

多策略改进状态基优化算法及其工程应用

郭 权, 曾钰清, 温子力

赣南科技学院智能制造与材料工程学院, 江西 赣州

收稿日期: 2026年3月1日; 录用日期: 2026年4月2日; 发布日期: 2026年4月13日

摘 要

本文提出一种多策略改进状态基优化算法, 用于改善原始状态基优化算法容易陷入局部最优、收敛速度差等问题。在初始化阶段引入Logistic-Tent混沌映射产生多样性更为丰富的种群; 在资源获取和资源评估阶段引入思维创新策略, 平衡算法的收敛精度和搜索能力。为测试改进算法的性能, 在CEC2022基准函数测试集中进行仿真测试, 实验结果证明了改进算法在收敛速度、求解精度和稳定性等方面得到了较大提升。此外, 选用了经典的弹簧设计问题进行对比仿真实验, 验证了算法具有优良的工程实用性。

关键词

状态基优化算法, 多策略改进, 弹簧设计问题

Multi-Strategy Improved State-Basis Optimization Algorithm and Its Engineering Application

Quan Guo, Yuqing Zeng, Zili Wen

School of Intelligent Manufacturing and Material Engineering, Gannan University of Science and Technology, Ganzhou Jiangxi

Received: March 1, 2026; accepted: April 2, 2026; published: April 13, 2026

Abstract

This paper proposes a multi-strategy improved state-based optimization algorithm to address the problems of local optima and poor convergence speed inherent in the original state-based optimization algorithm. In the initialization phase, a Logistic-Tent chaotic mapping is introduced to generate a more diverse population. In the resource acquisition and resource evaluation phases, innovative thinking strategies are introduced to balance the algorithm's convergence accuracy and

search capability. To test the performance of the improved algorithm, simulation tests were conducted on the CEC2022 benchmark function test set. Experimental results demonstrate that the improved algorithm achieves significant improvements in convergence speed, solution accuracy, and stability. Furthermore, a comparative simulation experiment was conducted using the classic spring design problem, verifying the algorithm's excellent engineering applicability.

Keywords

State-Based Optimization Algorithm, Multi-Strategy Improved, Spring Design Problem

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在现代科学与工程实践中, 优化问题无处不在且日益复杂。从电力系统中的最优潮流与经济调度问题[1][2], 到机器学习领域的超参数整定与大规模特征选择[3], 再到航空航天领域的飞行器轨迹规划与结构设计[4], 这些问题往往呈现出高维度、非线性、多约束以及多模态的特性。传统的数学规划方法在处理此类问题时常常面临维度灾难、依赖梯度信息以及易于陷入局部最优等困境, 难以满足实际应用的需求。

为了应对这些挑战, 研究者们从自然界和社会现象中汲取灵感, 开发了元启发式算法(Metaheuristic Algorithms, MAs) [5]。作为一类基于随机搜索的优化技术, 元启发式算法因其概念简单、不依赖问题梯度、易于实现等优点, 已成为解决复杂全局优化问题的强大工具。近年来, 元启发式算法领域涌现出大量新颖且高效的算法, 例如模拟海洋生物捕食行为的海洋捕食者算法(Marine Predators Algorithm, MPA) [6]、模仿哈里斯鹰协同狩猎的哈里斯鹰算法(Harris Hawks Optimization, HHO) [7]、以及模拟黏菌觅食过程的黏菌算法(Slime Mould Algorithm, SMA) [8]等。这些算法的核心在于通过有效的随机化算子和引导机制, 在全局探索(Exploration)和局部开发(Exploitation)之间取得动态平衡[9]。

状态基优化算法(Status-based Optimization, SBO)是一种新近提出的元启发式算法, 其通过模拟人类社会中个体追求更高社会地位的行为来进行寻优[10]。尽管 SBO 的设计新颖, 但在处理复杂优化问题时, 仍表现出部分元启发式算法的共性缺陷。特别是在迭代后期, 其种群多样性可能急剧下降, 导致算法收敛速度变慢, 且探索与开发之间的平衡被打破, 使得算法容易陷入局部最优解的困境, 从而影响其最终的求解精度和稳定性[11]。

为弥补 SBO 算法的不足, 本文提出一种融合了多种策略的改进状态基优化算法, 简称 ISBO。该算法通过在种群初始化与演化阶段引入增强机制, 显著提升了算法的全局搜索能力与鲁棒性。本文的主要贡献概括如下:

- (1) 针对原始 SBO 算法初始种群随机性不足的问题, 在初始化阶段引入 Logistic-Tent 混沌映射策略, 以生成空间分布更均匀、多样性更丰富的初始种群, 为算法的全局搜索奠定坚实基础。
- (2) 为防止算法陷入局部最优, 在资源获取和评估阶段引入新颖的思维创新策略(TIS)。该策略通过模拟人类的创新思维过程, 在算法搜索陷入局部最优时能够生成一个与当前种群关联度较低的新解, 为种群进化提供新的方向, 有效提升了算法的全局寻优能力。

为验证所提 ISBO 算法的有效性, 本文在 CEC 2022 国际公认的基准函数集以及经典工程设计问题

上, 将其与原始 SBO 及其他先进的元启发式算法进行了全面的仿真对比与分析。实验结果表明, 本文所提出的多策略改进方案在收敛速度、求解精度和鲁棒性方面均表现出显著优势。

2. SBO 算法的基本工作原理

SBO 算法的设计灵感来源于人类追求社会地位提升的内在驱动力。该算法通过模拟个体进入更高地位圈子, 并向高地位人群学习或获取资源的过程, 将这种社会模式转化为解决复杂优化任务的鲁棒计算方法。该算法巧妙地在全局探索和局部开发之间取得了平衡。算法的运行主要包括初始化、精英引领、资源获取和资源评估四个核心阶段。

2.1. 算法初始化

在初始化阶段, SBO 算法生成两个规模都为 $N \times D$ 初始种群, 分别记为 X^1 和 X^2 , 其中 N 代表种群中个体的数量, 即家庭的数量, D 代表问题的维度。在算法模型中, 具有相同的索引号 i 的两个个体 x_i^1 和 x_i^2 被视为同一个家庭的成员, 每个个体的初始状态通过以下公式在问题的搜索空间内均匀随机生成:

$$x(i, j) = U(lb_j, ub_j) \quad (1)$$

其中, $x(i, j)$ 是第 i 个个体的第 j 个决策变量, lb_j 和 ub_j 分别为该变量的下界和上界。初始化完成后, 算法会为每个家庭确定一个“精英成员”, 共同组成精英种群 X^e , 具体方法是比较同一家庭中两个成员的适应度值, 将表现更优的个体选为该家庭的精英, 该过程采用以下公式实现:

$$x_i^e = \begin{cases} x_i^1 & f_{obj}(x_i^1) < f_{obj}(x_i^2) \\ x_i^2 & f_{obj}(x_i^1) \geq f_{obj}(x_i^2) \end{cases} \quad (2)$$

其中, $f_{obj}(\cdot)$ 为目标函数。普通个体与其精英成员之间的互动模拟了以地位为导向的社会网络动态。

2.2. 精英引领

精英引领是 SBO 算法的核心探索阶段, 它模拟个体向社会中地位高的“导师”寻求指导以加速自身进步的过程。在这一阶段, 算法构建了一个动态的“高地位圈子”, 引导个体朝向更优的解空间移动。

首先, 算法采用轮盘赌选择法从精英种群 X^e 中选择一个精英个体 x_r^e , 这个被选中的精英 x_r^e 、当前种群中的最佳个体 x_b 以及正在更新的个体 x_i 共同定义了“高地位圈子”的影响范围。个体 x_i 将根据以下公式更新其位置, 向这个高地位圈子靠拢:

$$x_i' = \begin{cases} (1 - \omega_1 - \omega_2) * x_i + \omega_1 * x_r^e + \omega_2 * x_b & rand < \omega_3 \\ \omega_4 * ((1 - \omega_1 - \omega_2) * x_i + \omega_1 * x_r^e + \omega_2 * x_b) & rand \geq \omega_3 \end{cases} \quad (3)$$

其中, 权重系数 ω_1 、 ω_2 是通过 $randn$ 函数生成的正态分布随机数, 用于衡量个体自身、精英同伴和全局最优解的相对影响, 在引入随机性的同时确保个体的移动保持在合理边界内, 从而促进解空间中受控且多样化的搜索。参数 ω_3 、 ω_4 用于动态平衡调控探索与开发过程, ω_3 由下式决定:

$$\omega_3 = \tanh \left(\left(\frac{\sqrt{MaxFEs - randn * FEs}}{i} \right)^{\frac{FEs}{MaxFEs}} \right) \quad (4)$$

其中, $MaxFEs$ 是最大评估次数, FEs 是当前评估次数, 当随机数 $rand \geq \omega_3$ 时, 算法会启用公式(3)中的第二条规则, 此规则引入了一个缩放因子 ω_4 , 它在 $[-\omega_3, \omega_3]$ 区间内均匀随机取值, 该机制通过调整步长

显著增加了探索的多样性，有效防止算法陷入局部最优：

$$\omega_3 = \text{unifrnd}(-\omega_3, \omega_3) \quad (5)$$

2.3. 资源利用

资源获取阶段是算法从探索向利用过渡的关键环节。在此阶段，算法模拟个体获取和利用“社会资本”来提升自身。首先，算法为每个个体创建一个标志向量 $flag$ ，初始值均为 1，用于标记其在地位提升尝试中是否成功。

对于成功的个体(即在精英引领阶段找到更优位置的个体)，算法会选择性地整合来自家庭精英和全局最优个体两方面的影响。其资源更新方式如公式(6)所示，即对家庭精英 x_i^e 和全局最优解 x_b 的对应维度进行平均，从而获得新的位置 x_i^s ，具体过程由下式描述

$$x_{i,j_{idx}}^s = \frac{x_{i,j_{idx}}^e + x_{b,j_{idx}}}{2} \quad (6)$$

$$j_{idx} = \text{randn}(D) \quad (7)$$

其中，式(7)这一过程反映了家庭内部资源与外部顶级精英影响力的深度融合。相反，对于不成功的个体，它们将仅依赖家庭内部的资源进行改进。更新过程如公式(8)所示，通过一个随机生成的掩码向量 m 来决定哪些维度需要从其家庭精英 x_i^e 中学习，掩码向量 m 的生成方式由公式(9)决定。

$$x_{i,j}^s = x_{i,j}^e \quad \text{if } m_j = 1 \quad (8)$$

$$m(u(1:\text{ceil}(\text{randn} * D))) = 1 \quad (9)$$

2.4. 资源评估

资源评估是决策阶段，用于判断获取的资源是否真正提升了个体的适应度。该阶段利用之前设置的 $flag$ 向量来追踪个体的改进情况：1 表示适应度获得实质性提高(成功)，0 表示无改进(失败)。如果更新后的个体 x_i^s 的目标函数值优于原始状态 x_i ，则保留新状态：

$$x_i = x_i^s \quad \text{if } f_{obj}(x_i^s) < f_{obj}(x_i) \quad (10)$$

与此同时，该过程还将更新标志向量 $flag$ ，其由下式决定

$$flag_i = \begin{cases} 1 & f_{obj}(x_i^s) < f_{obj}(x_i) \\ 0 & f_{obj}(x_i^s) \geq f_{obj}(x_i) \end{cases} \quad (11)$$

这种选择性保留过程精确映射了现实世界中的社会地位晋升，从而推动算法持续向全局最优解收敛。

3. 多策略改进状态优化算法

3.1. Logistic-Tent 混沌映射策略

Logistic-Tent 混沌映射结合了 Tent 映射的遍历均匀性和 Logistic 映射的复杂混沌动力学特点，从而能够形成性能优越的混沌序列，其由下式定义：

$$x_{k+1} = \begin{cases} \frac{1}{2} * \left(r * x_k * (1 - x_k) + (4 - r) * \frac{x_k}{2} \right) & x_k < 0.5 \\ \frac{1}{2} * \left(r * x_k * (1 - x_k) + (4 - r) * \frac{1 - x_k}{2} \right) & x_k \geq 0.5 \end{cases} \quad (12)$$

其中, x_k 是第 k 次迭代的混沌值, r 为控制参数, 取值范围为 $(0,4)$ 。相较于单一的 Logistic 和 Tent 映射, Logistic-Tent 混沌映射产生的初始种群分布性更为均匀, 图 1 展示了三种混沌映射在迭代 1200 次时的空间分布与频数统计。由图可知, 单一的 Logistic 映射存在严重的“两极分化”现象(U 型分布), 导致初始种群过度集中于解空间边界; 而单一的 Tent 映射在有限计算机精度下, 为避免序列退化通常需牺牲控制参数, 这会导致其在解空间边缘产生覆盖缺陷。相比之下, Logistic-Tent 复合混沌映射不仅有效克服了 Logistic 映射中间搜索能力不足的问题, 同时避免了 Tent 映射的截断盲区。其生成的混沌序列在整个空间内分布最为广泛且相对均匀, 几乎不存在明显的覆盖盲区。因此, 采用 Logistic-Tent 混沌映射进行种群初始化, 能够为优化算法提供兼具高度多样性与全局空间遍历性的高质量初始解。

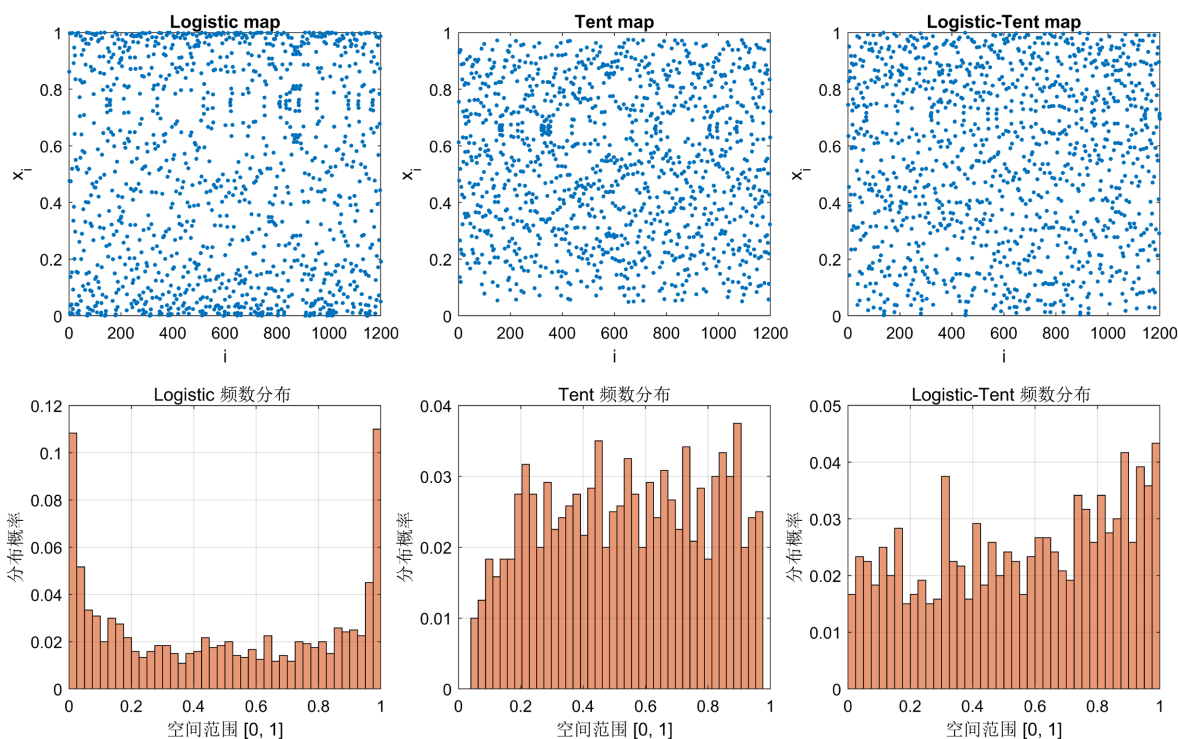


Figure 1. Comparison diagram of the three mappings

图 1. 三种映射的对比图

3.2. 思维创新策略

原始 SBO 算法在资源获取和评估阶段, 主要依赖于家庭精英和全局最优个体的信息, 这种方式虽然引导性强, 但当种群陷入局部最优时, 个体难以通过现有信息产生突破性的移动, 从而限制了算法的全局搜索能力。为了解决这一问题, 本文引入一种新颖的思维创新策略, 旨在模拟人类解决问题时的创新思维过程, 以增强算法的种群多样性和跳出局部最优的能力。思维创新策略(TIS)的灵感来源于人类的认知与创新过程, 策略示意图见图 2, 当面临复杂问题时, 人类并不仅仅依赖于固有的经验, 还会结合想象力与新知识来创造全新的解决方案。TIS 通过数学模型模拟了这一过程, 它主要包含三个核心概念: 信息事件 IE 、知识深度 DOK 、想象力 IM 。

TIS 策略并非直接寻找最优解, 而是基于信息事件 IE , 结合个体的知识深度(DOK)和想象力(IM), 构建一个全新的候选解, 从而为算法的进化提供全新的方向, 即: 一个新位置 pop_{new} 由当前个体位置 pop 、信息事件 IE 、知识深度 DOK 和想象力 IM 共同生成。

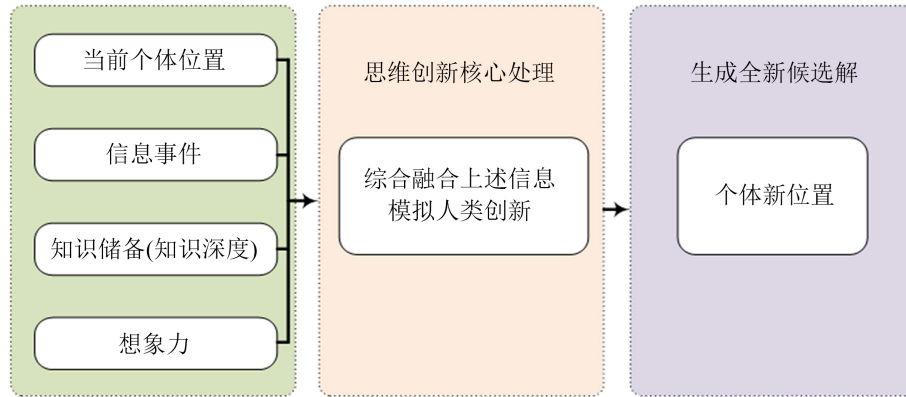


Figure 2. Diagram of thinking innovation strategies
图 2. 思维创新策略示意图

(1) 信息事件 IE

信息事件 IE 是一个动态更新的知识库，用于存储算法迭代过程中成功个体的历史信息。它代表了解决问题的“成功经验”，其更新表达式如下：

$$IE_{t+1} = \begin{cases} pop_{new} & f_{obj}(pop_{new}) < f_{obj}(IE_t) \\ IE_t & f_{obj}(pop_{new}) \geq f_{obj}(IE_t) \end{cases} \quad (14)$$

其中， IE_t 和 IE_{t+1} 分别是当前和下一代的信息事件。

从初始优势个体的选取出发， IE 通过算子交互引导种群向历史最优区域聚拢，并基于优胜劣汰准则实现参考点的动态修正。该机制实现了从随机搜索经验到高质量引导知识的转化，在显著抑制搜索盲目性的基础上，强化了算法的收敛精度与全局稳定性。

(2) 知识深度 DOK

知识深度 DOK 通过双阶段模型的有机融合实现寻优能力的动态平衡：前期采用线性增长模型(DOK_1)稳步积累基础知识，为搜索方向提供渐进式引导；后期借助高阶幂函数模型(DOK_2)模拟深度洞察的质变过程，实现末期的高效收敛。如式(15)所示，该设计兼顾了前期的全局广度探索与后期的精细深度开发，显著提升了算法的整体效能。

$$DOK = DOK_1 + DOK_2 \quad (15)$$

$$DOK_1 = C + \left(\frac{FEs}{MaxFEs} \right)^{0.5} \quad (16)$$

$$DOK_2 = FEs^{10} \quad (17)$$

其中， FEs 和 $MaxFEs$ 分别是当前和最大评估次数， C 一般取 0.5，这样一来，算法在启动时既保留了 50% 的探索随机，又注入了 50% 的经验引导。这种中庸的初始设定为算法提供了平稳的启动环境。

(3) 想象力 IM

想象力 IM 则是通过一个随机过程，模拟创新思维中的非确定性和跳跃性，其由下式决定：

$$IM = \pi * IE * rand \quad (18)$$

最终候选位置 pop_{new} 则由下式决定：

$$pop_{new} = \tan(IM - 0.5 * \pi) + \left(\frac{pop}{DOK} + IE \right) \quad (19)$$

式中，正切函数项 $\tan(IM - 0.5 * \pi)$ 作为核心创新算子，映射了“发散性思维”与“突破传统边界”的认知特征。正切函数由于其在 $\frac{\pi}{2}$ 附近的奇点特性，能够产生非线性的爆发式位移，使得个体能够突破当前局部吸引子的束缚。相比于 \sin 或 \exp 函数， \tan 函数通过将有限的认知区间 $(0, \pi)$ 映射到无限的搜索响应空间 $(-\infty, +\infty)$ 在数学上实现了从有限经验到无限可能的逻辑转换，这种特性在系统演化中有效地防止了种群的过早停滞。

3.3. ISBO 算法执行过程

本文提出的多策略改进状态基优化算法(ISBO)通过融合混沌映射与思维创新策略(TIS)，实现了一种宏观全局探索与微观局部开发高效协同的寻优框架。该算法首先利用混沌映射保障初始解空间的多样性；在迭代演化过程中，通过精英引领机制驱动种群进行大范围的全局寻优，并根据个体的社交成败状态执行差异化的资源重组以维持种群活力；同时，在位置更新与资源重组之后，算法两次穿插基于动态知识深度与历史最优记忆的 TIS 策略进行精细扰动，极大提升了局部深挖与跳出局部最优的能力；最终依靠贪婪选择与精英巩固机制实现种群的持续进化。其具体执行流程可归纳为以下七个步骤：

步骤 1：混沌初始化与参数设定。采用 Logistic-Tent 混沌映射生成初始种群，计算所有个体的初始适应度，筛选出局部精英与全局最优解。同时，构建用于记录历史优秀个体的 TIS 记忆库，并初始化所有个体的社交成败标志。

步骤 2：精英引领与位置更新。在精英引领阶段，驱动个体向高社会地位目标靠拢，通过个体间的状态交互与信息共享，引导种群完成大范围的全局探索，从而快速收敛至潜在的最优解区域。

步骤 3：第一次思维创新微调。对完成位置更新的个体应用 TIS 策略，结合随迭代次数动态累积的 DOK，并提 TIS 记忆库中的最优历史信息，利用非线性正切三角函数对个体当前位置施加精细扰动，开展深度的局部搜索。

步骤 4：资源获取与差异化演化。依据个体在上一轮迭代中的社交成功标志(Flag)，执行差异化的演化机制：社交“成功”的个体通过跨维度融合顶尖资源稳步提升；社交“失败”的个体则通过多维度复制局部精英个体的基因来实现位置重组，以此帮助陷入停滞的个体快速跳出局部最优，并维持种群的多样性。

步骤 5：二次思维创新微调：对历经上述资源重组后生成的新候选种群，再次引入 TIS 策略进行深度微调。此步骤旨在对高潜力解区域进行二次“开采”，最大化算法的局部寻优精度。

步骤 6：资源评估与贪婪选择。将微调后的候选个体与原个体进行适应度一对一对比，优胜劣汰替换最优位置并实时重置个体的社交成败标志。

步骤 7：巩固与全局更新：全面更新局部精英库与全局最优解，将最新的顶尖信息更新至 TIS 记忆库中，并记录收敛数据以推进下一轮迭代。

4. 实验分析

4.1. 标准测试函数集测试实验

4.1.1. 基准函数与对比算法

为检验改进算法的性能，采用 CEC2022 基准测试函数集开展仿真验证，具体函数信息详见表 1。该测试集由 12 个具有不同特征的函数组成，具体划分为四类：单峰函数(F1)、基础函数(F2~F5)、混合函数(F6~F8)以及组合函数(F9~F12)。此套函数测试集能够有效模拟现实工程应用中广泛存在的各类复杂优化环境，从而客观检验算法的综合求解能力。

Table 1. CEC 2022 function set
表 1. CEC2022 函数集

| 函数 | 名称 | 范围 | 最优值 |
|-----|--------------------------------------------------------------|-------------|------|
| F1 | Shifted and full Rotated Zakharov Function | [-100, 100] | 300 |
| F2 | Shifted and full Rotated Rosenbrock's Function | [-100, 100] | 400 |
| F3 | Shifted and full Rotated Expanded Schaffer's F6 Function | [-100, 100] | 600 |
| F4 | Shifted and full Rotated Non-Continuous Rastrigin's Function | [-100, 100] | 800 |
| F5 | Shifted and full Rotated Levy Function | [-100, 100] | 900 |
| F6 | Hybrid Function 1 ($N = 3$) | [-100, 100] | 1800 |
| F7 | Hybrid Function 2 ($N = 6$) | [-100, 100] | 2000 |
| F8 | Hybrid Function 3 ($N = 5$) | [-100, 100] | 2200 |
| F9 | Composition Function 1 ($N = 5$) | [-100, 100] | 2300 |
| F10 | Composition Function 2 ($N = 4$) | [-100, 100] | 2400 |
| F11 | Composition Function 3 ($N = 5$) | [-100, 100] | 2600 |
| F12 | Composition Function 4 ($N = 6$) | [-100, 100] | 2700 |

为了全面评估所提 ISBO 算法的寻优性能, 将其与原始 SBO 算法以及四种经典的群智能优化算法展开对比实验, 包括灰狼优化算法(GWO) [12]、蜣螂优化算法(DBO) [13]、粒子群算法(PSO) [14]和麻雀搜索算法(SSA) [15]。为保障仿真测试的公平性与可靠性, 将所有算法的种群规模统一设定为 $N = 30$, 最大迭代步长限制为 500 次。每项算法均在相同条件下独立重复执行 30 次。

4.1.2. 收敛性分析

在 12 种测试函数下, 6 种算法的收敛曲线如图 3 所示。从收敛曲线的走势可以清晰地看出, ISBO 算法在不同特征的测试基准上均展现出了卓越的寻优性能。具体而言, 单峰函数(F1)与基础函数(F2~F5)中, ISBO 展现出极为陡峭的初期下降斜率, 并以显著的优势收敛至极高精度, 验证了其极快的收敛速度与强悍的局部深度挖掘能力; 而在混合函数(F6~F8)以及复杂的组合函数(F9~F12)中, ISBO 不仅能迅速锁定优质解区域, 更能在其余多数算法陷入局部停滞时, 依然保持持续且平滑的向下探索动力。总体而言, ISBO 算法较好地协调了全局探索与局部开发的过程。相比所选取的传统及主流算法, ISBO 在收敛精度、初始收敛速度以及规避早熟收敛等方面均表现出明显的优势, 展现了较好的综合寻优潜力。

4.1.3. 性能分析

为了全面评估 ISBO 算法的综合竞争力, 本节对 ISBO 及其余 5 种对比算法在基准测试函数上的运行结果进行了深入的统计分析。具体而言, 本文选取了最优值(Best)、最差值(Worst)、平均值(Mean)、中位数(Median)及标准差(Std)作为核心评价指标。

然而, 单纯的数值罗列难以快速捕捉算法在跨函数、多指标下的整体优劣态势。为了更直观地展示各类算法在不同测试基准上的综合表现, 本文引入雷达图进行多维度的可视化分析。由于所考察的统计指标均属于极小化目标, 直接映射原始数据不仅会面临指标间量纲不统一的问题, 还会导致“算法性能越好, 雷达图覆盖面积反而越小”的视觉表达悖论。

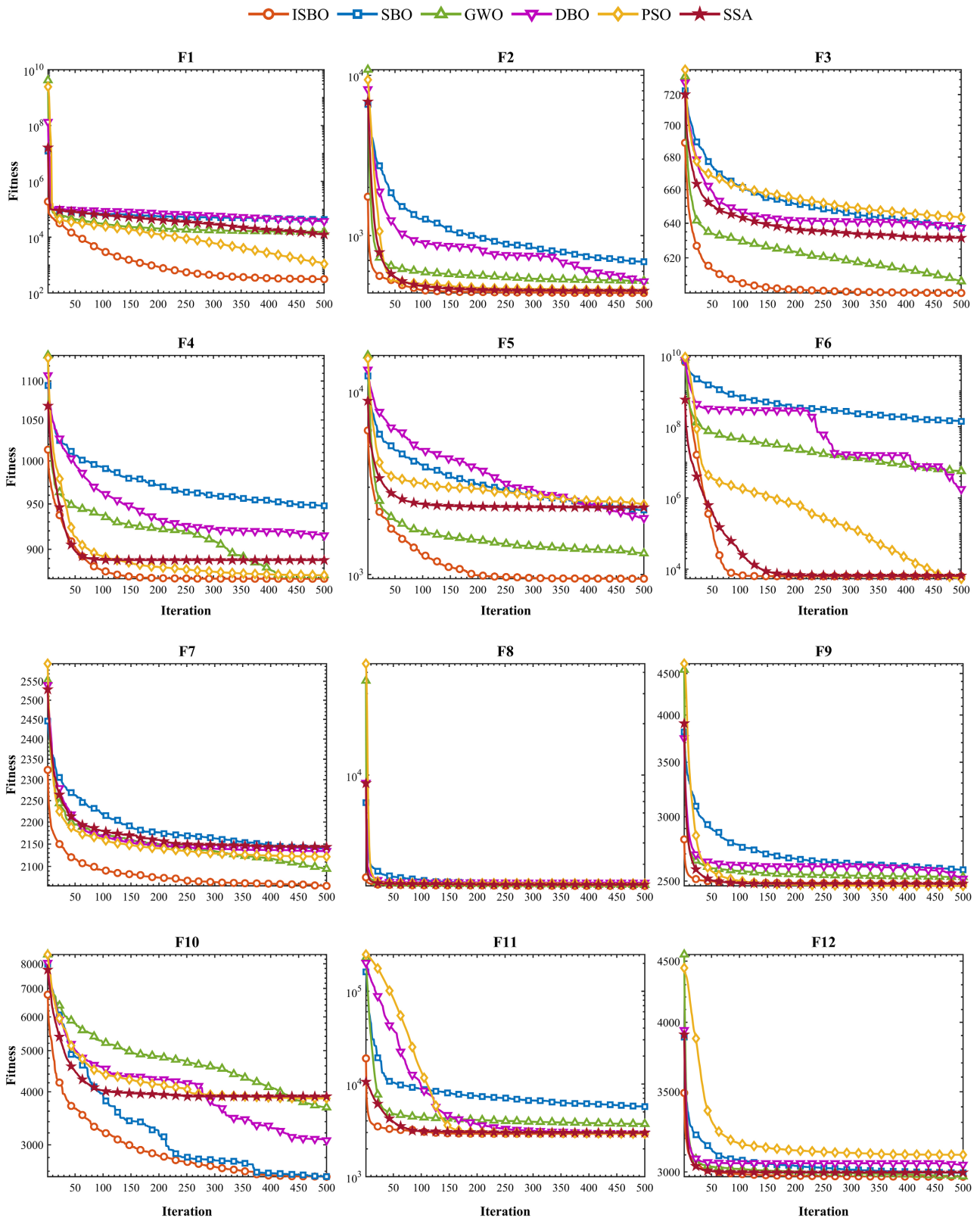


Figure 3. Convergence curve comparison chart
图 3. 收敛曲线对比图

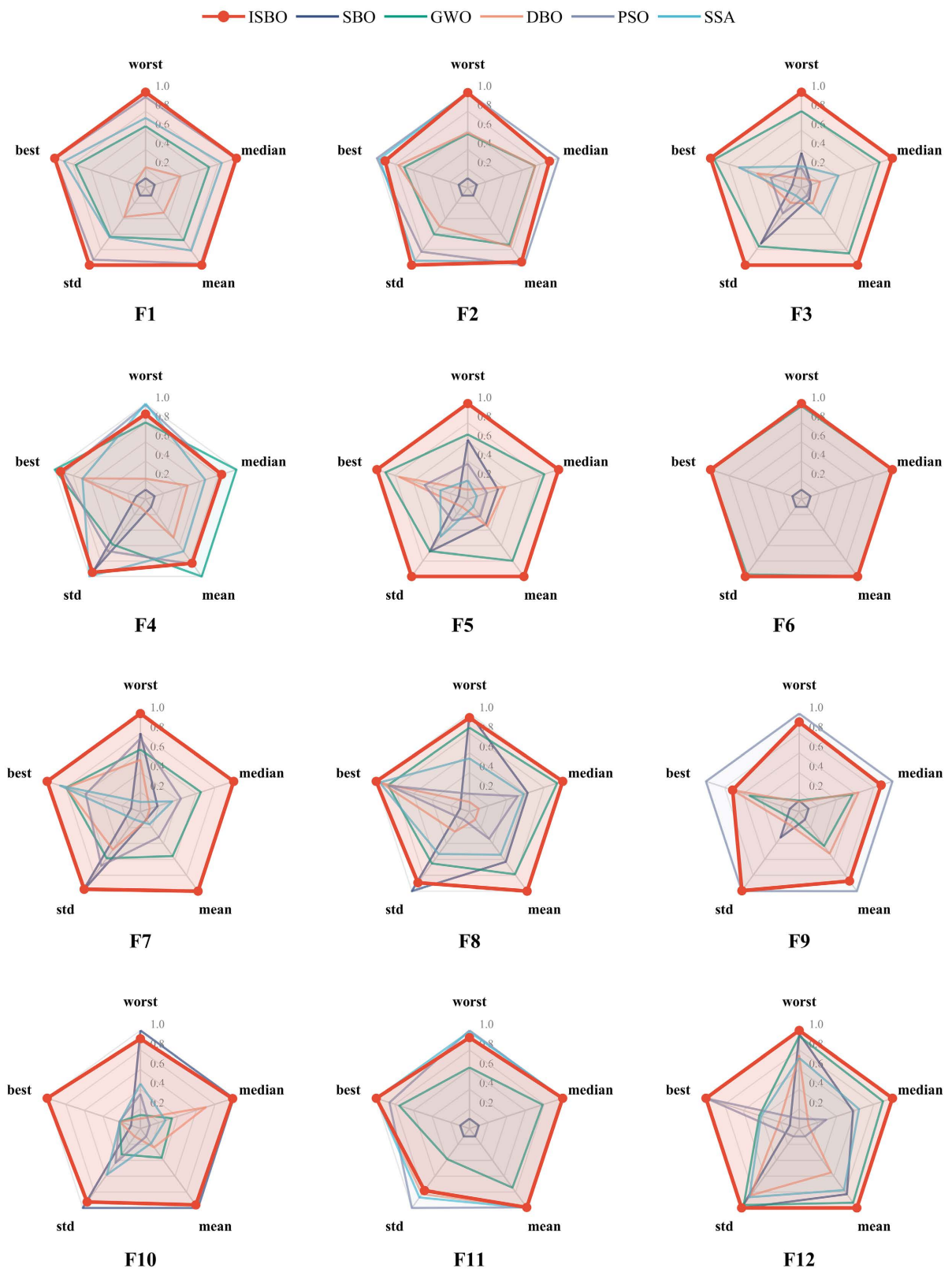


Figure 4. Test function indicator comparison radar chart
图 4. 测试函数指标对比雷达图

鉴于此,采用反向最大-最小归一化策略对实验结果进行预处理,将所有评价数据统一映射至[0.1, 1.0]的标准区间内。具体的转换模型如式(20)所示:

$$S_{ij} = \alpha + (1 - \alpha) \times \frac{X_{\max} - X_{ij}}{X_{\max} - X_{\min}} \quad (20)$$

式中, X_{ij} 代表第 j 种算法在第 i 项评估指标下的原始测试值; X_{\max} 与 X_{\min} 分别标识所有参评算法在该项指标下的全局最大值与最小值。 α 设定为非零的基础偏移量(本文取 $\alpha = 0.1$), 其目的在于规避最差性能点映射为零时造成的图形闭合畸变。

借由上述数据转换, 各项指标的理想极小值 (X_{\min}) 将被锚定在雷达图的绝对外缘(值为 1.0), 而相对最劣值 (X_{\max}) 则收敛于图心区域(值为 0.1)。以此为基础, 雷达图内所构成的多边形面积能够正向反映算法的寻优能力: 多边形面积越广阔, 即证实该算法在当前测试函数下的全局探索效能越优, 且鲁棒性更佳。

图 4 给出了 6 种算法在 CEC2022 基准函数(F1~F12)上的雷达图对比结果。通过反向归一化处理, 雷达图的绘图面积直观且正向地反映了算法的综合表现。

从整体分布来看, ISBO 算法(红线所示)在四类测试函数中均展现了极强的统治力, 其构成的多边形面积在大多数情况下显著大于 SBO、GWO、DBO、PSO 及 SSA。特别是在单峰函数 F1 与基础函数(如 F2、F5)中, ISBO 的多项统计指标(包含极值与方差)均逼近理想外边界, 体现了其寻优结果的高精度与低波动性。

此外, 在评估算法跳出局部最优能力的混合函数(F6~F8)与组合函数(F9~F12)中, 传统算法极易因早熟收敛导致多边形向图心收缩, 而 ISBO 依然能够保持极大的覆盖面积(尤以 F6 和 F9 最为显著)。这一结果进一步有力地证明了改进后的 ISBO 算法在复杂多模式搜索空间中, 能够有效平衡全局探索与局部开发, 具备更优的综合求解性能与极强的鲁棒性。

4.2. 经典工程设计问题测试实验

在完成基准函数测试后, 为进一步考察 ISBO 算法应对实际工程约束问题的求解能力, 本节选用经典的张力/压缩弹簧设计问题进行仿真验证。在统一参数设置的前提下, 各对比算法均独立执行 30 次, 并统计其寻优结果的最优值、平均值及标准差进行评估。

该问题是工程优化领域中一个经典的非线性约束优化基准案例。弹簧结构如图 5 所示, 它的核心目标是在严格满足弹簧工作时的各项物理与力学性能约束的前提下, 通过对导线直径(d)、平均线圈直径(D)以及活动线圈数量(N)这三个关键连续变量进行精确寻优, 最终实现弹簧整体重量(或制造成本)的最小化。其数学描述如下:

$$\begin{aligned} \text{变量: } \bar{x} &= [x_1 \quad x_2 \quad x_3] = [d \quad D \quad N] \\ \text{目标函数: } f(\bar{x}) &= (x_3 + 2)x_2x_1^2 \\ \text{约束条件: } \begin{cases} g_1(\bar{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \\ g_2(\bar{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0 \\ g_3(\bar{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\ g_4(\bar{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \end{cases} \end{aligned}$$

其中, $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15$ 。

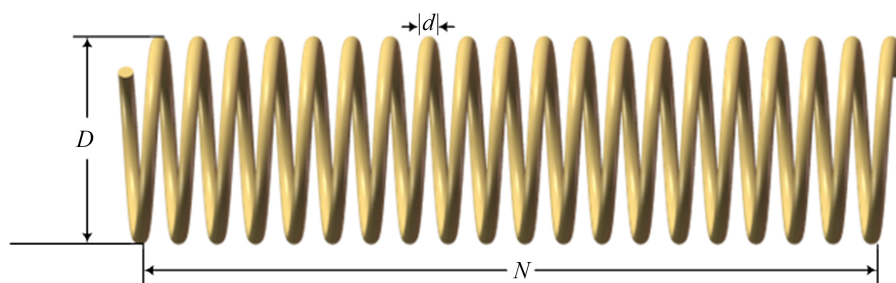


Figure 5. Spring structure diagram
图 5. 弹簧结构图

表 2 记录了 6 种对比算法在张力/压缩弹簧设计问题上的统计寻优结果。由表中数据可以清晰地观察到, ISBO 算法在所有评价维度上均取得了优异的表现。该算法不仅能够有效实现弹簧设计重量的最小化, 更在寻优精度与稳定性之间取得了绝佳的平衡, 进一步验证了该算法在解决现实复杂工程设计问题中的卓越潜力和高度可靠性。

Table 2. Test results comparison of tension/compression spring design issues
表 2. 张力/压缩弹簧设计问题测试结果对比

| | DBO | PSO | GWO | SSA | SBO | ISBO |
|-------|-------------|-------------|-------------|-------------|------------|--------------------|
| 平均值 | 0.01386525 | 0.01317851 | 0.01274341 | 0.01351248 | 0.01318689 | 0.01266856 |
| 标准差 | 0.00178699 | 0.00115504 | 8.67153E-05 | 0.00141091 | 0.00043872 | 4.99917E-06 |
| 最优值 | 0.01266906 | 0.01266601 | 0.01268318 | 0.01266821 | 0.01276287 | 0.01266523 |
| x_1 | 0.05123259 | 0.05149801 | 0.05148815 | 0.05209515 | 0.05122456 | 0.051 |
| x_2 | 0.34583529 | 0.35213726 | 0.35182722 | 0.36656621 | 0.35673293 | 0.31698307 |
| x_3 | 11.95669424 | 11.56271242 | 11.59829309 | 11.73410501 | 14.1054352 | 11.28808113 |

5. 结论

针对原始状态基优化算法(SBO)易陷入局部最优、收敛速度慢等缺陷, 本文提出了一种多策略改进状态基优化算法(ISBO)。本文的主要工作与结论如下:

(1) 机制创新: 算法在初始化阶段引入 Logistic-Tent 混沌映射, 有效提升了初始种群的多样性与空间分布均匀度; 同时, 结合思维创新策略(TIS), 利用历史成功经验与知识深度为停滞种群提供新方向, 增强了算法跳出局部最优的能力。

(2) 基准函数测试表现: 在 CEC2022 标准函数集上的多维测试表明, ISBO 在初始收敛速率、求解精度及规避早熟收敛方面, 均优于原始 SBO 及 GWO、DBO、PSO、SSA 算法, 表明 ISBO 具有较好的综合寻优能力和鲁棒性。

(3) 工程应用价值: 在张力/压缩弹簧设计这一具有复杂非线性约束的实际工程问题中, ISBO 在寻优精度与求解稳定性上均取得了更优的表现, 验证了其在解决现实受限工程优化问题中的有效性和可靠性。

6. 局限性与未来工作

尽管本文提出的多策略改进状态基优化算法(ISBO)在标准测试函数和经典工程问题中表现出优越的

寻优性能和鲁棒性，但仍存在局限性值得进一步探讨和改进：

(1) ISBO 算法在引入 Logistic-Tent 混沌映射与思维创新策略后，虽然显著提升了全局搜索能力和跳出局部最优的能力，但也相应地增加了每轮迭代的计算复杂度。在处理计算资源极度受限或目标函数极其耗时的超大规模实时优化问题时，其时效性仍需进一步优化。

(2) 尽管 TIS 策略在多数测试函数中表现良好，但不同问题类型可能对 TIS 参数的敏感性不同，对于具有极高维度或特殊地形的搜索空间，现有的参数配置可能并非最优，模型在跨领域应用时的参数自适应调节能力仍有待加强。

展望未来，后续研究将重点关注算法的以下两个方面：一方面，针对 ISBO 算法在引入思维创新策略后计算耗时大的问题，探索轻量化的 TIS 变体或并行计算架构，以降低时间复杂度，提升算法在大规模优化问题中的适用性；另一方面，深入研究 TIS 策略中关键参数的自适应调节机制，从而进一步增强算法的适用性和鲁棒性。

基金项目

江西省教育厅科学技术研究项目：GJJ2203620 基于多智能体的智慧路灯微电网群协同优化调度研究。

参考文献

- [1] Reddy, S.S. and Momoh, J.A. (2015) A Comprehensive Review of Meta-Heuristic Optimization Techniques for Optimal Power Flow Problems. *Electric Power Systems Research*, **121**, 1-15.
- [2] Too, J. and Mirjalili, S. (2021) A Hyper-Heuristic Framework for Solving the Optimal Power Flow Problem. *Energies*, **14**, Article 5893.
- [3] Jia, H., Sun, L. and Zhang, R. (2022) A Survey on Metaheuristic Algorithms for Feature Selection: A Decade of Research. *Neurocomputing*, **489**, 347-372.
- [4] Zhang, X., Wang, Y. and Cui, G. (2021) A Survey of Meta-Heuristic Algorithms for Trajectory Optimization in Aerospace Engineering. *Astrodynamics*, **5**, 221-246.
- [5] Dokeroglu, T., Sevinc, E., Kucukyilmaz, T. and Cosar, A. (2019) A Survey on New Generation Metaheuristic Algorithms. *Computers & Industrial Engineering*, **137**, Article 106040. <https://doi.org/10.1016/j.cie.2019.106040>
- [6] Faramarzi, A., Heidarinejad, M., Mirjalili, S. and Gandomi, A.H. (2020) Marine Predators Algorithm: A Nature-Inspired Metaheuristic. *Expert Systems with Applications*, **152**, Article 113377. <https://doi.org/10.1016/j.eswa.2020.113377>
- [7] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. (2019) Harris Hawks Optimization: Algorithm and Applications. *Future Generation Computer Systems*, **97**, 849-872. <https://doi.org/10.1016/j.future.2019.02.028>
- [8] Li, S., Chen, H., Wang, M., Heidari, A.A. and Mirjalili, S. (2020) Slime Mould Algorithm: A New Method for Stochastic Optimization. *Future Generation Computer Systems*, **111**, 300-323. <https://doi.org/10.1016/j.future.2020.03.055>
- [9] Hussain, K., Mohd Salleh, M. N., Cheng, S. and Shi, Y. (2019) On the Exploration and Exploitation in Popular Metaheuristics. *Expert Systems with Applications*, **134**, 1-18.
- [10] Mohammadi, H. and Abdi, H. (2023) Status-Based Optimization: A Novel Meta-Heuristic Algorithm for Solving Optimization Problems. *Applied Soft Computing*, **139**, Article 110214.
- [11] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M. and Gandomi, A.H. (2021) The Arithmetic Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, **376**, Article 113609. <https://doi.org/10.1016/j.cma.2020.113609>
- [12] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [13] Xue, J. and Shen, B. (2022) Dung Beetle Optimizer: A New Meta-Heuristic Algorithm for Global Optimization. *The Journal of Supercomputing*, **79**, 7305-7336. <https://doi.org/10.1007/s11227-022-04959-6>
- [14] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, 27 November 1995-1 December 1995, 1942-1948. <https://doi.org/10.1109/icnn.1995.488968>
- [15] Xue, J. and Shen, B. (2020) A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. *Systems Science & Control Engineering*, **8**, 22-34. <https://doi.org/10.1080/21642583.2019.1708830>