



计算机科学与探索

Journal of Frontiers of Computer Science and Technology

ISSN 1673-9418, CN 11-5602/TP

《计算机科学与探索》网络首发论文

题目：融合排序弹性碰撞的改进麻雀搜索算法
作者：王子恺，黄学雨，朱东林，郭伟
网络首发日期：2022-07-15
引用格式：王子恺，黄学雨，朱东林，郭伟. 融合排序弹性碰撞的改进麻雀搜索算法[J/OL]. 计算机科学与探索.
<https://kns.cnki.net/kcms/detail/11.5602.TP.20220714.1731.002.html>



网络首发：在编辑部工作流程中，稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定，且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式（包括网络呈现版式）排版后的稿件，可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定；学术研究成果具有创新性、科学性和先进性，符合编辑部对刊文的录用要求，不存在学术不端行为及其他侵权行为；稿件内容应基本符合国家有关书刊编辑、出版的技术标准，正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性，录用定稿一经发布，不得修改论文题目、作者、机构名称和学术内容，只可基于编辑规范进行少量文字的修改。

出版确认：纸质期刊编辑部通过与《中国学术期刊（光盘版）》电子杂志社有限公司签约，在《中国学术期刊（网络版）》出版传播平台上创办与纸质期刊内容一致的网络版，以单篇或整期出版形式，在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊（网络版）》是国家新闻出版广电总局批准的网络连续型出版物（ISSN 2096-4188，CN 11-6037/Z），所以签约期刊的网络版上网络首发论文视为正式出版。

融合排序弹性碰撞的改进麻雀搜索算法

王子恺¹, 黄学雨²⁺³, 朱东林¹, 郭伟¹

1. 江西理工大学 信息工程学院, 江西 赣州 341000
 2. 江西理工大学 软件工程学院, 南昌 330013
 3. 南昌市虚拟数字工程与文化传播重点实验室, 南昌 330013
- + 通信作者 E-mail: 1137753107@qq.com

摘要: 为了改善麻雀搜索算法 (Sparrow Search Algorithm, SSA) 种群初始化结果不充分导致多样性丧失, 勘探和开采过程中易受到个别位置信息干扰影响寻优精度等缺点, 提出了融合排序弹性碰撞的新型麻雀搜索算法, 简称为 XSSA。首先, 采用改进的无限折叠迭代混沌映射 (Iterative Chaotic Map with Infinite Collapses, ICMIC) 初始化种群, 提升了初始种群分布的分散程度; 其次, 使用高斯随机游走策略平衡算法的勘探和开发能力; 此外, 在发现者更新后对所有个体执行排序弹性碰撞策略, 避免算法过早地收敛到局部极值; 最后, 根据不同阶段的寻优特点制定多策略边界处理机制, 保留住种群数量, 避免多样性的丧失。同时, 结合重要的位置信息对超出边界的个体进行位置再更新, 使得处理后的位置更加合理, 为接下来的迭代搜索提供质量保证。通过对 12 个基准函数进行仿真实验, 并画出收敛精度图直观展示算法性能。借助各策略的贡献测试、Wilcoxon 秩和检验、Friedman 检验的综合排名等证明了 XSSA 的有效性、独特性和具有较好的寻优性能。

关键词: 麻雀搜索算法; 无限折叠迭代混沌映射 (ICMIC); 高斯随机游走; 排序弹性碰撞; 多策略边界处理

文献标志码: A **中图分类号:** TP301.6

Improved Sparrow Search Algorithm Combining Ranking-Based Elastic Collision

WANG Zikai¹, HUANG Xueyu²⁺³, ZHU Donglin¹, GUO Wei¹

1. School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China
2. School of Software Engineering, Jiangxi University of Science and Technology, Nanchang 330013, China
3. Nanchang Key laboratory of Virtual Digital Factory and Cultural Communications, Nanchang 330013, China

Abstract: In order to improve the Sparrow Search Algorithm (SSA) population initialization results are insufficient, resulting in loss of diversity, and is easy to be interfered by individual location information during the exploration and mining process, which affects the optimization accuracy. An improved sparrow search algorithm based on fusion ranking elastic collision, referred to as XSSA, is proposed to solve the problem easily disturbed by individual location information. Firstly, an improved Iterative Chaotic Map with Infinite Collapses (ICMIC) improves the dispersion degree of the initial population distribution. Then, the Gaussian random walk strategy is used to balance the

基金项目: 国家重点研发计划重点专项 (2020YFB1713700)。

This work was supported by the National key research and development program of China (Grant Nos. 2020YFB1713700).

exploration and development capabilities of the algorithm. In addition, all individuals are sorted elastically collided after the discoverer is updated, which avoids premature convergence of the algorithm to the local extreme value. Finally, a multi-strategy boundary processing mechanism is formulated according to the characteristics of optimization at different stages to retain the population and avoid the loss of diversity. At the same time, the position of individuals beyond the boundary is re-updated in combination with important position information, so that the processed position is more reasonable and provides quality guarantee for the next iterative search. Through the simulation experiment of 12 benchmark functions, and the convergence accuracy graph demonstrates the performance of the algorithm. By means of contribution tests for each strategy, Wilcoxon rank sum test, and the comprehensive ranking of Friedman test, the effectiveness, uniqueness and superior performance of XSSA are proved.

Key words: Sparrow search algorithm; ICMIC map; Gauss walk learning; Ranking-based elastic collision; Multi-strategy boundary processing mechanism

随着工业制造、航天、医疗等多个领域发展迅猛,伴随着出现了很多复杂的优化问题。传统的优化方案已经无法在时间和精度上同时满足技术发展所需求的高效率。为了解决这些难题,研究者们根据自然界中生物习性,探索出了一系列群智能优化算法。较早的群智能算法包括粒子群算法(Particle Swarm Optimization Algorithm, PSO)^[1-2]、蚁群算法(Ant Colony System, ACS)^[3]、较新提出的如狮群算法(Lion Swarm Optimization, LSO)^[4]、蝴蝶算法(Butterfly optimization algorithm, BOA)^[5]等。相比于传统的优化方法,群智能算法具有参数少、逻辑较简单、实验灵活等特点从而受到广泛的关注和研究。这些优点也为群智能优化算法在自主水下航行器(autonomous underwater vehicle, AUV)路径规划^[6]、文本聚类^[7]、求解分布式异构并行机调度^[8]、图像分割^[9]等实际问题中的应用提供了帮助。

麻雀搜索算法由Xue J K等首次提出^[10]。其主要特点是自组织能力强、运行时间短、鲁棒性强^[11],但容易受随机性和局部最优值影响使得表现不稳定,同时伴随着因收敛速度过快导致多样性丧失严重并且带来精度降低等问题。

针对上述不足,许多学者做出了改进并结合了实际应用检验其性能。Zhu D L等^[12]将透镜反向学习和正弦余弦机制结合并提出了一种学习型麻雀搜索算法(learning sparrow search algorithm, LSSA)并在机器人路径规划问题上验证了可行性;Liang Q K等^[13]为了解决天线阵优化问题,将齐次混沌策略、

自适应算子和边界约束结合到新提出的改进麻雀搜索算法(modified sparrow search algorithm, MSSA)上并实现了最大旁瓣电平的最大值缩减。Liu G Y等^[14]利用融合混沌算子和自适应策略的麻雀搜索算法(CASSA)的摆脱最优解的能力,引入集成变异算子等办法使得3D无人机的高度更合适、长度和平滑度更好。Wang Z K等^[15]提出多策略学习型麻雀搜索算法(IHSSA),引入混和反向学习和纵横交错方法提升算法的寻优精度,并在工程优化问题上成功寻优。王振东等^[16]结合非线性收敛因子和柯西变异算子,同时提出新颖的边界处理机制,提高了无线传感网络中的覆盖率,证明了所提出的增强型麻雀搜索算法(enhanced sparrow search algorithm, ESSA)的有效性。尽管上述文献提出了较好的具有建设性的尝试,但是,仍然存在部分结合的策略优势有限。且随着在文献中使用频率的增大,更多更优的组合策略和基于理论原理的新策略有待发掘。同时,如何对有利解的信息进行充分使用是值得考虑的问题。基于对上述问题的改进和探索,本文进行了新的尝试,主要贡献如下:

(1) 使用改进的ICMIC初始化种群,减少了个体的重合和堆叠,增加初始麻雀个体的分布合理性和分散程度。

(2) 根据麻雀活动的习性引入高斯随机游走策略平衡算法的勘探和开发能力。

(3) 将排序配对思想和弹性碰撞原理结合,创造出全新的排序弹性碰撞策略帮助算法跳出局部

最优，从而提高了寻优精度。

(4) 针对各阶段超出边界的数量不同的情况，设计不同的边界处理策略。在使得处理机制更多元化的同时，利用了重要的位置信息，增加了合理性。

1 麻雀搜索算法

在觅食过程中，麻雀种群通常分工明确。10%-20%的个体作为发现者领导整个种群的觅食方向，其他的麻雀则作为跟随者朝着目标地点前进。考虑到对天敌的预警能力，有一部分麻雀会担任警戒者，时刻预防危险的发生，并在遇险时提醒整个种群做出反哺食行为。整个种群机动前进，不断更新各自的位置。

发现者会根据与安全阈值的比较进行不同的位置更新：

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot T}\right) & \text{if } R_2 < ST \\ x_{i,j}^t + Q \cdot L & \text{if } R_2 \geq ST \end{cases} \quad (1)$$

式中， X 表示麻雀位置，上标表示迭代次数。下标则表示麻雀编号和数据表示维度（如 $x_{i,j}^t$ 表示第 t 代，第 i 只麻雀在第 j 维的位置）， T 表示最大迭代次数， t 表示当前迭代次数； α 是符合 $(0,1]$ 的均匀数； Q 是服从标准正态分布的随机数； L 是全 1 的矩阵；报警值 $R_2 \in [0,1]$ ，安全值 $ST \in [0.5,1]$ 。

$R_2 < ST$ 时，环境安全可以广泛搜索；当危险来临时， $R_2 \geq ST$ ，发现者带领队伍迅速逃离当前位置，躲避天敌。

跟随者会根据自己适应度的排序进行差别较大的更新：

$$x_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{x_w^t - x_{i,j}^t}{i^2}\right) & \text{if } i > N/2 \\ x_p^{t+1} + |x_{i,j}^t - x_p^{t+1}| \cdot A^+ \cdot L & \text{otherwise} \end{cases} \quad (2)$$

式中， x_w 、 $x_{i,j}^t$ 、 $x_{b,j}^{t+1}$ 分别表示当前全局最差、第 i 只麻雀、第 j 维的全局最优位置； A 是一个 $1 \times d$ 的矩

阵，随机赋值 -1 或 +1； A^+ 满足 $A^+ = A^T (AA^T)^{-1}$ 。当 $i > N/2$ 时，适应度较差的个体必须去往较远的地点去发现食物；而对于另一部分适应度较好的麻雀，他们时刻监视着发现者的位置并找机会取代他们的角色。

警戒者的位置更新如下：

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + Q |x_{i,j}^t - x_{b,j}^t| & , f_i \neq f_g \\ x_{i,j}^t + K \frac{x_{i,j}^t - x_{w,j}^t}{(f_i - f_w) + \varepsilon} & , f_i = f_g \end{cases} \quad (3)$$

式中， K 是 $[-1,1]$ 内的随机数，和 Q 一样用于步长的控制； ε 是一个接近无穷小的值，为了防止分母而存在； f_i 、 f_g 和 f_w 分别表示当前麻雀的适应度、全局最优和全局最差的适应度值。 $f_i \neq f_g$ 表示边缘位置的个体在危险时向群内靠近， $f_i = f_g$ 则表示位于群中心的个体在其它个体之间随机游走。

2 融合排序弹性碰撞的改进麻雀搜索算法

2.1 改进的 ICMIC 混沌初始化种群

群智能算法中的位置极为重要初始状态下的位置可能将整个优化过程引导至完全不同的最终效果。利用混沌序列进行种群初始化会常常能取得比伪随机数更好的效果。经典的混沌映射 ICMIC 生成的序列可以看作是均匀分布的更好的伪随机模型^[17]。同时，当前的 SSA 文献中并没有尝试用该混沌来初始化麻雀种群。因此，本文选取改进的 ICMIC 混沌来初始化种群位置，其基本思想是在生成 $[0,1]$ 之间的混沌序列并且载波至整个搜索空间内^[18]。基础 ICMIC 的数学表达为：

$$x_{n+1} = \sin\left(\frac{\alpha\pi}{x_n}\right), \alpha \in (0,1) \quad (4)$$

本文使用的改进 ICMIC 公式如下：

$$x_{n+1} = \sin\left(\frac{\alpha\beta}{x_n}\right) \quad (5)$$

$$\beta = 3 \cos(rand) + 0.5$$

改进后的 ICMIC 对常数 π 进行调整, 增加了一个小范围的浮动, 进一步增加了随机性, 符合原始 SSA 中对于初始化随机性的要求。图 1.(a) 和图 1.(b) 分别展示了随机初始化和改进的 ICMIC 初始化后

的种群分布情况。如图圈出了个体重合的情况, 可以看出混沌操作后的种群分布更佳, 重合和边界上的个体数量更少。在该阶段的合理分布是整个寻优过程质量的重要保证。

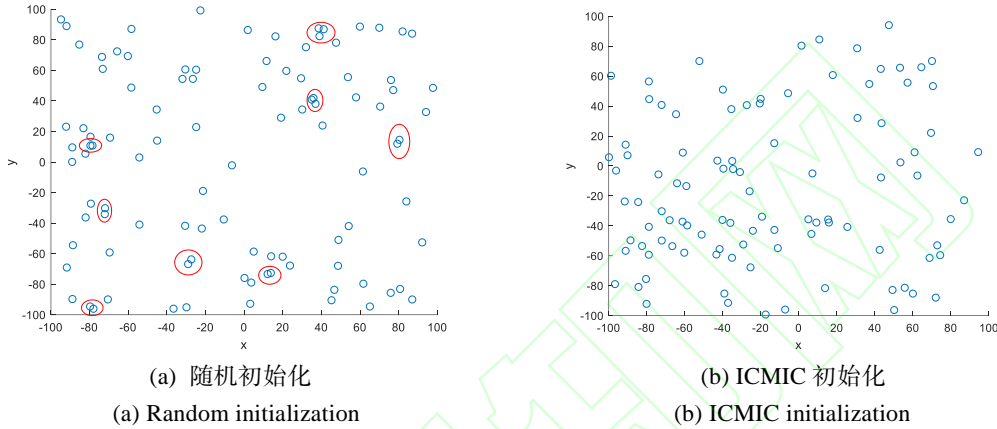


Fig.1 Distribution of initial population

图 1 初始种群分布图

2.2 高斯随机游走

作为经典的随机游走策略之一, 高斯随机游走策略 (Gauss walk learning, GWL) 具有较强的开发能力^[19], 并在差分进化算法中展现了不错的效果。麻雀个体在发现者的引导下, 会逐渐向着更好的觅食点前进, 跟随者会围绕着发现者作无规则运动。受到发现者在安全范围内自由勘探和跟随者在发现者周围无序飞行的行为启发, 采用高斯随机游走策略模拟作为发现者的麻雀的更新, 以平衡该阶段的勘探与开发能力。GWL 的更新公式如下:

$$\begin{aligned} x_b'' &= \text{Gaussian}(x_b', \tau) + (r_1 \cdot x_b' - r_2 \cdot x_i') \\ \tau &= \exp(\alpha) \cdot c \cdot (x_i' - x_w') \\ \alpha &= \sin\left(\pi \cdot \frac{T}{t}\right) + \cos\left(\frac{\pi}{2} \cdot \frac{T}{t}\right) - 1 \end{aligned} \quad (6)$$

式中, x_b' 表示最佳位置, x_b'' 表示最佳位置更新后的位置, $\text{Gaussian}(x_b', \tau)$ 表示以 x_b' 为期望 τ 为标准差的高斯分布, x_w' 表示最差位置, x_i' 表示随机个体的位置, r_1 和 r_2 是两个 $[0,1]$ 之间的随机数, c 是一个 $[0,1]$ 之间的常数。 $(r_1 x_b' - r_2 x_i')$ 调整方向,

$\exp(\alpha) \cdot c$ 控制步长。

2.3 排序弹性碰撞策略

学习机制是模拟人们日常生活中的知识传递方法提出的一种信息传递的方法, 将预设的一些位置或者速度信息赋予其他的个体或者向量, 从而对该个体或向量进行变换。其中, 最知名的学习机制是反向学习, 而随着研究的深入, 一些结合物理学知识的新型学习机制也逐渐进入人们的视野, 比如透镜学习^[12]、精英反向学习、一般反向学习^[15]等。本文将介绍两种优秀的学习策略并组合使用最大程度上体现算法的性能优势。

2.3.1 排序配对策略

文献[20]提出了一种排序配对的学习方法和适应度差异最大化的思想。这种思想强调表现差的个体向最佳的个体学习不能实现辅导效率的最大化, 最好的学习技巧应当是将所有个体分成两组, 并按照能力的高低分别对两组成员进行编号。之后, 将差解向与之相同编号的优解进行配对学习以实现效益最大化。这种学习思想称为排序配对 (ranking paired, RP)。RP 的提出是为了在保证优秀个体位置不变的前提下, 对较差的部分进行更新, 使得整

个种群寻优能力最大化提升。

2.3.2 弹性碰撞策略

弹性碰撞原理属于物理学中的力学范畴，文献[21]提出将该策略用于对最优个体的扰动更新。其原理为，当一个物体 A 以一定的初速度去撞击静止的物体 B，两者的速度及运动方向会发生变化。在不考虑外界摩擦阻力的前提下，根据动量守恒原则，两个物体在碰撞之后会出现如下三种情况（设 A 初始运动方向为正方向）：

- （1）若 A 的质量小于 B 的质量，A 向负方向运动，B 向正方向运动。
- （2）若 A 的质量等于 B 的质量，A 会处于静止状态，B 向正方向运动，且速度等同于 A 的初始速度。
- （3）若 A 的质量大于 B 的质量，A 和 B 都向正方向运动。

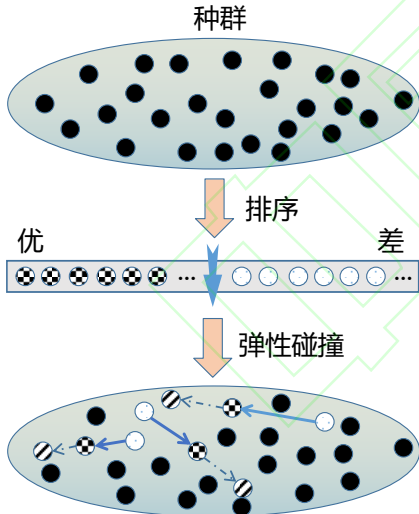


Fig.2 Ranking-based elastic collision
图 2 排序弹性碰撞示意图

2.3.3 基于排序配对的弹性碰撞策略

对于较差的个体，我们按照如下公式赋予其初速度 v_0 ，速度变化曲线如图 3：

$$v_0 = T \cdot e^{-(x_w - x_i)^2}$$

$$T = \frac{1}{2} \left(\left(\frac{2t}{T} - 1 \right)^3 + 1 \right) \quad (7)$$

假设时间忽略不计，即不考虑到碰撞时速度的改变，根据弹性碰撞的原理和动量守恒定律可知：

$$\begin{cases} m_1 v_0 = m_1 v_1 + m_2 v_2 \\ \frac{1}{2} m_1 v_0^2 = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \end{cases} \quad (8)$$

式中， m_1 和 m_2 分别表示差的个体和优的个体的质量， v_1 和 v_2 分别表示差的个体和优的个体碰撞后的速度， v_1 和 v_2 满足：

$$\begin{cases} v_1 = \frac{m_1 - m_2}{m_1 + m_2} \cdot v_0 \\ v_2 = \frac{2m_1}{m_1 + m_2} \cdot v_0 \end{cases} \quad (9)$$

并按下式将弹性碰撞后差解的位置进行更新：

$$x' = x + v_1 \quad (10)$$

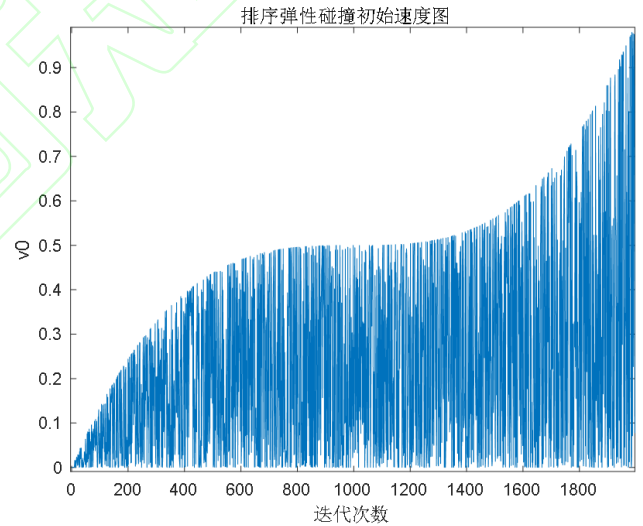


Fig.3 Graph of initial velocity
图 3 初始速度变化图

2.4 多策略边界处理

合理的边界处理策略是优秀的算法必不可少的组成部分，原始的 SSA 只是简单地讲超出边界的个体更新在边界上。这种统一的处理方式会给边界及其附近区域造成负担，造成多样性的缺失。鉴于 SSA 中各阶段的差异性，本文提出了一种多策略边界处理方式解决该问题。

在发现者阶段，其目的是尽可能广得发散种群，在大面积内部找寻潜在的最优位置。因此，在全局范围内的随机性较高的处理方式更合适。

$$x'_i = ub - (ub - lb) \cdot rand, x > ub \text{ or } x < lb \quad (11)$$

同理,警戒者更新完成后的位置信息留作下一代的初始值,种群越分散,勘探的能力就越强。结合着两个阶段的特点,制定如下边界处理方式:

$$x'_i = lb + (ub - lb) \cdot rand, x > ub \text{ or } x < lb \quad (12)$$

式中, x'_i 分别表示按照式(11)、(12)处理后生成的新位置。

在跟随者更新阶段,由于跟随者数量最多,更新后的超出边界的个体数量也最多。考虑到跟随者的作用是在跟随发现者进行寻优,并时刻保持着对发现者角色的竞争。因此,该阶段的界外个体重置机制应当充分利用当前全局最优位置的信息,同时,该阶段更新前的初始位置和超出边界的位置信息反映了该更新的方向和更新长度,具有较高的参考意义。考虑到超出上边界和超出下边界的情况略有差异,更新公式也应有所区别。依据上述原则,提出如下公式:

超出上边界的边界处理更新方法为:

$$x''_i = x_b + |x_b - ub| \cdot \frac{|x_i - ub|}{|x'_i - x_i|} \quad (13)$$

超出下边界的边界处理更新方法为:

$$x''_i = x_b - |x_b - lb| \cdot \frac{|x_i - lb|}{|x'_i - x_i|} \quad (14)$$

式中,最优位置和边界的差值的绝对值,即 $|x_b - ub|$ 和 $|x_b - lb|$ 作为步长因子,后面的分式提供了步数, x'' 表示边界处理后的新位置, x'_i 表示按照式(2)更新后的超出边界的位置, x_i 则表示按照式(2)更新前的初始位置。注意,这里的 x'_i 区别于式(11)、(12)中的 x'_i 。其余各参数含义均遵照前文中所注。

2.5 XSSA 流程

XSSA 算法利用改进的 ICMIC 初始化种群,在

发现者更新阶段引入高斯随机游走策略用于平衡全局和局部搜索的能力。此外,排序弹性碰撞策略的结合具备防止算法陷入局部最优的能力。同时,根据不同阶段的更新意义制定多策略边界处理机制,提高了处理机制的合理性,也进一步保证了种群的多样性。XSSA 流程可以总结为算法1中所示的伪代码:

算法1 XSSA

输入:

M : 种群规模

PD : 发现者比例

SD : 警戒者比例

$R2$: 报警值

ST : 安全值

T : 最大迭代次数

ε : 精度阈值

输出:

x_b : 最优位置

f_b : 最优位置的适应度值

1: 使用公式(5)初始化种群.

2: **while** ($t < T$)

3: 根据适应度值对整个种群进行排序并找出最优和最差个体.

4: **for** $i = 1: PD$

5: 根据公式(1)更新位置并根据公式(11)进行边界处理.

6: **end for**

7: 根据公式(6)更新当前最优位置.

8: 根据适应度值进行排序并根据公式(7-10)更新种群位置.

9: **for** $i = (PD+1): M$

10: 根据公式(2)更新位置并根据公式(13-14)进行边界处理.

11: **end for**

12: **for** $i = 1: SD$

13: 根据公式(3)更新位置并根据公式(12)进行边界处理.

14: **end for**

15: 得到新种群.

16: 如果新的位置优于当前位置,则更新它.

17: $t = t + 1$.

18: **end while**

19: 返回目标函数值.

2.6 复杂度分析

时间复杂度是一个判断算法性能和计算运行代价的重要指标之一。原始的 SSA 时间复杂度为 $O(T \times D \times M)$ 。其中， T 、 D 、 M 分别表示最大迭代次数、数据维度、种群规模。XSSA 时间复杂度分析如下：

(1) 引入改进的 ICMIC 初始化种群，时间复杂度为 $O_1 = O(D \times M)$ 。

(2) 引入 GWL 策略，假设发现者占比是 PD，则 GWL 额外增加的时间复杂度为 $O_2 = O(T \times D \times M \times PD)$

(3) 排序弹性碰撞策略主要在排序和学碰撞的过程，时间复杂度仍为 $O_3 = O(T \times D \times M)$

(4) 边界处理机制并未增加时间复杂度。

综上所述，XSSA 的时间复杂度也是 $O = O(T \times D \times M) + O_1 + O_2 + O_3 = O(T \times D \times M)$ ，和 SSA 一致，并没有增加复杂度。

3 算法性能测试

为了验证 XSSA 的性能，本文选取 12 个基准测试函数^[10、12]用来测试其收敛速度和寻优精度，

并和其它 7 种算法进行比较。这些算法包括原始的 SSA 及两个变体：吕鑫提出的混沌麻雀搜索算法（chaotic sparrow search algorithm, CSSA）^[22]和 Wang Z K 等提出的 IHSSA^[15]，3 种新颖的群智能算法：2014 年提出的灰狼优化算法（grey wolf optimizer, GWO）^[23]、2019 年提出的 BOA^[5]和 2021 年提出的改进灰狼算法（improved grey wolf optimizer, IGWO）^[24]。还有 Peng H 等改进的多策略串行布谷鸟搜索算法（multi-strategy serial CS, MSSCS）^[19]。多角度的选择比较算法旨在更全面的验证 XSSA 的性能，证明其优势。参数设置都遵照算法提出的原文中所示，整理在表 1 中。12 个基准函数的具体信息如表 2 所示。为了实验的公平性和有效性，本文中所有算法的种群数量都定为 500，最大迭代次数 500，并将各实验独立运行 30 次，选取平均值、标准差、最优值作为三个评价指标。实验数据整理在表 3 中，最优数据进行加粗处理。本文的实验环境均为 Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz. 并采用 Matlab2019a 进行仿真。

表 1 参数设置

Table 1 Parameter settings

算法	参数设置
SSA	PD=0.2, ST=0.6, SD=0.2
CSSA	PD=0.2, ST=0.8, SD=0.2
IHSSA	PD=0.2, ST=0.8, SD=0.2
XSSA	PD=0.2, ST=0.8, SD=0.2
MSSCS	$\alpha = 0.01, \beta = 1.5, Pa = 0.25, c = 0.2, P_{Amax} = 0.35, P_{Amin} = 0.25$
BOA	$a = 0.1, b = 0.025, c = 0.1, p = 0.8$
LSO	$a_{max} = 2, a_{min} = 0$
GWO	$a_{max} = 2, a_{min} = 0$

表 2 测试函数

Table 2 Test function

函数	维度	搜索区域	理论最优值
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
$F_3(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[-100,100]	0

表 3 测试函数结果对比

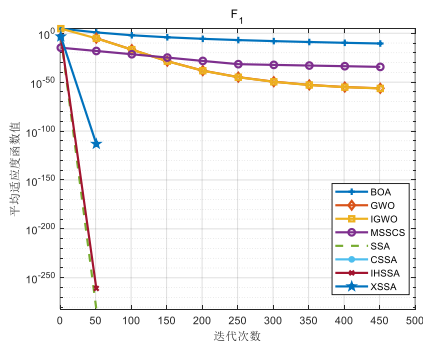
[illegible]

	平均值	1.55E-03	2.83E-08	1.23E-24	-	1.50E-23	5.38E-24	1.10E-22	5.05E-24
F8	标准差	1.34E-03	2.54E-08	1.07E-24	-	3.03E-23	1.67E-23	2.94E-22	7.51E-24
	最优值	4.94E-05	1.26E-09	1.04E-25	-	4.07E-27	1.55E-29	1.75E-25	9.54E-30
	平均值	2.28E+00	1.89E-05	1.38E-05	3.51E-13	3.13E-09	8.47E-10	5.98E-10	3.44E-10
F9	标准差	1.74E+00	1.87E-05	9.17E-06	2.40E-13	7.40E-09	2.16E-09	1.84E-09	7.49E-10
	最优值	1.70E-01	2.74E-06	2.97E-06	6.83E-14	1.35E-14	2.04E-15	1.32E-20	1.09E-20
	平均值	1.27E-12	0.00E+00	0.00E+00	-	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	标准差	2.11E-13	0.00E+00	0.00E+00	-	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	最优值	8.96E-13	0.00E+00	0.00E+00	-	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	平均值	1.05E+00	9.98E-01	9.98E-01	-	1.33E+00	9.98E-01	9.98E-01	9.98E-01
F11	标准差	2.77E-01	7.17E-12	3.33E-16	-	7.39E-01	3.49E-16	1.78E-13	4.21E-16
	最优值	9.98E-01	9.98E-01	9.98E-01	-	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	平均值	3.22E+31	5.55E+14	6.13E+13	1.00E+10	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	标准差	6.96E+31	7.48E+14	3.39E+13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	最优值	2.01E+22	9.31E+12	1.21E+13	1.00E+10	0.00E+00	0.00E+00	0.00E+00	0.00E+00

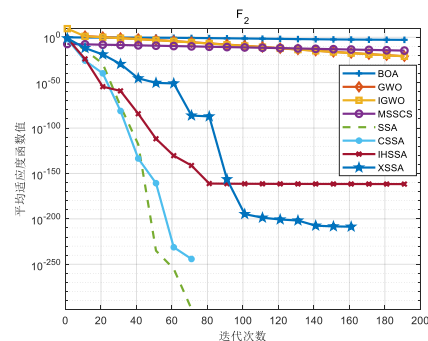
对于单峰函数 F1 和 F2，四种 SSA 优势明显，都能够找到最优值的同时，效果稳定。对于另外两个单峰函数 F3 和 F4，XSSA 排名第二，和最好的 IHSSA 的差距在一个量级以内。而对于多峰函数 F5-F7，XSSA 和其它 SSA 变体一样都能找到最优解，且寻优速度很快。经典算法中，仅 LSO 表现出较强的寻优能力。可见 SSA 整体在多峰函数上的优势。对于复杂的定维函数，由于维度的特殊性，MSSCS 无法在 F8、F10 和 F11 上进行寻优，表中用“-”表示。XSSA 表现出了很强的性能，在 F10、F12 这两个函数中都排名第一。对于 F8，XSSA 在均值和标准差上略差于 IGWO，但最优值能够领先 5 个量级。对于 F9，MSSCS 在均值和标准差上占优，但是其挖掘最优解的能力太差，XSSA 综合性

能更好。对于 F11，SSA 变体在 30 次实验中都找到了最优解，原始的 SSA 则无法找到最优。综上所述，XSSA 在 9 个函数中的性能高居第一，两个函数上表现第二，F11 仅标准差一个指标略逊于 CSSA。因此，可以判断 XSSA 整体性能最好。

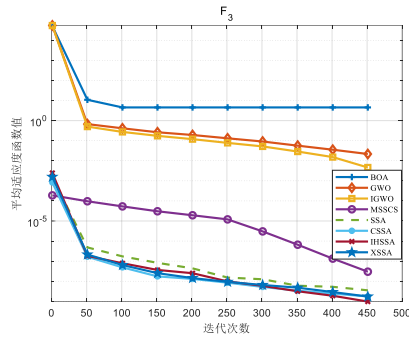
此外，从收敛精度图（图 4）中可见，不论是收敛速度还是寻优精度，XSSA 都表现出色。F1、F5、F7、F10、F12 这 5 个函数中不仅找到了最优解，速度更是最快 F3、F6 和 F9 这 3 个函数中，在各 SSA 变体收敛速度相当的情况下，XSSA 能够寻觅到更优的解，在 F4 中，仅次于 IHSSA。而在 F2 中，虽然三种 SSA 都能找到理论最优解，但是，XSSA 需要花费更多的迭代次数（80-90 代）。



(a) F1 收敛曲线
(a) F1 convergence curve

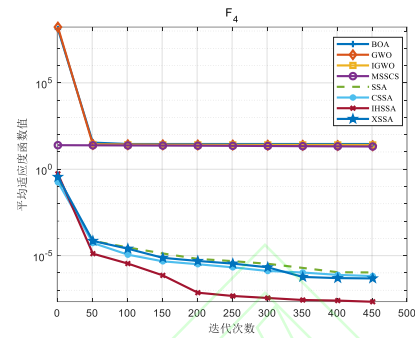


(b) F2 收敛曲线
(b) F2 convergence curve



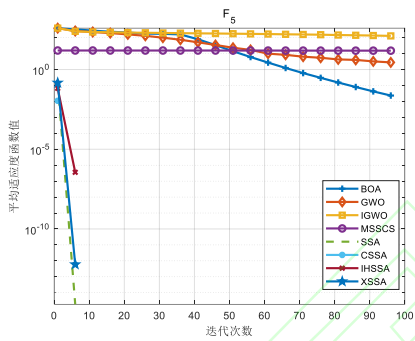
(c) F3 收敛曲线

(c) F3 convergence curve



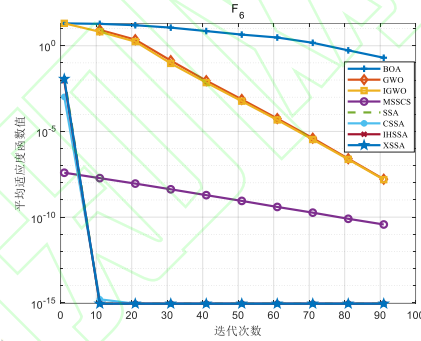
(d) F4 收敛曲线

(d) F4 convergence curve



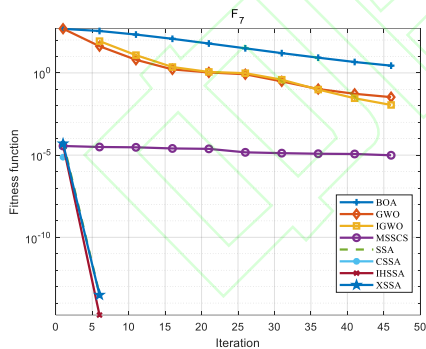
(e) F5 收敛曲线

(e) F5 convergence curve



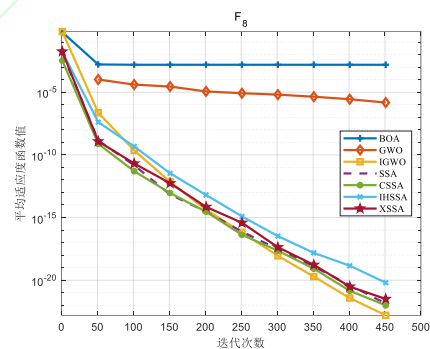
(f) F6 收敛曲线

(f) F6 convergence curve



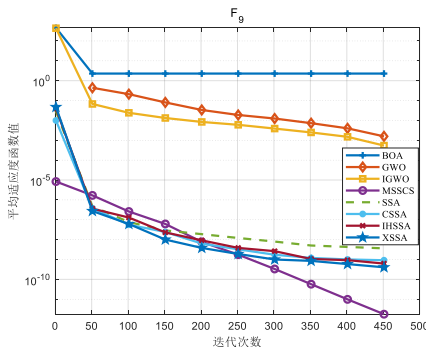
(g) F7 收敛曲线

(g) F7 convergence curve



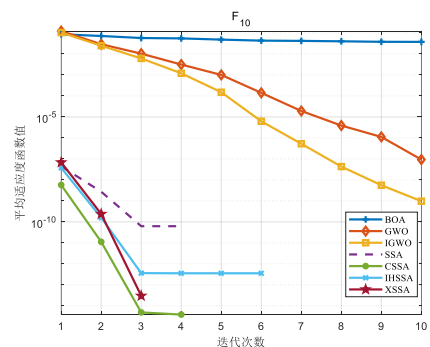
(h) F8 收敛曲线

(h) F8 convergence curve



(i) F9 收敛曲线

(i) F9 convergence curve



(j) F10 收敛曲线

(j) F10 convergence curve

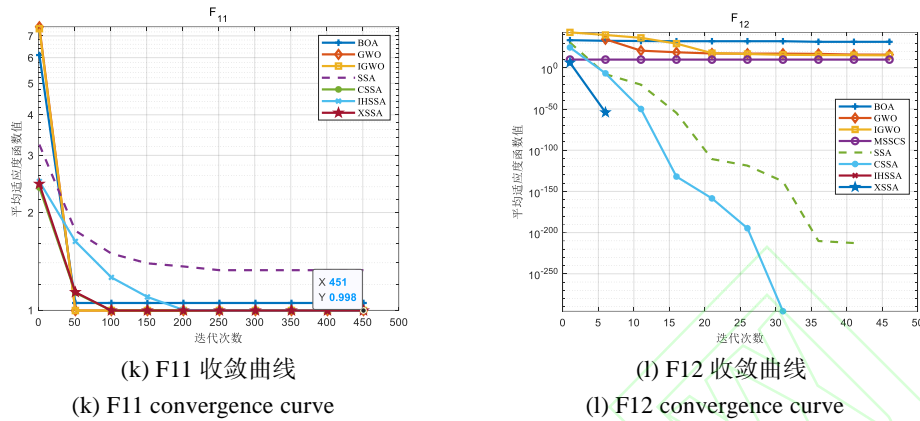


Fig.4 Convergence curve of test functions.

图 4 测试函数收敛曲线

(注：对于 F2、F5、F6、F7、F10、F12，几个 SSA 变体算法早早地收敛，再展示 500 代的收敛图不直观，因此，只展示到收敛至最优值附近的代数。)

此外，为了验证改进 XSSA 各策略在寻优过程中发挥的作用，本文将原始的麻雀搜索算法 (SSA)、改进的 ICMIC 初始化种群的 SSA (ISSA1)、加入高斯随机游走策略的 SSA (ISSA2)、加入排序弹性碰撞原则的 SSA (ISSA3)、结合 ISSA1 和 ISSA2 的 SSA (ISSA4)、结合 ISSA1 和 ISSA3 的 SSA (ISSA5)、结合排序弹性碰撞和高斯随机游走的 SSA (ISSA6) 还有结合所有策略的 XSSA 进行比较，评价指标统计在表 4 中，最优的指标加粗表示。

从表 4 中可以清晰地看出，XSSA 在 12 个测试函数中都表现最佳，其中 9 个更是在三个指标全部领先，在 F3 和 F9 这两个函数中寻到的最优值略差于第一名，F8 中的均值和方差略差，但 XSSA 能找到更优解。通过 ISSA2 和 SSA 的比较可以看出，高斯随机游走策略的引入使得算法在 F4、F8、F9、F11 函数上的寻优精度都有所提升，F9 的最优值更

是有 7 个量级的提升，可见该策略的结合能够提高开发的能力，同时，均值和标准差的提升说明其在一定程度上避免了算法陷入局部最优，增加了勘探的能力。综上所述，高斯随机游走策略能够起到平衡勘探与开发能力的作用。通过 ISSA3 和 SSA 的比较则可以看出，两者在 6 个测试函数上都始终能找到理论最优。在其余的函数中，ISSA3 在 F3 和 F9 中全面占优。对于 F4 和 F8，SSA 能找到更好的最优值，但均值和标准差都不如结合排序弹性碰撞的 ISSA3，且有一个量级的差异。由此可见，排序弹性碰撞策略的结合保留了 SSA 开发能力的同时，又提高了勘探的性能，也增加了鲁棒性。此外，8 种算法在 6 个函数中都能寻到理论最优解 0。而在其他 6 个函数中，相比于原始的 SSA，各改进 SSA 都有所提升，特别是 F3、F8、F11 提升明显。

表 4 各策略贡献测试

Table 4 Contribution test for each strategy

函数	统计结果	SSA	ISSA1	ISSA2	ISSA3	ISSA4	ISSA5	ISSA6	XSSA
F1	平均值	0	0	0	0	0	0	0	0
	标准差	0	0	0	0	0	0	0	0
	最优值	0	0	0	0	0	0	0	0
F2	平均值	0	0	0	0	0	0	0	0
	标准差	0	0	0	0	0	0	0	0
	最优值	0	0	0	0	0	0	0	0
F3	平均值	2.0E-09	1.3E-09	3.3E-09	1.9E-09	1.3E-09	1.5E-09	1.6E-09	1.2E-09
	标准差	3.8E-09	2.3E-09	7.8E-09	3.2E-09	1.0E-09	2.7E-09	2.4E-09	2.1E-09
	最优值	2.3E-12	1.8E-12	3.4E-12	3.5E-13	1.3E-11	2.0E-13	9.8E-12	2.0E-13
F4	平均值	1.0E-06	5.6E-07	1.0E-06	1.1E-06	1.0E-06	1.1E-06	6.7E-07	4.2E-07
	标准差	2.6E-06	1.2E-06	1.9E-06	1.5E-06	1.8E-06	3.3E-06	1.3E-06	7.1E-07
	最优值	1.4E-10	4.2E-11	6.5E-11	2.7E-10	2.7E-10	4.5E-10	3.4E-10	5E-14
F5	平均值	0	0	0	0	0	0	0	0
	标准差	0	0	0	0	0	0	0	0
	最优值	0	0	0	0	0	0	0	0
F6	平均值	8.9E-16	8.9E-16	8.9E-16	8.9E-16	8.9E-16	8.9E-16	8.9E-16	8.9E-16
	标准差	9.9E-32	9.9E-32	9.9E-32	9.9E-32	9.9E-32	9.9E-32	9.9E-32	9.9E-32
	最优值	8.9E-16	8.9E-16	8.9E-16	8.9E-16	8.9E-16	8.9E-16	8.9E-16	8.9E-16
F7	平均值	0	0	0	0	0	0	0	0
	标准差	0	0	0	0	0	0	0	0
	最优值	0	0	0	0	0	0	0	0
F8	平均值	1.5E-23	2.8E-24	4.3E-24	2.0E-24	1.4E-23	1.0E-24	7.1E-24	5.1E-24
	标准差	3.0E-23	5.7E-24	7.8E-24	3.2E-24	4.3E-23	3.0E-24	1.8E-23	7.5E-24
	最优值	4.1E-27	8.0E-28	1.8E-27	7.4E-27	1.0E-28	1.5E-28	2.8E-28	9.5E-30
F9	平均值	3.1E-09	4.7E-09	9.3E-09	1.3E-09	6.8E-10	5.6E-10	4.4E-10	3.4E-10
	标准差	7.4E-09	2.1E-08	2.8E-08	2.5E-09	1.2E-09	1.1E-09	9.3E-10	7.5E-10
	最优值	1.4E-14	1.1E-19	1.7E-21	4.5E-15	7.9E-15	3.0E-23	2.8E-17	1.1E-20
F10	平均值	0	0	0	0	0	0	0	0
	标准差	0	0	0	0	0	0	0	0
	最优值	0	0	0	0	0	0	0	0
F11	平均值	1.328687	1.13028	1.06632	1.196414	1.19641	1.06414	1.07298	0.998
	标准差	0.739431	0.49492	0.35592	0.59523	0.59523	0.35616	0.35769	4.2E-16
	最优值	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
F12	平均值	0	0	0	0	0	0	0	0
	标准差	0	0	0	0	0	0	0	0
	最优值	0	0	0	0	0	0	0	0

研究表明，平均值、标准差和最优值三个指标不能够全面的评价算法的性能。考虑到假设检验具

有评价算法相似性的能力，本文采用 Wilcoxon 秩和检验对 XSSA 进行验证，判断其与其它所提及的 7

种算法之间的差异性。显著水平设置为 5%，检验结果展示在表 5 中，算法相似度较高的用 NaN 表示，无法比较的用“-”代替。可以看出，XSSA 与各算

法均存在显著性差异，尤其和 SSA 变体以外的四种算法相比，仅个别相似。这进一步验证了 HSSA 算法的独特性。

表 5 测试函数 Wilcoxon 秩和检验

Table 5 Wilcoxon rank sum test of test function

函数	BOA	GWO	IGWO	MSSCS	SSA	CSSA	IHSSA
F1	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NaN	NaN	NaN
F2	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NaN	NaN	4.57E-12
F3	3.02E-11	3.02E-11	3.02E-11	1.41E-09	5.79E-01	8.53E-01	2.64E-01
F4	3.02E-11	3.02E-11	3.02E-11	3.02E-11	6.20E-01	8.42E-01	4.22E-04
F5	1.19E-12	4.19E-02	5.77E-11	1.21E-12	NaN	NaN	NaN
F6	1.21E-12	3.77E-13	6.12E-14	1.17E-12	NaN	NaN	NaN
F7	1.21E-12	4.19E-02	1.10E-02	3.34E-01	NaN	NaN	NaN
F8	3.02E-11	3.02E-11	1.96E-01	-	NaN	2.32E-02	1.04E-04
F9	3.02E-11	3.02E-11	3.02E-11	2.27E-03	3.64E-02	3.63E-01	8.30E-01
F10	1.21E-12	NaN	NaN	-	NaN	NaN	NaN
F11	2.37E-12	2.37E-12	1.00E+00	-	1.00E-01	5.84E-01	1.01E-11
F12	1.21E-12	1.21E-12	1.21E-12	1.69E-14	NaN	NaN	NaN

此外，将记录的 7 种算法的 30 次仿真实验得到的各测试函数的平均值和标准差采用 Friedman’s 检验^[9]的综合排名记录在表 6 中。可以看出，XSSA

以均值 2.5417 排名第一，再次证明其性能最优。而四种 SSA 变体都好过三种经典群智能算法，亦可见麻雀优化算法的优越性。

表 6 Friedman 统计结果

Table 6 Friedman statistical results

算法	BOA	GWO	IGWO	SSA	CSSA	IHSSA	XSSA
平均排名	6.3750	5.5833	4.3750	3.5625	2.7500	2.8125	2.5417

4 结束语

为了改善基础 SSA 的寻优不稳定，容易受局部最优吸引的缺点，本文提出了一种融合排序弹性碰撞的改进麻雀搜索算法 XSSA，并在 12 个基础函数上和其它 7 种算法进行性能的比较，得出如下结论：

(1) XSSA 的寻优性能较优，可以有效地应对陷入局部最优的情况。在 9 个函数中排名第一，2 个排名第二，Friedman 统计结果更是直观看出其优势。

(2) XSSA 得到的收敛精度较好，收敛速度较快。改进的 ICMIC 进行种群初始化，减少了重叠点，增加了分布的分散程度；模拟麻雀的飞行特点融入了高斯随机游走策略平衡算法的全局和局部搜索能力；发现者更新阶段加入了全新的排序

弹性碰撞策略，通过配对的撞击减少局部最优对全局寻优的不利影响。进而使得 XSSA 性能较好。

(3) XSSA 各策略贡献明显，且有机结合后达到最好的效果。消融实验证明了各阶段较原始 SSA 有性能的提升，但是 XSSA 能够在 12 个函数上实现对于各种组合策略的超越。

(4) 总的来数，XSSA 有较好的性能，且具有一定的独特性，但是仍存在寻优速度不够快（如 F2），在 F4、F9、F11 中精度不够高等问题。由此可见，仍需要尝试发掘更优的策略提高收敛速度，增加求解精度。下一步的计划是尝试在各阶段引入新的策略，减少寻优的时间消耗，增加寻优精度，并在神经网络的参数优化、无线传感网络覆盖优化、工程优化问题等领域进一步验证改

进的可行性和有效性。

参考文献:

- [1] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]//MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Piscataway: IEEE, 1995: 39-43.
- [2] Kennedy J, Eberhart R. Particle swarm optimization[C]//ICNN'95. Proceedings of international conference on neural networks. Piscataway: IEEE, 1995, 4: 1942-1948.
- [3] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on evolutionary computation, 1997, 1(1): 53-66.
- [4] 刘生建, 杨艳, 周永权. 一种群体智能算法: 狮群算法[J]. 模式识别与人工智能, 2018, 31(5): 431-441.
LIU S J, YANG Y, ZHOU Y Q. A Swarm Intelligence Algorithm – Lion Swarm Optimization. Pattern Recognition and Artificial Intelligence, 2018, 31(5): 431-441.
- [5] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization[J]. Soft Computing, 2019, 23(3): 715-734.
- [6] 朱佳莹, 高茂庭. 融合粒子群与改进蚁群算法的 AUV 路径规划算法[J]. 计算机工程与应用, 2021, 57(6): 267-273.
Zhu J Y, Gao M T. AUV path planning based on particle swarm optimization and improved ant colony optimization[J]. Computer Engineering and Applications, 2021, 57(6): 267-273.
- [7] 胡晓敏, 王明丰, 张首荣等. 用于文本聚类的新型差分进化粒子群算法[J]. 计算机工程与应用, 2021, 57(4): 61-67.
Hu X M, Wang M F, Zhang S. New differential evolution with particle swarm optimization algorithm for text clustering[J]. Computer Engineering and Applications, 2021, 57(4): 61-67.
- [8] 黄元元, 钱斌, 吴丽萍, 等. 混合果蝇算法求解分布式异构并行机调度[J]. 控制工程, 2020, 27(2): 254-263.
HUANG Y Y, QIAN B. Hybrid fruit fly algorithm for distributed heterogeneous parallel machine scheduling[J]. Control Engineering of China, 2020, 27(02): 254-263.
- [9] Zhu D, Xie L, Zhou C. K-Means Segmentation of Underwater Image Based on Improved Manta Ray Algorithm[J]. Computational Intelligence and Neuroscience, 2022, 2022.
- [10] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. Systems Science & Control Engineering, 2020, 8(1): 22-34.
- [11] 张伟康, 刘升, 任春慧. 混合策略改进的麻雀搜索算法[J]. 计算机工程与应用, 2021, 第 57 卷(24): 74-82.
Zhang W K, Liu S, Ren C H. Mixed Strategy Improved Sparrow Search Algorithm[J]. Computer Engineering and Applications, 2021, 57(24): 74-82.
- [12] Ouyang C, Zhu D, Wang F. A learning sparrow search algorithm[J]. Computational Intelligence and Neuroscience, 2021, 2021.
- [13] Liang Q, Chen B, Wu H, et al. A novel modified sparrow search algorithm with application in side lobe level reduction of linear antenna array[J]. Wireless Communications and Mobile Computing, 2021, 2021.
- [14] Liu G, Shu C, Liang Z, et al. A modified sparrow search algorithm with application in 3d route planning for UAV[J]. Sensors, 2021, 21(4): 1224.
- [15] Wang Z, Huang X, Zhu D. A Multistrategy-Integrated Learning Sparrow Search Algorithm and Optimization of Engineering Problems[J]. Computational Intelligence and Neuroscience, 2022, 2022.
- [16] 王振东, 汪嘉宝, 李大海. 一种增强型麻雀搜索算法的无线传感器网络覆盖优化研究[J]. 传感技术学报, 2021, 34(6): 818-828.
Wang Z D, Wang J B, Li D H. Study on WSN Optimization Coverage of an Enhanced Sparrow Search Algorithm[J]. Chinese Journal of Sensors and Actuators, 2021, 34(6): 818-828.
- [17] He D, He C, Jiang L G, et al. Chaotic characteristics of a one-dimensional iterative map with infinite collapses[J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2001, 48(7): 900-906.
- [18] 王依柔, 张达敏. 融合正弦余弦和无限折叠迭代混沌映射的蝴蝶优化算法[J]. 模式识别与人工智能, 2020, 33(7): 660-669.
Wang Y R, Zhang D M. Butterfly Optimization Algorithm Combining Sine Cosine and Iterative Chaotic Map with Infinite Collapses[J]. Pattern Recognition and Artificial Intelligence, 2020, 33(7): 660-669.
- [19] Peng H, Zeng Z, Deng C, et al. Multi-strategy serial cuckoo search algorithm for global optimization[J]. Knowledge-Based Systems, 2021, 214: 106729.
- [20] Deng H, Peng L, Zhang H, et al. Ranking-based biased learning swarm optimizer for large-scale optimization[J]. Information Sciences, 2019, 493: 120-137.
- [21] Zhu D, Huang Z, Xie L, et al. Improved Particle Swarm Based on Elastic Collision for DNA Coding Optimization Design[J]. IEEE Access, 2022.

[22] 吕鑫, 慕晓冬, 张钧, 等. 混沌麻雀搜索优化算法[J]. 北京航空航天大学学报, 2021, 47(8): 1712-1720.
 LYU X, MU X D, ZHANG J, et al. Chaos sparrow search optimization algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2021, 47(8): 1712-172.

[23] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in engineering software, 2014, 69: 46-61.

[24] Nadimi-Shahraki M H, Taghian S, Mirjalili S. An improved grey wolf optimizer for solving engineering problems[J]. Expert Systems with Applications, 2021, 166: 113917.

	<p>王子恺 (1998—), 男, 安徽合肥人, 硕士研究生, 主要研究方向为群智能算法、工业智能等。</p> <p>WANG Zikai, born in 1998, M.S. candidate. His research interests include swarm intelligence algorithm, industrial intelligence, etc.</p>
	<p>黄学雨 (1970—), 男, 江西赣州人, 教授, 主要研究方向为工业智能等。</p> <p>HUANG Xueyu, born in 1970, professor. His research interests include industrial intelligence, etc.</p>
	<p>朱东林 (1997—), 男, 江西九江人, 硕士研究生, 主要研究方向为群智能算法、图像处理等。</p> <p>Zhu Donglin, born in 1997, M.S. candidate. His research interests include swarm intelligence algorithm, image processing etc.</p>
	<p>郭伟 (1997—), 男, 安徽阜阳人, 硕士研究生, 主要研究方向为无线传感网络、智能算法等。</p> <p>GUO Wei, born in 1997, M.S. candidate. His research interests include wireless sensor network, swarm intelligence algorithm, etc.</p>