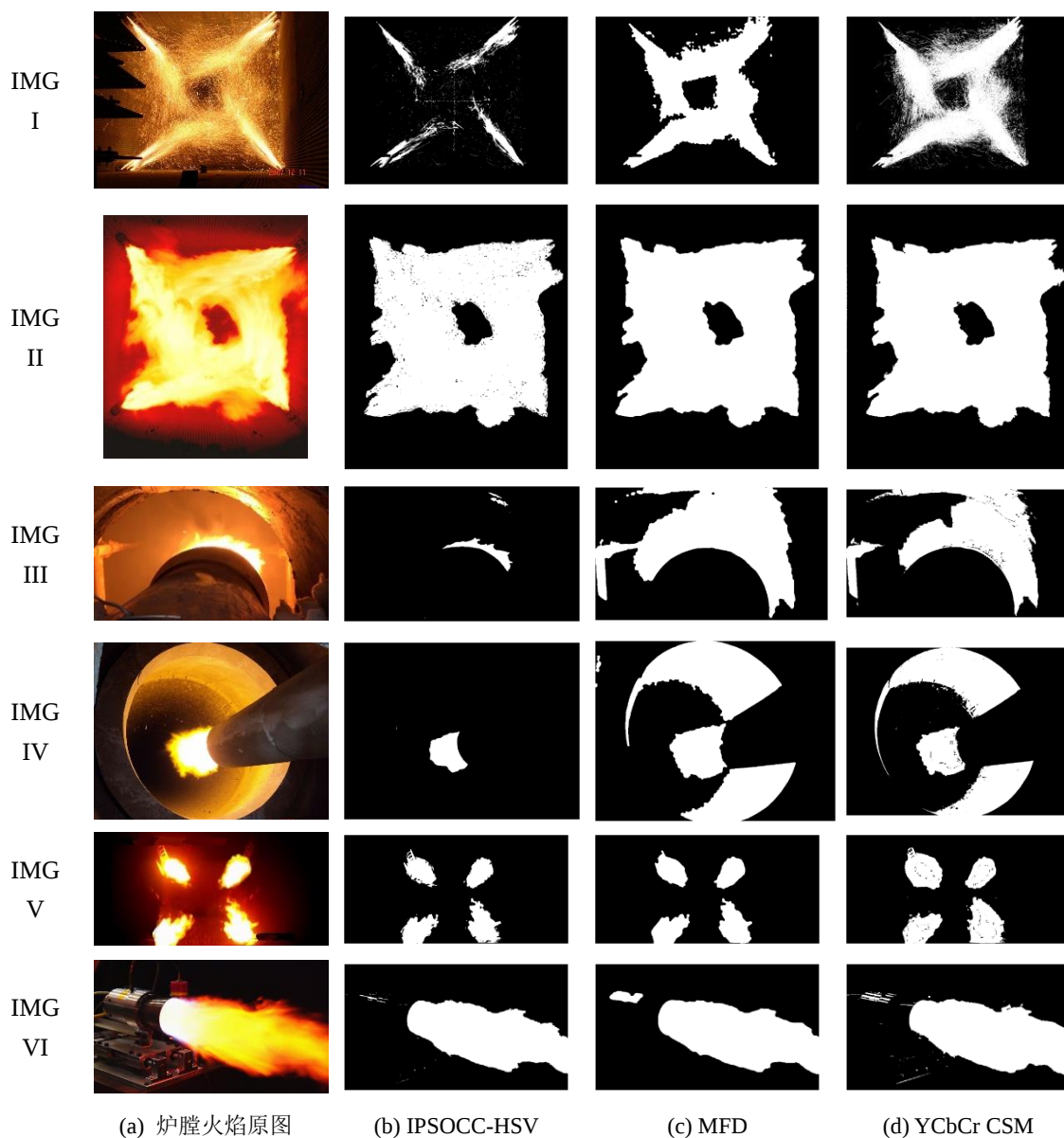


图 4-4 炉膛火焰识别实验结果

在火焰识别的应用实验中，对比了三种火焰识别模型。实验结果表明，IPSOCC-HSV 在火焰图像识别任务中表现出色，在处理绝大部分火焰分类问题均获得最高的相似度得分。相较于 YCbCr CSM 和 MFD 颜色识别模型，IPSOCC-HSV 在面对复杂火焰特征时，能够更好地利用多样化的搜索策略，减少误识别的情况。表明 IPSOCC-HSV 在处理火焰识别中存在的高变异性和复杂背景时，具有更强的精确性和鲁棒性。

表 4-9 炉膛火焰识别实验数据

| 图像    | 模型         | 评价指标         |                |                |                 |
|-------|------------|--------------|----------------|----------------|-----------------|
|       |            | 错误像素数        | 正确像素数          | ER             | DA              |
| 源图像 1 | MFD        | 305919       | 922881         | 24.8958%       | 75.1042%        |
|       | YCbCr CSM  | 358541       | 870259         | 29.1781%       | 70.8219%        |
|       | IPSOCC-HSV | <b>39275</b> | <b>1189525</b> | <b>3.1962%</b> | <b>96.8029%</b> |
| 源图像 2 | MFD        | <b>22613</b> | <b>677035</b>  | <b>3.2321%</b> | <b>96.7679%</b> |
|       | YCbCr CSM  | 25646        | 674002         | 3.6656%        | 96.3344%        |
|       | IPSOCC-HSV | 24468        | 675180         | 3.4972%        | 96.5028%        |
| 源图像 3 | MFD        | 155015       | 284485         | 35.2708%       | 64.7292%        |
|       | YCbCr CSM  | 125753       | 313747         | 28.6127%       | 71.3873%        |
|       | IPSOCC-HSV | <b>976</b>   | <b>438524</b>  | <b>0.2221%</b> | <b>99.7779%</b> |
| 源图像 4 | MFD        | 194646       | 554354         | 25.9874%       | 74.0126%        |
|       | YCbCr CSM  | 165701       | 583299         | 22.1230%       | 77.8770%        |
|       | IPSOCC-HSV | <b>313</b>   | <b>748687</b>  | <b>0.0418%</b> | <b>99.9582%</b> |
| 源图像 5 | MFD        | 25327        | 763153         | 3.2121%        | 96.7879%        |
|       | YCbCr CSM  | 51944        | 736536         | 6.5879%        | 93.4121%        |
|       | IPSOCC-HSV | <b>16819</b> | <b>771661</b>  | <b>2.1331%</b> | <b>97.8669%</b> |
| 源图像 6 | MFD        | 20290        | 210110         | 8.8064%        | 91.1936%        |
|       | YCbCr CSM  | 23252        | 207148         | 10.0920%       | 89.9080%        |
|       | IPSOCC-HSV | <b>16849</b> | <b>213551</b>  | <b>7.3129%</b> | <b>92.6871%</b> |

4.4 基于新型神经网络的炉膛火焰识别方法研究

传统的火焰检测方法多依赖于颜色模型和单一且固定的阈值分割技术，这些方法容易受到环境光照变化和背景复杂性等因素的干扰，导致检测结果的准确性较低。为了解决这些局限，基于神经网络的火焰检测方法逐渐受到广泛关注，特别是神经网络在处理图像分类和目标检测任务中的优秀表现。神经网络通过其深层次的非线性特征学习能力，能够有效提取图像中的复杂元素，从而提高火焰识别的准确性。然而，神经网络的性能极大依赖于网络参数（如权重和偏置）的有效训练，因此，如何优化这些参数以提高分类准确率和模型鲁棒性，成为提升火焰检测效果的关键。

本文将 ECCHLO 算法应用于神经网络的权重与偏置优化。通过 ECCHLO 算法对神经网络的参数进行优化，使优化后的神经网络在火焰图片分类任务中具备更高的分类精度和鲁棒性。

#### 4.4.1 新型神经网络模型构建

在传统的 BP 神经网络架构中，输入层节点通过隐藏层节点逐步将信息传递至输出层节点，从而完成非线性映射的学习和表达。然而，这种架构在特定情况下可能存在信息传递路径冗长、关键特征被隐藏层滤波而导致学习效率降低等问题。为此，本文研究了一种改进的 BP 神经网络模型，该模型在常规网络架构的基础上，额外增加了直接连接输入层与输出层的节点路径。通过这一直接连接，网络能够将部分关键输入信息绕过隐藏层，直接传递至输出层节点，从而增强对输入数据重要特征的捕捉能力。新型神经网络模型如图 4-5 所示：

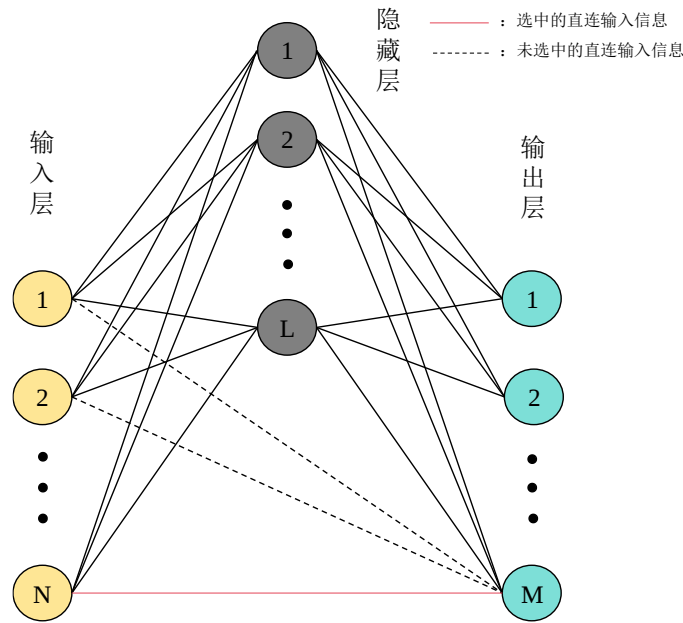


图 4-5 新型神经网络模型

新型网络结构的引入具有多方面优势，通过直接连接的输入输出路径为网络提供了更短的传播通道，使得关键特征无需完全依赖隐藏层的变换与提取，直接影响输出结果。该结构能够在减少过拟合风险的同时，提升模型在低样本场景下的泛化能力和训练效率。此外，这种结构在一定程度上也拓宽了网络的设计空间，使其能够适应更复杂的任务需求。相较于传统 BP 神经网络，该新型模型能够在多个测试任务中均表现出更高的预测准确性和学习稳定性。

神经网络的前向传播过程首先从输入层到隐藏层，将处理的输入数据  $X$  通过第一层权重矩阵  $W_1$  和偏置向量  $b_1$  映射到隐藏层，隐藏层的线性变换公式如下：

$$Z_1 = X \cdot W_1 + b_1 \quad (4.31)$$

通过激活函数引入非线性公式如下：

$$A_1 = \tanh(Z_1) \quad (4.32)$$

其次，从隐藏层到输出层，隐藏层输出  $A_1$  通过第二层权重矩阵  $W_2$  和偏置向量  $b_2$  映射到输出层。输出层的线性变换公式如下：

$$Z_2 = A_1 \cdot W_2 + b_2 \quad (4.33)$$

通过 Sigmoid 激活函数计算神经网络的输出  $y_1$ ：

$$y_1 = \sigma(Z_2) = \frac{1}{1 + e^{-z}} \quad (4.34)$$

从输入层到输出层的线性变换公式如下：

$$Z_3 = X \cdot W_3 \quad (4.35)$$

将两个数值进行整合得到最终输出  $y$ ：

$$y = y_1 + Z_3 \quad (4.36)$$

将神经网络的输出  $y$  进行归一化：

$$\dot{y} = \frac{y - y_{\min}}{y_{\max} - y_{\min}} \quad (4.37)$$

通过输出  $\dot{y}$  进行分类决策，得到二值矩阵  $\dot{y}_{class}$ ：

$$\dot{y}_{class} = \begin{cases} 1, & \text{if } \dot{y} \geq T \\ 0, & \text{if } \dot{y} < T \end{cases} \quad (4.38)$$

其中， $\dot{y}_{class}$  中的元素 1 为识别出的火焰，0 为识别出的火焰背景，最后显示识别的火焰。

将 ECCHLO 算法对网络的参数进行优化。神经网络中层与层之间的权重和偏置作为优化的参数集合，神经网络的权重和偏置为  $\{W_1, W_2, W_3, \dots, W_L\}$  和  $\{b_1, b_2, b_3, \dots, b_L\}$ ，则优化目标是通过 ECCHLO 算法调整这些参数，即对以下集合进行优化。

$$\theta = \{W_1, W_2, W_3, \dots, W_L, b_1, b_2, b_3, \dots, b_L\} \quad (4.39)$$

通过 ECCHLO 算法调整这些参数以提高模型的性能。优化目标是最大化新型神经网络对火焰图像识别的准确性，其适应度函数由识别精度决定。

#### 4.4.2 新型神经网络模型的目标函数

使用神经网络分类火焰图片的目标是将每个像素点分类为火焰区域（1）非火焰区域（0）。将原始火焰图片转为 HSV 空间，因此处理后的图片矩阵包含三个通道，表示为：

$$I(i, j) = \{H(i, j), S(i, j), V(i, j)\} \quad (4.40)$$

每个像素位置  $I(i, j)$  都对应一个三维特征向量。标签矩阵的处理方式则是将包含火焰的像素置为 1，背景像素置为 0。

$$M(i, j) = \begin{cases} 1, & \text{if } I(i, j) \text{ 属于火焰区域} \\ 0, & \text{if } I(i, j) \text{ 属于背景区域} \end{cases} \quad (4.41)$$

神经网络通过学习输入特征与标签之间的映射关系，对每个像素点进行分类。令输入图片的宽度为（W），高度为（H），那么每个像素都具有 HSV 三个特征，随后将图片展平成特征矩阵  $X \in R^{(N \times 3)}$ ，其中 N 的大小为  $W * H$ 。

优化过程中，首先将火焰图像转换至 HSV 颜色空间，以充分利用 HSV 空间对火焰特征的表达能。图像的 HSV 特征经过神经网络模型处理后，输出分类结果  $y_{class}$  与真实标签  $I$  进行比较两个矩阵大小均为  $M \times N$ ，以计算分类结果与真实标签中的相同像素数为 S：

$$S = \sum_{i=1}^M \sum_{j=1}^N \Pi(y_{class}(i, j) = I(i, j)) \quad (4.42)$$

其中， $\Pi(A(i, j) = B(i, j))$  为指示函数，当  $A(i, j) = B(i, j)$  时取值为 1，否则为 0。

分类结果与真实标签中的不同像素数为 D：

$$D = \sum_{i=1}^M \sum_{j=1}^N \Pi(y_{class}(i, j) \neq I(i, j)) \quad (4.43)$$

其中， $\Pi(A(i, j) \neq B(i, j))$  为指示函数，当  $A(i, j) \neq B(i, j)$  时取值为 1，否则为 0。通过计算相同像素数与不同像素数以计算分类结果与真实标签的相似度 DA：

$$DA = S / (S + D) * 100\% \quad (4.44)$$

在 ECCHLO 算法中，每个个体对应于一个神经网络参数组合（权重和偏置）。算法通过迭代更新，探索权重和偏置的最优配置，目标是使得适应度函数（即识别精度）达到最大值，并最终获得性能最优的神经网络模型。

#### 4.4.3 新型神经网络模型的实现流程

研究了一种基于 ECCHLO 算法优化神经网络参数的火焰图像识别方法，其具体流程如图 4-6 所示。首先，从火焰图片中提取出火焰样本和非火焰样本，并将这些 RGB 图像样本转换为 HSV 颜色空间，以充分利用 HSV 空间在表达火焰

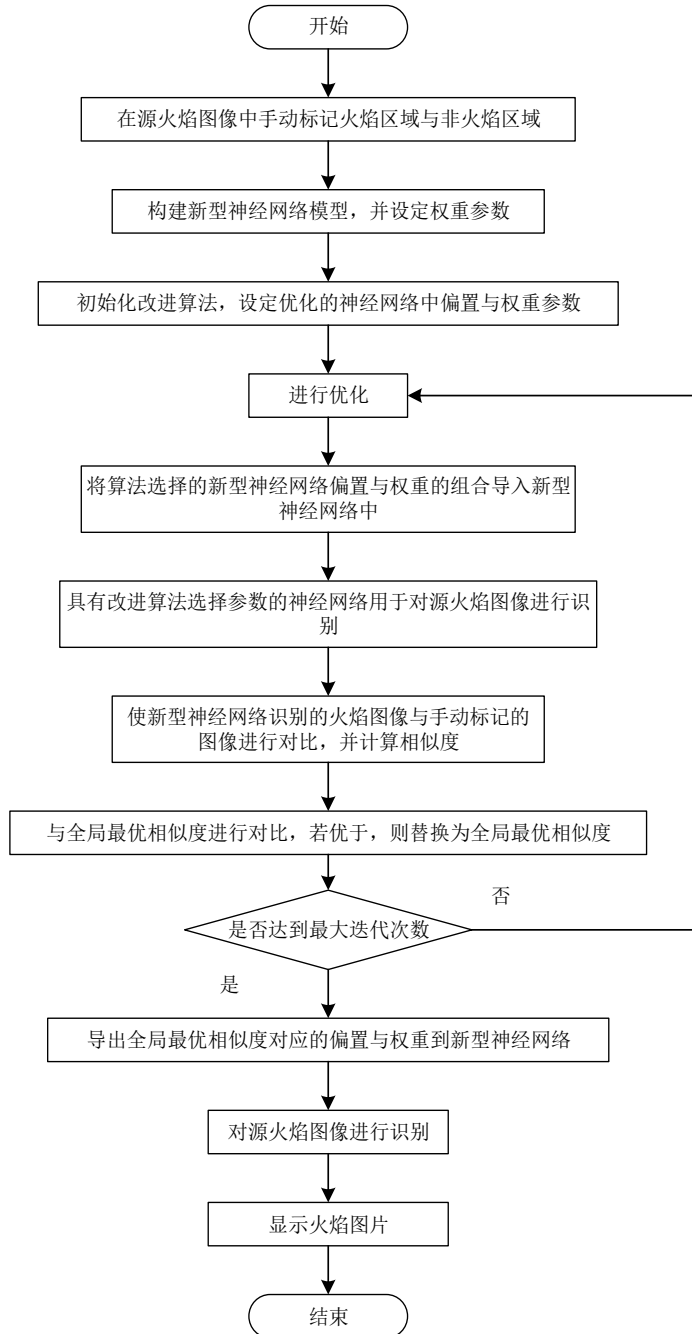


图 4-6 ECCHLO 算法优化新型神经网络用于火焰识别流程图

特征方面的优势。随后，初始化 ECCHLO 算法，并设置相关的优化参数，以便通过迭代过程不断优化神经网络的参数配置。在迭代过程中，ECCHLO 算法通过全局搜索策略有效避免局部最优解，最终确定一组能够最大化识别精度的网络

参数。优化完成后，得到的神经网络模型被应用于火焰识别任务，并输出相应的识别结果。该方法不仅确保了参数优化的系统性，还显著提升了火焰图像识别的准确性和鲁棒性。

#### 4.4.4 仿真实验与结果分析

为了验证 ECCHLO 算法优化新型神经网络在火焰图像分类中有效性。实验基于火焰图像，分别分析了 ECCHLO 优化新型神经网络模型(ECCHLO-NNNM)、基于 YCbCr 颜色分割模型、以及基于 BP 神经网络的火焰识别模型<sup>[61]</sup>三种模型的分类性能，并进行了对比。实验重点探讨了优化算法对神经网络权重和偏置初始化的影响，并通过火焰图像分类任务来评估各算法的分类精度和鲁棒性。在实验中，通过提供关键性能指标，包括正确像素值、错误像素值、错误率(ER)和正确率(DA)，评估所有方法的识别效果，当 DA 数值越大或 ER 数值越小时，火焰的识别精度越高、识别效果越好。

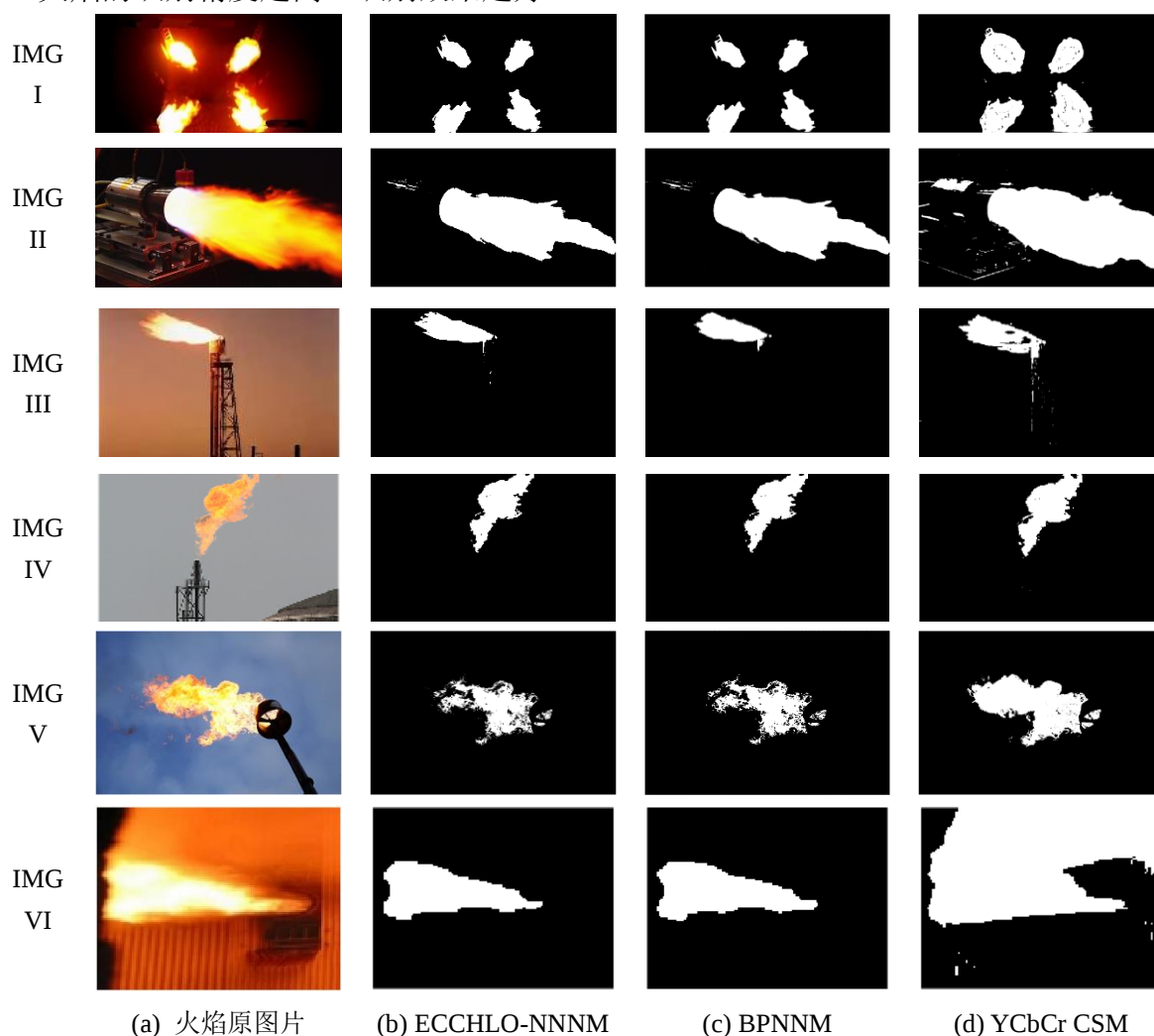


图 4-7 ECCHLO 优化新型神经网络用于火焰识别实验结果

结合图 4-7 与表 4-10 的实验结果，表明了 ECCHLO 算法在优化新型神经网络模型应用于火焰图像分类结果在 6 个火焰源图像中的 5 个比对比模型（BPNNM 模型和 YCbCr CSM 模型）表现更为优越。说明 ECCHLO-NNNM 在图像分类精度和处理问题的鲁棒性等关键性能指标上均取得更好的结果，能够有效地调整神经网络权重和偏置，实现更低的误差和更高的分类准确率，尤其在复杂火焰图像场景下表现出更强的适应性和稳定性。实验结果进一步验证了 ECCHLO 算法在优化新型神经网络和火焰图像分类领域的有效性和可行性，为实际应用提供了可靠的技术支持。

表 4-10 ECCHLO 优化新型神经网络用于火焰识别实验数据

| 图像    | 算法          | 评价指标        |               |                |                 |
|-------|-------------|-------------|---------------|----------------|-----------------|
|       |             | 错误像素数       | 正确像素数         | ER             | DA              |
| 源图像 1 | ECCHLO-NNNM | <b>6227</b> | <b>782253</b> | <b>0.7897%</b> | <b>99.2103%</b> |
|       | BPNNM       | 6325        | 782155        | 0.8022%        | 99.1978%        |
|       | YCbCr CSM   | 51944       | 736536        | 6.5879%        | 93.4121 %       |
| 源图像 2 | ECCHLO-NNNM | <b>1241</b> | <b>229159</b> | <b>0.5386%</b> | <b>99.4614%</b> |
|       | BPNNM       | 1406        | 228994        | 0.6102%        | 99.3898%        |
|       | YCbCr CSM   | 23252       | 207148        | 10.0920%       | 89.9080%        |
| 源图像 3 | ECCHLO-NNNM | <b>273</b>  | <b>199227</b> | <b>0.1368%</b> | <b>99.8632%</b> |
|       | BPNNM       | 541         | 198959        | 0.2712%        | 99.7288%        |
|       | YCbCr CSM   | 1815        | 197685        | 0.9098%        | 99.0902%        |
| 源图像 4 | ECCHLO-NNNM | <b>177</b>  | <b>139791</b> | <b>0.1260%</b> | <b>99.8740%</b> |
|       | BPNNM       | 209         | 139759        | 0.1493%        | 99.8507%        |
|       | YCbCr CSM   | 374         | 139594        | 0.2672%        | 99.7328%        |
| 源图像 5 | ECCHLO-NNNM | 16694       | 403332        | 3.9745%        | 96.0255%        |
|       | BPNNM       | 18073       | 401953        | 4.3028%        | 95.6972%        |
|       | YCbCr CSM   | <b>5013</b> | <b>415013</b> | <b>1.1935%</b> | <b>98.8065%</b> |
| 源图像 6 | ECCHLO-NNNM | <b>105</b>  | <b>11645</b>  | <b>0.8936%</b> | <b>99.1064%</b> |
|       | BPNNM       | 186         | 11564         | 1.5830%        | 98.4170%        |
|       | YCbCr CSM   | 4170        | 7580          | 35.4894%       | 64.5106%        |

4.5 本章小结

本章针对 ECCHLO 与 IPSOCC 在实际工程等问题中的应用进行了探讨，重点分析了两种改进算法在简单工程设计问题中的表现。此外，设计了两种炉膛火焰识别模型，并将所提出的算法应用于火焰识别问题。通过对比实验，验证了 ECCHLO 和 IPSOCC 在处理复杂工程优化和火焰识别任务中的有效性。实验结果表明，ECCHLO 和 IPSOCC 在提高设计精度、加速收敛以及避免局部最优解

方面表现优异，尤其在多目标优化和高维问题中具有更好的适应性。针对火焰识别问题，算法能够有效提高识别精度，并增强在不同工况下的鲁棒性，为实际工程应用提供了有力的支持。

## 第5章 总结与展望

### 5.1 总结

在飞速发展的时代背景下,优化问题诸多领域中变得愈发复杂和高维化,传统优化算法在面对这些高度非线性、强耦合以及动态变化的优化任务时,往往难以兼顾全局搜索能力和局部搜索精度。对此,为了解决 PSO 算法以及 HLO 算法在后期迭代过程中缺少种群多样性和容易陷入局部最优等问题,本文针对两种算法的缺点引入了合作竞争机制,研究了具有合作竞争机制的粒子群优化算法和具有合作竞争机制的人类学习优化算法,并将所提出算法分别应用于复杂模型优化的任务以及炉膛火焰识别方法,从而验证改进算法在处理复杂应用问题的求解能力,进一步促进了算法应用与发展。具体研究内容如下:

1) 研究了一种具有合作竞争机制的粒子群优化算法 (PSOCC), 通过引入粒子协同共享搜索策略和粒子竞争部署策略以及粒子停滞处理策略,提升了算法在求解空间的开发和挖掘能力。通过参数学习分析研究了 PSOCC 的参数组合,并通过 AD 曲线分析了其收敛性。在此基础上,进一步考虑搜索过程中种群对信息的利用效率差异特征,研究了一种信息交互式策略,并设计了具有信息交互式合作竞争机制的粒子群优化算法 (IPSOCC) 算法。实验结果表明, PSOCC 和 IPSOCC 算法在 CEC2005 基准函数上均表现出优异的性能,特别是在单峰和多峰函数的高维与低维问题中,展现了良好的稳定性。

2) 设计了一种合作竞争学习算子,并研究了具有合作竞争机制的人类学习优化算法 (CCHLO), 增强了算法的探索与开发能力。此外,引入的重新学习机制确保了 IKD 的多样性,提升了算法跳出局部最优的能力。并通过平均距离曲线对算法的收敛特性进行了分析。在此基础上,进一步探讨了合作竞争学习算子与社会学习算子在种群信息利用效率方面的差异,研究了一种增强探索与开采平衡合作竞争人类学习优化算法 (ECCHLO)。通过参数学习深入研究了影响算法性能的最佳参数组合。并将 CCHLO 和 ECCHLO 算法与其他对比算法应用于 CEC2005 测试集,以验证所提出的改进算法的优越性。实验结果表明,无论在处理低维还是高维问题 CCHLO 和 ECCHLO 算法相比于对比算法,具有更快的收敛速度和更强的精确度。

3) 研究了将 IPSOCC 和 ECCHLO 算法在工程设计问题中的应用, 分析了两种算法在简单工程设计中的表现。此外, 以炉膛燃烧场火焰识别问题为典型应用案例, 设计了两种炉膛火焰识别模型, 并将所提出的算法应用于火焰识别任务。实验结果表明, IPSOCC 和 ECCHLO 在提高设计精度、加速收敛和避免局部最优解方面表现优异, 尤其在复杂的工程设计问题中展现了更强的适应性。在火焰识别模型中, 算法有效提高了识别精度, 并增强了系统在不同工况下的鲁棒性。

## 5.2 展望

在研究了具有合作竞争机制群智能优化算法研究及其应用后, 本文展望了未来研究和应用的多个方向:

(1) 未来的研究可以集中在算法的自适应机制上。自适应机制能够根据当前搜索状态动态调整参数, 从而优化算法的性能。例如, 调整学习因子和惯性权重可以帮助算法在不同阶段平衡全局搜索与局部开发的能力。这种灵活性有助于在复杂的优化环境中更有效地找到最优解, 尤其是在面对动态变化的问题时。

(2) 结合其他智能技术, 如深度学习和强化学习, 可能会显著提升这两种算法的性能。深度学习能够帮助算法识别和学习复杂模式, 而强化学习的反馈机制可以促使算法在探索 and 开发之间进行更有效的权衡。这种跨学科的融合有望提高算法的适应能力, 处理更高维和更复杂的优化问题。

## 参考文献

- [1] Crespo Márquez A. The curse of dimensionality[M]//Digital Maintenance Management: Guiding Digital Transformation in Maintenance. Cham: Springer International Publishing, 2022: 67-86.
- [2] 陈光灿. 智能算法在结构优化中的应用[J]. 计算机科学与应用, 2024, 14(11): 11-20.
- [3] Kennedy J, Eberhart R. Particle swarm optimization[C]//Proceedings of ICNN'95-international conference on neural networks. ieee, 1995, 4: 1942-1948.
- [4] Chang W D. Multimodal function optimizations with multiple maximums and multiple minimums using an improved PSO algorithm[J]. Applied Soft Computing, 2017, 60: 60-72.
- [5] Wolpert DH, Macready WG (1997) Free lunch theorems for optimization[J/OL]. IEEE Trans Evolut Comput 1(1):67–82.
- [6] Krink T, Vesterström J S, Riget J. Particle swarm optimisation with spatial particle extension[C]//Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600). IEEE, 2002, 2: 1474-1479.
- [7] Kao YT, Zahara E. A hybrid genetic algorithm and particle swarm optimization for multimodal functions [J/OL]. Appl. Soft Comput., 2008, 8(2): 849-857.
- [8] Chai Z, Nwachukwu A, Zagayevskiy Y, Amini S, Madasu S. An integrated closed-loop solution to assisted history matching and field optimization with machine learning techniques[J/OL]. J. Petroleum Sci. Eng., 2021, 198: 108204.
- [9] Epitropakis MG, Plagianakos VP, Vrahatis MN. Evolving cognitive and social experience in particle swarm optimization through differential evolution: A hybrid approach[J/OL]. Inf. Sci., 2012, 216: 50-92.
- [10] Sayah S, Hamouda A. A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems[J/OL]. Appl. Soft Comput., 2013, 13(4): 1608-1619.
- [11] Kıran MS, Gündüz M, Baykan ÖK. A novel hybrid algorithm based on particle swarm and ant colony optimization for finding the global minimum[J/OL]. Appl. Math. Comput., 2012, 219(4): 1515-1521.

- [12] Eappen G, Shankar TJ. Hybrid PSO-GSA for energy efficient spectrum sensing in cognitive radio network[J]. *Phys. Commun.*, 2020, 40: 101091.
- [13] Van Laarhoven PJ, Aarts EH. Simulated annealing[M]. In: *Simulated Annealing: Theory and Applications*. Dordrecht, The Netherlands: Springer, 1987, pp. 7-15.
- [14] Liang JJ, Qin AK, Suganthan PN, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J/OL]. *IEEE Trans. Evol. Comput.*, 2006, 10(3): 281-295.
- [15] Chen YP, Peng WC, Jian MC. Particle swarm optimization with recombination and dynamic linkage discovery[J/OL]. *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, 2007, 37(6): 1460-1470.
- [16] Liang JJ, Suganthan PN. Dynamic multi-swarm particle swarm optimizer with local search[C]. In: *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 522-528.
- [17] Kennedy J, Mendes R. Population structure and particle swarm performance[C]. In: *Proceedings of the IEEE congress on evolutionary computation (CEC)*, Honolulu, HI. Piscataway: IEEE, 2002, pp. 1671-1676.
- [18] Liu Q, Wei W, Yuan H, et al. Topology selection for particle swarm optimization[J]. *Information Sciences*, 2016, 363: 154-173.
- [19] J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance[C], In: *Proceedings of IEEE Congress on Evolutionary Computation*, 1999, pp. 1931–1938.
- [20] P.N. Suganthan, Particle swarm optimiser with neighborhood operator, In: *Proceedings of IEEE Congress on Evolutionary Computation*[J], 2002, pp.1958–1961.
- [21] Wang L, Ni H, Yang R, et al. A Simple Human Learning Optimization Algorithm[J]. 2014:56-67.
- [22] Yang R, Fei M, Du X, Wang L, Pardalos P (2015) An adaptive simplified human learning optimization algorithm[C]. *Inf Sci Int J* 320:126–139.
- [23] Yang R, Xu M, He J, et al. An intelligent weighted fuzzy time series model based on a sine-cosine adaptive human learning optimization algorithm and its application to financial markets forecasting[C]//*Advanced Data Mining and Applications: 13th International Conference, ADMA 2017, Singapore, November 5–6, 2017, Proceedings 13*. Springer International Publishing, 2017: 595-607.

- [24] Wang L, An L, Pi J, Fei M, Pardalos PM. A diverse human learning optimization algorithm[J]. J Glob Optim (2017) .67(1–2):1–41.
- [25] Du J, Wen Y, Wang L, et al. An adaptive human learning optimization with enhanced exploration–exploitation balance[J]. Annals of Mathematics and Artificial Intelligence, 2023, 91(2): 177-216.
- [26] Wang L, Ni H, Yang R, et al. An adaptive simplified human learning optimization algorithm[J]. Information Sciences, 2015, 320: 126-139.
- [27] 张枫雪,刘勇. 广义最大覆盖模型的新型人类学习优化算法 [J]. 计算机仿真,2023,40(01):394-399.
- [28] 胡雪莲. 基于反馈控制的自适应人类学习优化算法及其应用[D]. 上海大学, 2021.
- [29] Du J, Wen Y, Wang L, et al. An adaptive human learning optimization with enhanced exploration–exploitation balance[J]. Annals of Mathematics and Artificial Intelligence, 2023, 91(2): 177-216.
- [30] 孟晗,马良,刘勇.融合学习心理学的人类学习优化算法[J].计算机应用,2022,42(05):1367-1374.
- [31] 曹倬铭,王文国.基于配对机制的人类学习优化算法研究[J].通信技术,2018,51(03):546-549.
- [32] Zhang P, Du J, Wang L, et al. A human learning optimization algorithm with reasoning learning[J]. Applied Soft Computing, 2022, 122: 108816.
- [33] Zhang P, Wang L, Fei Z, et al. A novel human learning optimization algorithm with Bayesian inference learning[J]. Knowledge-Based Systems, 2023, 271: 110564.
- [34] 杨韬,戴健.海鸥算法的改进及其在工程设计优化问题中的应用[J].数据与计算发展前沿,2022,4(06):129-144.
- [35] 祁明宏,任茂文.遗传算法在工程车辆悬架优化设计中的应用 [J]. 机械研究与应用,2009,22(06):5-8.
- [36] Celik T, Demirel H, Ozkaramanli H, et al. Fire detection using statistical color model in video sequences[J]. Journal of Visual Communication and Image Representation, 2007, 18(2): 176-185.
- [37] Chen X, Nguyen M, Yan W Q. Multiple flames recognition using deep learning[M]//Handbook of Research on Multimedia Cyber Security. IGI Global, 2020: 296-307.

- [38] Ma Z, Cheng Y, Wang H, et al. Research of flame image recognition algorithm based on svm[C]//2009 First International Conference on Information Science and Engineering. IEEE, 2009: 1399-1401.
- [39] Li W, Wang D, Chai T. Flame image-based burning state recognition for sintering process of rotary kiln using heterogeneous features and fuzzy integral[J]. IEEE Transactions on Industrial Informatics, 2012, 8(4): 780-790.
- [40] Strasser S, Goodman R, Sheppard J, et al. A new discrete particle swarm optimization algorithm[C]//Proceedings of the Genetic and Evolutionary Computation Conference 2016. 2016: 53-60.
- [41] Taşgetiren M F, Liang Y C. A binary particle swarm optimization algorithm for lot sizing problem[J]. Journal of Economic and Social Research, 2003, 5(2): 1-20.
- [42] Zolghadr-Asli B, Bozorg-Haddad O, Chu X. Crow search algorithm (CSA)[J]. Advanced optimization by nature-inspired algorithms, 2018: 143-149.
- [43] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-based systems, 2016, 96: 120-133.
- [44] Li YG, Wei YX. Research of rolling shear motion laws based on improved PSO[J]. International Journal of Control and Automation, 2015, 8(6): 99-112.
- [45] Shapiro A M. Without Miracles: Universal Selection Theory and the Second Darwinian Revolution[J]. BioScience, 1996, 46(11): 872-874.
- [46] Mayer R E. Cognition and instruction: Their historic meeting within educational psychology[J]. Journal of educational Psychology, 1992, 84(4): 405.
- [47] May M A, Leonard William D. Competition and cooperation[M]. Social Science research council, 1937.
- [48] Bonta B D. Cooperation and competition in peaceful societies[J]. Psychological bulletin, 1997, 121(2): 299.
- [49] Boyd R, Richerson P J, Henrich J. The cultural niche: Why social learning is essential for human adaptation[J]. Proceedings of the National Academy of Sciences, 2011, 108(supplement\_2): 10918-10925.
- [50] Chen Y, Xie W, Zou X. A binary differential evolution algorithm learning from explored solutions[J]. Neurocomputing ,2015,149, 1038–1047.

- [51] Mirjalili S, Mirjalili S.M., Yang, X.S. Binary bat algorithm[J]. Neural Comput. Appl,2014,25(3-4), 1-19.
- [52] Tang J, Liu G, Pan Q. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends[J]. IEEE/CAA Journal of Automatica Sinica, 2021, 8(10): 1627-1643.
- [53] SS V C. A multi-agent ant colony optimization algorithm for effective vehicular traffic management[C]//Advances in Swarm Intelligence: 11th International Conference, ICSI 2020, Belgrade, Serbia, July 14-20, 2020, Proceedings 11. Springer International Publishing, 2020: 640-647.
- [54] Li H, Shi N. Application of genetic optimization algorithm in financial portfolio problem[J]. Computational Intelligence and Neuroscience, 2022, 2022(1): 5246309.
- [55] Zou D, Gao L, Li S, Wu J. Solving 0-1 knapsack problem by a novel global harmony search algorithm[J]. Appl. Soft Comput,2011,11(2), 1556-1564.
- [56] Wang L, Ni H, Yang R. An adaptive simplified human learning optimization algorithm[J]. Inf. Sci, 2015, 320, 126-139.
- [57] Rao R V, Savsani V J, Vakharia D P. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems[J]. Computer-aided design, 2011, 43(3): 303-315.
- [58] Abderazek H, Sait S M, Yildiz A R. Optimal design of planetary gear train for automotive transmissions using advanced meta-heuristics[J]. International Journal of Vehicle Design, 2019, 80(2-4): 121-136.
- [59] Prabhu M K S, Prasanna S R M. Flame Detection using Morphological Operations[J]. International Journal of Image Processing, 2015, 9(1): 22-30
- [60] Premal C E, Vinsley S S. Image processing based forest fire detection using YCbCr colour model[C]//2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]. IEEE, 2014: 1229-1237.
- [61] 郭键,张涛.基于 BP 神经网络的仓储火灾火焰识别研究[J].数学的实践与认识,2015,45(07):190-198.

## 攻读学位期间发表的学术成果

- [1] **Qiliang Wang**, Pinggai Zhang, Yiheng Liu, et al. Novel cooperative particle swarm optimization with interactive learning strategy[C]//International Conference on Life System Modeling and Simulation. Singapore: Springer Nature Singapore, 2024: 235-249. (EI 收录)
- [2] **汪启梁**,张峰晔,张静怡,等.具有语音功能的跑步机自适应调速控制系统设计[J].佳木斯大学学报(自然科学版),2024,42(12):10-16.DOI:10.20232/j.cnki.jmsdxxb.2024.12.029.
- [3] 张平改, **汪启梁**. 工业炉膛燃烧场火焰识别与性能评测软件,2025-2-14, 中国,2025R11S0017131. (软件著作权, 授权)
- [4] 张平改,吴其林,胡保玲,魏利胜,**汪启梁**. 基于推理人类学习优化算法的工业炉膛温度预测系统,2025-2-28,中国,202411102957.7. (发明专利, 授权)

## 攻读学位期间获得的奖励

- [1] 获得 2024-2025 学年安徽工程大学研究生**三等奖学金**
- [2] 获得 2023-2024 学年安徽工程大学研究生**三等奖学金**
- [3] 获得 2022-2023 学年安徽工程大学研究生**一等奖学金**
- [4] 获得安徽省大学生创新大赛(2024)总决赛高教主赛道铜奖

## 致 谢

岁月如梭，转瞬间我的研究生生涯已接近尾声。值此之际，我怀着无尽的感恩之情，向所有在我的学术旅程中给予帮助与支持的每一位表示最诚挚的谢意。

“慈母手中线，游子身上衣。”在我人生的旅程中，父母始终是我坚实的后盾。无论我遇到什么困难，父母总是默默地为我排忧解难，给予我无尽的鼓励 and 信任。父母的辛勤付出让我明白了责任和坚持的重要，他们的教诲让我学会了如何面对人生的挑战，如何从失败中汲取教训，如何在困境中找到希望。是他们无私的爱和支持让我有勇气追逐梦想，勇敢前行。

“怀抱梦想行远路，桃李满园心自宽。”感谢安徽工程大学，感谢这片沃土给予我无限的启迪与成长。正是在这里，我得以与才俊为伍，共同追求真理，汲取知识的养分。学校为我提供了一个良好的平台，促使我不断拓展视野，提升自我。

“师者，传道授业解惑。”在此，我要特别感谢我的导师张平改老师与魏利胜老师以及校外导师刘小琿老师。老师们不仅在学术上给予我深刻的指导，更在为人处世上给予我诸多教诲。老师们严谨治学、求真务实的精神深深感染了我，每一次的指导都让我在科研的道路上更加坚定与前行。在老师们的悉心培养下，我在科研的道路上走得更加自信与稳健。

“同道中人，相知更相惜。”我还要特别感谢我的女朋友吴雪。她不仅在生活中给予我极大的帮助，也在学术上与我共同探讨，携手并肩，互相激励。她的鼓励与支持让我在艰难时刻不曾放弃，继续努力追寻我的目标。

“感恩之心常存，同行的路永记。”感谢所有曾给予我帮助的人，正是因为你们的支持和陪伴，我才走到今天，收获了丰富的学术成果。我会将你们的关爱与教诲铭记心中，在未来的道路上，不断进取，继续前行。我们江湖再见！

汪启梁

安徽工程大学电气工程学院

二〇二五年五月