

基于多阶段调度框架的麻雀搜索优化算法

王 毅¹, 郑宏志¹, 黄 欣^{2*}, 洪国栋¹, 闫小婕¹

(1. 西北大学信息科学与技术学院, 陕西西安 710127; 2. 西安交通大学, 陕西西安 710049)

摘要: 本文提出一种多阶段调度框架, 实现对麻雀种群的初始位置、觅食、侦查与反捕食不同阶段的多策略调度。利用Halton序列与Tent映射提升种群个体质量与初始位置的分布均匀性。在觅食阶段, 针对发现者与加入者因位置争夺导致种群质量劣化, 设计最佳适配比调控二者数量关系, 对超出适配比的加入者采用碰撞反弹算子改变其优化轨迹。满足适配比后则通过侦查判断是否存在天敌, 若有则进入反捕食阶段, 并利用Levy飞行并结合指数分布设计随机迁移机制, 生成潜在的全局最优解区域; 当连续多次没有发现天敌时为避免种群陷入局部极值, 建立模拟预警机制并采用蝗虫算法进行多路径开发, 避免寻优方向单一化。不同策略与机制的交替运行、协同调度, 平衡了算法的多样性与收敛性。实验结果表明, 与最近麻雀变体算法和元启发改进算法相比, 该算法在寻优效率与收敛精度上显著优于对比方法。

关键词: 麻雀搜索算法; 多阶段调度框架; 最佳适配机制; 模拟预警机制; 随机迁移机制

基金项目: 国家自然科学基金重大仪器专项(No.42027806); 国家重点研发计划项目(No.2018YFC1504705); 国家自然科学基金(No.61731015, No.61402517); 陕西省重点研发计划项目(No.2022GY-331); 陕西省自然科学基金(No.2018JM6029)

中图分类号: TP182; TP391

文献标识码: A

文章编号: 0372-2112(2024)09-3086-11

电子学报URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20230152

Sparrow Search Optimization Algorithm Based on Multi-Stage Scheduling Framework

WANG Yi¹, ZHENG Hong-zhi¹, HUANG Xin^{2*}, HONG Guo-dong¹, YAN Xiao-jie¹

(1. School of Information Science and Technology, Northwestern University, Xi'an, Shaanxi 710127, China;

2. Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China)

Abstract: This paper proposes a multi-stage scheduling framework to realize multi-strategy scheduling of sparrow populations in different stages of initial location, foraging, detection, and anti-predation. Halton sequence and Tent mapping are used to improve the quality of the population individuals and the distribution uniformity of initial position. In the foraging stage, aiming at the deterioration of the population quality caused by the position competition between the finder and the joiner, the best fit ratio is designed to control the quantitative relationship between the two, and the collision rebound operator is used to change the optimal trajectory of the joiner beyond the fit ratio. After the adaptation ratio is met, judge whether there is a natural enemy through investigation, and if there is, enter the anti-predation stage, and use Levy flight and combine exponential distribution to design a random migration mechanism to generate a potential global optimal solution area; when no natural enemy is found for many times in a row in order to prevent the population from falling into local extremum, an early warning mechanism is established and the locust algorithm is used for multi-path development to avoid a single optimization direction. The alternate operation and coordinated scheduling of different strategies and mechanisms balance the diversity and convergence of the algorithm. Experimental results show that, compared with the latest sparrow variant algorithm and meta-heuristic improved algorithm, the algorithm is significantly better than the comparison methods in terms of optimization efficiency and convergence accuracy.

Key words: sparrow search algorithm; multi-stage scheduling framework; best adaptation mechanism; simulated early warning mechanism; random migration mechanism

Foundation Item(s): Major Instrument Project of National Natural Science Foundation of China (No.42027806); National Key Research and Development Project (No. 2018YFC1504705); National Natural Science Foundation of China (No.61731015, No.61402517); Key Research and Development Program of Shaanxi (No.2022GY-331); Natural Science Foundation of Shaanxi Province (No.2018JM6029)

1 引言

2020年提出的麻雀搜索算法^[1](Sparrow Search Algorithm, SSA)具有随机性觅食行为和反捕食行为特点,适用于复杂问题求解.但仍存在易陷入局部最优,搜索速度慢的问题,国内外学者进行了多方面的研究与改进.通过对现有文献分析发现,影响SSA算法性能的因素主要包括提升种群初始质量、自适应参数调节与改进策略三个方面.

(1)初始化种群质量.种群个体质量与初始位置是影响算法的搜索速度与寻优效率的首要因素.与其他元启发算法类似,麻雀搜索算法也存在初始个体质量差异大,分布空间随机等共性,通常采用对立学习^[2]、混沌序列^[3]等手段来提升种群质量. Xu^[4]提出一种改进的基于密度的初始化策略以生成具有一定随机性的均匀初始种群. Zhang^[5]利用轮盘赌选择算子优化麻雀初始种群,加快算法全局搜索效率. Li^[6]采用Sin混沌映射初始化麻雀种群的位置,提高了算法初始解的质量以及搜索性能. Huang^[7]利用Tent混沌映射对麻雀种群进行初始化,从而获得分布均匀,更有利于聚类的初始种群.

(2)自适应参数调节.参数设置对算法在觅食与反捕食间的自适应平衡具有重要作用.在麻雀搜索算法中,随着种群迭代,算法全局寻优效率会显著降低.通过自适应调整步长控制参数来调控搜索范围和局部搜索精度是一种有效途径^[8,9].此外,也可设计惯性权重因子对麻雀个体速度进行控制,提高全局寻优性能的同时增强了局部极值的逃逸能力^[10].

(3)改进策略.随着麻雀种群搜索过程中迭代次数增加,种群多样性降低,这将导致SSA易陷入局部极值.结合其他元启发式算法的优秀策略进行改进是必要且可行的.学习行为选择策略^[11]、正余弦策略^[12]、直觉模糊多属性决策^[13]、随机跟随策略^[14]等具有较强扰动能力的策略可以很好的维持种群多样性,已被许多学者引入到麻雀搜索算法中.

以上改进方法在不同程度上增强了SSA算法的性能,但却忽略算法自身机制的重要性.麻雀搜索算法的机制通过发现者、加入者与预警者这三种类别,控制着种群的初始位置、觅食、侦查与反捕食过程,很大程度上决定了种群的生存能力.因此,机制是否完善、优异对算法本身至关重要.因此,本文针对麻雀行为的不同阶段,设计多种策略,提出基于多阶段调度框架的麻雀

搜索优化算法(Sparrow Search optimization Algorithm based on Multi-stage Scheduling Framework, MSFSSA),以提升算法的全局寻优效率和收敛速度.

2 多阶段调度框架的SSA算法

本文针对麻雀搜索算法在初始位置、觅食、侦查与反捕食四个阶段,分别设计了初始种群优化策略、最佳适配机制、随机迁移策略与模拟预警机制,建立了多阶段调度框架,如图1所示.

在框架内采用Halton序列和Tent映射优化初始种

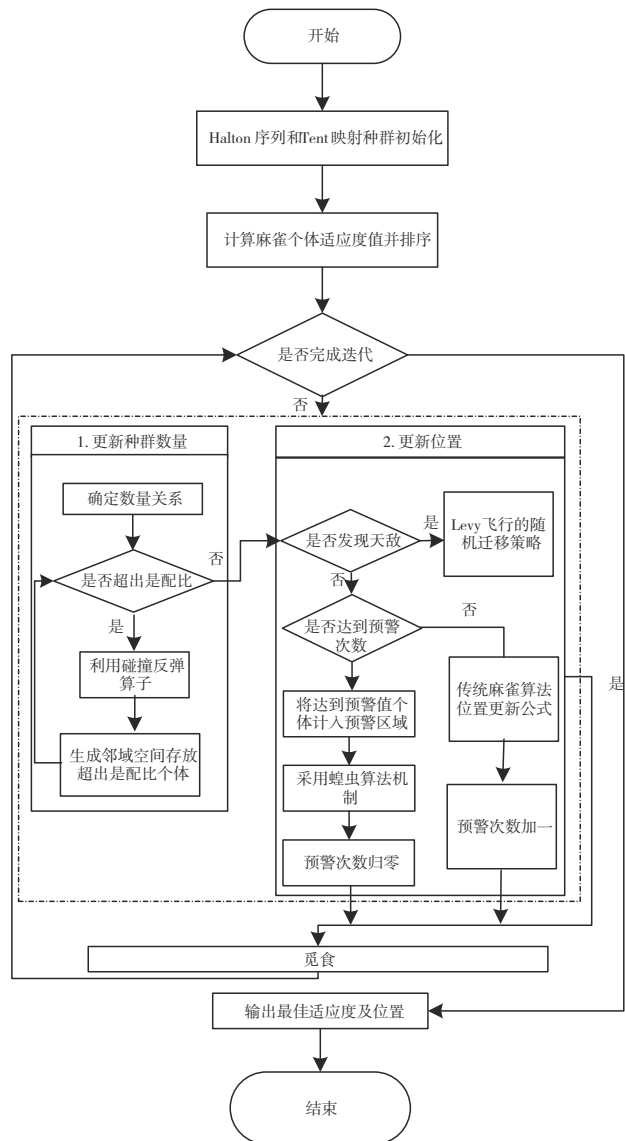


图1 MSFSSA算法框架

群,提高种群个体均匀分布性,通过计算个体适应度值及排序确定麻雀个体质量与种群数量.为缓解麻雀种群因位置争夺而导致种群质量劣化,建立最佳适配机制并依据该机制判定发现者与加入者数量是否超出最佳适配比,对超出适配比的加入者采用碰撞反弹算子改变其优化轨迹并生成新的邻域空间,将邻域空间中的麻雀个体进行适应度排序并再次划分发现者与加入者的数量关系,保证种群稳定的同时持续勘探、增强种群多样性,避免无效内耗.对满足适配比的麻雀种群则进入侦查阶段.在该阶段中若预警者发现天敌,则进入反捕食阶段并采用随机迁移策略扩大搜索范围、增强全局寻优能力;若麻雀个体连续多次未发现天敌时,则将其个体计入预警区域并采用蝗虫算法进行多路径寻优,避免寻优方向单一化,寻优后将该个体预警次数归零;若种群个体迭代未达到预警次数时,则采用原麻雀算法更新其位置并将其预警次数加一,直到迭代结束.通过麻雀搜索算法中不同行为阶段的四种策略与机制,向全局最优与邻域最优学习,多策略机制交替运行、协同合作,平衡算法的多样性与收敛性.

2.1 Halton序列和Tent映射种群初始化

麻雀搜索算法采用rand随机函数生成初始化种群的方法易使种群分布不均匀,多样性不足,不利于后代优化.因此,本文采用Halton序列和Tent映射来提高种群质量.Halton序列属于低差异序列一种,二维Halton序列^[15]的实现过程为选取两个质数作为基础量,通过对两个基础量不断切分,从而组合成一系列均匀分布且不重复的点.然后再通过改进的Tent映射避免小周期问题的同时将其扩散到整个解空间.

假设种群规模为 n ,通过如下切分过程得到二维均匀的Halton序列:

$$n = \sum_{i=0}^m b_i \cdot p^i = b_m \cdot p^m + \dots + b_1 \cdot p^1 + b_0 \quad (1)$$

$$\theta(n) = b_0 p^{-1} + b_1 p^{-2} + \dots + b_m p^{-m-1} \quad (2)$$

$$H(n) = [\theta_1(n), \theta_2(n)] \quad (3)$$

其中, $n \in [1, N]$ 为任意整数, p 是大于等于2的质数, $b_i \in \{0, 1, 2, \dots, p-1\}$ 为常数, p 为Halton序列基础量, $\theta(n)$ 是定义的序列函数, $H(n)$ 为最后得到的二维均匀Halton序列.

将该序列通过Tent映射生成混沌序列 y .同时针对原始Tent映射的小周期和周期点不稳定问题,在保留原有Tent映射随机性、遍历性和规律性的特点上引入随机变量^[16] $\text{rand}(0, 1) \times 1/N$,改进后表达式如下:

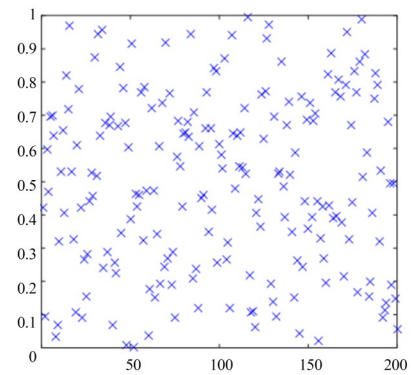
$$y_{i+1} = \begin{cases} 2y_i + \text{rand}(0, 1) \times \frac{1}{N} & , y_i \in [0, 0.5] \\ 2(1 - y_i) + \text{rand}(0, 1) \times \frac{1}{N} & , y_i \in (0.5, 1] \end{cases} \quad (4)$$

在式(4)中, N 代表产生Tent混沌序列的总数,即种群中麻雀个数, $\text{rand}(0, 1)$ 表示 $[0, 1]$ 之间的随机数;将生成的Tent混沌序列 y 置于麻雀的求解空间中,公式如下:

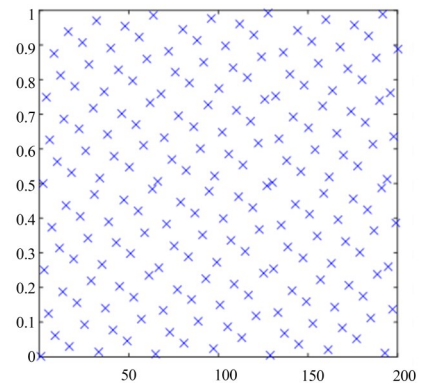
$$x_i = x_{lb} + (x_{ub} - x_{lb}) \times y_i \quad (5)$$

其中, x_i 表示生成第 i 个麻雀的初始位置, x_{ub} 和 x_{lb} 分别代表麻雀位置的上限和下限.

由图2所示,经Halton序列和Tent映射后种群分布更加均匀扩散至解空间.



(a) 随机生成的种群个体分布



(b) 与Halton序列和Tent映射后生成的种群个体分布

图2 种群个体分布

2.2 最佳适配机制

在麻雀搜索算法中,发现者具有较高的能量储备,适应度值高,主要为加入者提供觅食区域和方向.加入者跟随适应度值最高的发现者寻找食物以获得自身能量储备,提升个体适应度,进而争夺食物并替代发现者位置.由此可知,发现者与加入者的数量比极大影响着种群稳定性和勘探能力.

因此,本文在多阶段调度框架中设计最佳适配机制,避免在觅食过程中麻雀种群过于“内聚”或“发散”.当发现者与加入者数量关系超过适配比 s 时,对超出的加入者执行碰撞反弹算子改变其寻优轨迹,以缓解随加入者的数量不断增加,发现者与加入者之间的频繁

竞争,避免种群陷入无意义的内耗.为获取最佳适配比,本算法设置不同数量比 s ,并记录其寻优率,如图 3 所示.图中可知,通过对加入者数量的动态调整以及相应寻优率变化可知,当发现者与加入者数量比 $s=1:3$ 时,寻优率最佳,小于或大于该数量比时,算法寻优率较低.随后,将超出适配比的加入者采用碰撞反弹算子改变其轨迹,生成新的邻域空间,增强种群多样性.

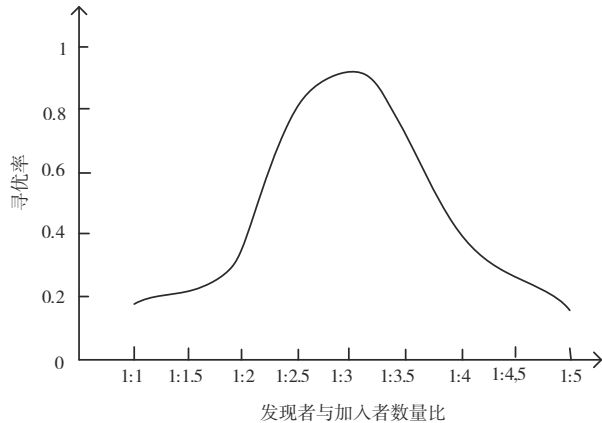


图 3 适配比 s 的寻优率分布图

王^[17]的碰撞反弹算子由经典物理中笛卡尔的动量守恒定理而来.当两个运动的物体相碰撞时,每个物体的移动方向和速度可由下式计算得出:

$$V'_{ij} = \frac{(m_i - m_j)v_i + 2m_j v_j}{m_i + m_j} \quad (6)$$

$$m_i = \frac{1}{f_i - \min f + \Delta} \quad (7)$$

式(6),中 V'_{ij} 表示麻雀个体 i 和个体 j 碰撞后个体 i 的速度方向向量; v_i 表示当前麻雀个体的速度方向向量; v_j 表示与当前个体发生碰撞的个体的速度方向向量. m_i 和 m_j 分别表示个体 i 和 j 的质量;式(7)是个体质量计算公式,其中 f_i 表示个体 i 的适应度, Δ 是一个防止分母为零的常量,本文设 $\Delta=0.1$.若当前个体和多个个体碰撞,其最终的移动方向和速度向量由下式计算:

$$u'_i = \sum_{j=1}^{Z_{\text{coll},i}} v'_{ij} \quad (8)$$

式(8)中, u'_i 表示粒子 i 最终的速度方向向量; $Z_{\text{coll},i}$ 是一个与粒子 i 碰撞的粒子所构成的集合.

2.3 基于 Levy 飞行的随机迁移策略

麻雀种群完成最佳适配机制后则进入下一过程的侦查阶段,原始麻雀搜索算法根据有无天敌的存在进行位置更新:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j} \cdot \exp(-\frac{i}{\alpha \cdot t_{\max}}) & , R < ST \\ X_{i,j} + Q \cdot L & , R \geq ST \end{cases} \quad (9)$$

其中, t 代表当前迭代数, $j=1, 2, \dots, D$; t_{\max} 表示最大迭代

次数; $X_{i,j}$ 表示第 i 个麻雀在第 j 维中的位置信息; $\alpha \in (0, 1]$ 是一个随机数. $R (R \in [0, 1])$ 和 $ST (ST \in [0.5, 1])$ 分别表示为预警值和安全值. Q 是服从正态分布的随机数. L 表示一个 $1 \times d$ 元素为 1 的矩阵.

当 $R < ST$ 时,即觅食环境周围没有天敌,发现者可以进行全局搜索;当 $R \geq ST$ 时,预警者发现天敌时警报,同时迅速向安全区域移动以获得更好的位置进行觅食,处于种群中间的麻雀则随机行走靠近别的麻雀,即反捕食行为.但原始算法中反捕食阶段具有初始位置冗僻,搜索范围狭窄的问题,因此选用 Levy 飞行对该阶段进行优化.尹等^[18]将其用于麻雀搜索算法中,优化侦察预警的麻雀位置更新公式,提升了算法的寻优能力. Levy 飞行公式如下:

$$X_{i,j}^{t+1} = \begin{cases} \text{Levy}(d) \cdot X_{\text{best}}^t + \beta \cdot |X_{i,j}^t - \text{Levy}(d) \cdot X_{\text{best}}^t|, & f_i > f_g \\ X_{i,j}^t + K \cdot \left| \frac{X_{i,j}^t - X_{\text{worst}}^t}{(f_i - f_w) + \varepsilon} \right|, & f_i = f_g \end{cases} \quad (10)$$

其中 d 为向量维度, Levy 计算公式如下:

$$\text{Levy}(d) = 0.01 \times \left(\frac{r_1 \cdot \sigma}{|r_2|^{1/\beta}} \right) \quad (11)$$

$$\sigma = \frac{\tau(1 + \beta_0) \times \sin(\frac{\pi \cdot \beta_0}{2})}{\tau(1 + \beta_0) \times \beta_0 \times 2^{2/\beta_0}} \quad (12)$$

其中, τ 为伽马函数, $\beta = \beta_0 = 1.5$, $r_1 \in (0, 1)$, $r_2 \in (0, 1)$.

Levy 飞行能够使算法跳出局部最优,提高种群总体寻优效率.但由于麻雀种群中初始加入者的低能量限制,原有的 Levy 飞行方法寻优速度缓慢,因此,本文采用指数分布^[19]方式对步长进行优化,实现“指数爆炸”式扩张搜索空间,本文将指数分布与 Levy 飞行进行有机结合,构造随机迁移策略.对于天敌存在的情况,将采用式(13)进行位置更新,即将改进的式(13)替换传统式(11)进行全局寻优,公式如下:

$$\text{Levy}(d) = 0.01 \times \left(\frac{r_1^{-\sigma}}{|r_2|^{1/\beta}} \right) \quad (13)$$

2.4 模拟预警机制

本文在侦查阶段没有遇见天敌的情况下首次设计出模拟预警机制,以此提高 MSFSSA 算法在该阶段的深度开发与局部精细搜索能力.

虽然天敌的存在一定程度增强了算法的全局搜索能力,但当种群长时间处于无天敌状态下则会导致迭代迟滞问题,为此,本文设计了模拟预警机制用于算法后期的深度开发与局部精细搜索.若种群个体连续 A 次未发现天敌时,其种群适应度值将会随迭代次数随之衰减易陷入局部极值,为提升种群个体的局部逃逸能力,启动预警机制.将第 A 次未发现天敌的种群个体

计入预警区域并采用蝗虫算法进行多路径寻优并将寻优后将该个体预警次数归零;若种群个体迭代未达到预警次数 A 时,则采用原麻雀算法更新其位置并将其预警次数加一,直到迭代结束.其中 A 取5效果最好.

由于蝗虫算法^[20]具有良好的种群优异值及寻优能力,且其更新公式中考虑了重力、风速等多种因素,能够更好的摆脱局部最优陷阱,因此将其结合在模拟预警机制中.蝗虫算法位置更新公式如下:

$$X_i^d = c \left(\sum_{j=1, j \neq i}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d| \frac{x_j - x_i}{d_{ij}}) \right) + T_d \quad (14)$$

其中, ub_d 与 lb_d 分别为 D 维搜索空间的上下界, T_d 为当前种群中的最优个体,不考虑重力影响且假设风向总是指向最优解所在位置, c 为线性递减系数,表达式如下:

$$c = c_{\max} - l \cdot \frac{c_{\max} - c_{\min}}{L} \quad (15)$$

其中, l 表示算法当前迭代次数, L 表示最大迭代次数,本文取 $c_{\max} = 1, c_{\min} = 0.000\ 01$. $s(r)$ 为定义个体间相互作用力的函数:

$$s(r) = f e^{\frac{r}{l}} - e^{-r} \quad (16)$$

其中, f 表示引力强度, l 为引力尺度范围,一般设为 $f=0.5, l=1.5$.

MSFSSA算法伪代码如算法1所示.

3 函数测试与结果分析

3.1 实验环境配置及参数设置

为了评估MSFSSA算法性能,实验从文献[21]和IEEE CEC2017选取18个测试函数包括单峰、多峰与复合函数进行多方位测试,其中测试函数 $F_1 \sim F_6$ 为单峰函数, $F_7 \sim F_{12}$ 为多峰函数, $F_{13} \sim F_{18}$ 为复合函数.具体测试函数如表1与表2所示.

实验环境采用Inter i7-6700CPU,主频3.40 GHz和内存16 GB的PC机,操作系统为64位Windows10,编程语言采用MATLAB 2019B.对比算法选取了近年的SSA变体算法和主流元启发改进算法,包括麻雀搜索算法^[11](SSA)、一种改进的麻雀搜索优化算法^[18](Improved Sparrow Search optimization Algorithm, ISSA)、混沌麻雀搜索优化算法^[21](Chaos Sparrow Search Optimization Algorithm, CSSOA)、混合均值中心反向学习粒子群优化算法^[22](Hybrid mean Center Opposition-based learning Particle Swarm Optimization, HSOPSO)、鲸鱼优化算法^[23](Whale Optimization Algorithm, WOA)、基于改进蝗虫优化算法的特征选择方法^[24](Improved Grasshopper Optimization Algorithm, IGOA)等6种对比算法.

算法1 MSFSSA算法的伪代码

输入:种群规模 N 、迭代次数 t_{\max} 、适配比 s 、预警次数 A

```

1. Halton序列和Tent映射初始化种群位置
2. For  $i=1:N$  Do
3.     计算麻雀个体的适应度值并排序
4. End for
5. 计算最优个体极值 $P_{\text{best}}$ 、最优个体适应度值 $f_g$ 、最优个体位置 $X_{\text{best}}$ 、最劣个体极值 $P_{\text{worst}}$ 、最优个体适应度值 $f_w$ 、最优个体位置 $X_{\text{worst}}$ 
6. 设置初始发现者与加入者的数量关系
7. For  $t=1:t_{\max}$  Do
8.     For  $i=1:N$  Do
9.         if  $s>1:3$ 
10.            根据式(6)-(8)分别更新发现者和加入者的速度及数量关系,生成邻域空间
11.        Else  $s+1$ 
12.        End if
13.    End for
14.    if  $R \geq ST$ 
15.        根据随机迁移策略更新种群位置
16.    Else
17.        if  $A \geq 5$ 
18.            将达到预警值个体计入预警空间
19.            采用蝗虫算法机制更新种群位置
20.             $A=0$ 
21.        Else
22.            根据式(9)更新种群位置
23.             $A+1$ 
24.        End if
25.    End if
26.    更新 $P_{\text{best}}$ 、 $f_g$ 、 $X_{\text{best}}$ 、 $P_{\text{worst}}$ 、 $f_w$ 、 $X_{\text{worst}}$ 
27. End for
输出 $f_g$ 

```

为遵循实验公平,所有算法将采用统一数据,设置种群规模为30,解空间维度为30,迭代次数为100,每个函数独立运行30次,最大评估次数MaxFEs=3 000 D.本文算法MSFSSA相关参数设置如下:发现者数量占整个种群的20%,预警者占整个种群的20%,安全阈值 $ST = 0.8$,Levy飞行中 $\beta = \beta_0 = 1.5$,蝗虫算法中 $c_{\max} = 1, c_{\min} = 0.000\ 01$,吸引力强度 $f=0.5$,吸引力步长 $l=1.5$.对比算法的各类参数采用原文设置.所有算法在18个测试函数上运行30次的平均值和标准差进行对比,实验结果如表2所示.

3.2 实验结果与分析

这里将30次实验结果取平均值和标准差后,通过平均值评价算法收敛精度,标准差评价算法稳定性,如表3所示.加粗数值表示各类算法在相应测试函数上

表 1 测试函数

函数类型	函数表达式	维度	搜索空间	极值
单峰函数	$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100,100]^n$	0
	$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10,10]^n$	0
	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100,100]^n$	0
	$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	30	$[-100,100]^n$	0
	$F_5(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]^n$	0
	$F_6(x) = \sum_{i=1}^n -x \sin(\sqrt{ x_i })$	30	$[-500,500]^n$	-418.982 9n
多峰函数	$F_7(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]^n$	0
	$F_8(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	$[-32, 32]^n$	0
	$F_9(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]^n$	0
	$F_{10}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	$[-65, 65]^2$	1
	$F_{11}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2\right)$	6	$[0, 1]^4$	-3.32
	$F_{12}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]^{10}$	-10.536

表 2 CEC2017 复合测试函数

函数类型	函数表达式	维度	搜索空间	极值
CEC2017 复合函数	$F_{13}(x) = \text{Composition Function 1 } (N=3)$	30/50/100	$[-100, 100]$	2 100
	$F_{14}(x) = \text{Composition Function 2 } (N=4)$	30/50/100	$[-100, 100]$	2 300
	$F_{15}(x) = \text{Composition Function 3 } (N=4)$	30/50/100	$[-100, 100]$	2 400
	$F_{16}(x) = \text{Composition Function 4 } (N=5)$	30/50/100	$[-100, 100]$	2 500
	$F_{17}(x) = \text{Composition Function 5 } (N=6)$	30/50/100	$[-100, 100]$	2 700
	$F_{18}(x) = \text{Composition Function 6 } (N=6)$	30/50/100	$[-100, 100]$	2 800

的最优值,最后一行为相应算法在所有测试函数上取得最优均值与标准差的结果个数。

从表 3 可以看出,MSFSSA 算法在 $F_1 \sim F_{18}$ 的 18 个测试函数上取得 14 个最优均值的结果,而 SSA、CSSOA、ISSA、HCOPSO、WOA 和 IGOA 分别在 1 个、4 个、3 个、0 个、0 个和 3 个测试函数上取得最优均值,表明 MSFSSA 算法具有更好的综合寻优能力。为进一步分析算法性能,这里对 7 种算法在单峰、多峰和复合 18 个测试函数上进行求解精度与稳定性分析。

3.2.1 算法性能对比分析

单峰函数的求解精度与稳定性分析 由表 3 可知,在单峰函数 $F_1 \sim F_6$ 中,除函数 F_5 外,其他 5 个测试函数上 MSFSSA 均取得了最优均值。获得的最优均值与同类麻

雀算法相比均至少提升 20 个数量级以上,更是远超出其他启发算法。ISSA 取得了三个函数的最优标准差,但并未取得任何最优均值,表明 ISSA 找到的最优解较为集中,但却并未找到精确的全局最优解;IGOA 主要在 F_5 上取得最优均值与标准差,表明该算法仅在某个特定函数上具有寻优性能。MSFSSA 算法在单峰函数上的表现出优异的收敛速度和寻优性能;同时波动幅度较小的标准差表明该算法的稳定性较强,优于所有对比算法。

多峰函数的求解精度与稳定性分析 多峰函数产生的多个局部最优解,表现为一种全局最优预测值,因此常用来检测算法的全局搜索寻优能力。在多峰函数 $F_7 \sim F_{12}$ 上,MSFSSA 在 F_7, F_9, F_{10}, F_{11} 上取得了最优均值,

而 CSSOA 在 F_8 、 F_9 、 F_{12} 上获得仅次于 MSFSSA 的最优均值,但在标准差的表现优于 MSFSSA 算法. ISSA 与 IGOA 只能在 F_9 和 F_{10} 上表现出较好的寻优能力,而对于 SSA、HSOPSO、WOA 在 6 个多峰函数上均没有获得最优均值

与标准差,性能表现欠佳. 结果表明,MSFSSA 算法所具有的最佳适配机制与随机迁移策略可有效扩大搜索范围的同时提升全局寻优能力,缓解在觅食过程中麻雀种群过于“内聚”或“发散”而产生的内耗,提升寻优效率.

表 3 MSFSSA 与对比算法的函数测试结果

函数	评价指标	SSA	CSSOA	ISSA	HCOPSO	WOA	IGOA	MSFSSA
F_1	平均值	4.654×10^{-32}	6.124×10^{-18}	1.840×10^{-178}	2.641×10^{-03}	1.561×10^{-04}	2.113×10^{-04}	5.146×10^{-232}
	标准差	2.043×10^{-18}	3.385×10^{-19}	$0.000 \times 10^{+00}$	4.825×10^{-03}	3.972×10^{-04}	2.428×10^{-04}	$0.000 \times 10^{+00}$
F_2	平均值	3.437×10^{-07}	1.547×10^{-15}	5.776×10^{-90}	7.248×10^{-01}	4.756×10^{-05}	6.243×10^{-05}	5.432×10^{-128}
	标准差	1.253×10^{-07}	4.158×10^{-15}	4.039×10^{-90}	7.483×10^{-01}	6.539×10^{-05}	8.763×10^{-05}	3.427×10^{-97}
F_3	平均值	9.854×10^{-12}	3.217×10^{-40}	6.427×10^{-102}	5.186×10^{-01}	$1.426 \times 10^{+02}$	5.754×10^{-32}	1.597×10^{-130}
	标准差	6.231×10^{-11}	1.486×10^{-65}	$0.000 \times 10^{+00}$	3.247×10^{-01}	$1.348 \times 10^{+02}$	1.182×10^{-34}	$0.000 \times 10^{+00}$
F_4	平均值	3.016×10^{-17}	6.178×10^{-18}	6.913×10^{-90}	5.369×10^{-03}	1.541×10^{-01}	2.684×10^{-06}	4.812×10^{-122}
	标准差	1.273×10^{-17}	2.615×10^{-18}	3.145×10^{-90}	5.153×10^{-03}	2.872×10^{-01}	3.289×10^{-06}	2.395×10^{-122}
F_5	平均值	4.275×10^{-03}	7.124×10^{-04}	3.178×10^{-04}	1.614×10^{-02}	2.969×10^{-03}	3.042×10^{-04}	3.631×10^{-04}
	标准差	3.384×10^{-03}	5.984×10^{-04}	2.681×10^{-04}	7.851×10^{-03}	4.398×10^{-03}	2.681×10^{-04}	3.329×10^{-04}
F_6	平均值	$-8.476 \times 10^{+03}$	$-1.118 \times 10^{+04}$	$-1.465 \times 10^{+49}$	$-2.806 \times 10^{+03}$	$-1.135 \times 10^{+04}$	$-9.574 \times 10^{+02}$	$-2.827 \times 10^{+98}$
	标准差	$5.677 \times 10^{+02}$	$6.828 \times 10^{+02}$	$8.127 \times 10^{+49}$	$5.267 \times 10^{+02}$	$1.588 \times 10^{+03}$	$1.924 \times 10^{+02}$	$4.653 \times 10^{+99}$
F_7	平均值	6.514×10^{-03}	5.796×10^{-03}	7.618×10^{-04}	4.773×10^{-02}	8.832×10^{-03}	4.632×10^{-04}	5.364×10^{-05}
	标准差	3.536×10^{-03}	2.742×10^{-03}	6.522×10^{-04}	1.228×10^{-02}	4.137×10^{-03}	2.857×10^{-04}	3.315×10^{-05}
F_8	平均值	$1.336 \times 10^{+01}$	$1.1461 \times 10^{+00}$	$1.326 \times 10^{+00}$	$5.351 \times 10^{+00}$	$2.447 \times 10^{+00}$	$9.346 \times 10^{+00}$	$4.386 \times 10^{+00}$
	标准差	$5.264 \times 10^{+00}$	5.253×10^{-01}	$4.306 \times 10^{+00}$	$2.903 \times 10^{+00}$	$3.124 \times 10^{+00}$	$4.513 \times 10^{+00}$	8.813×10^{-01}
F_9	平均值	4.372×10^{-08}	$0.000 \times 10^{+00}$	$0.000 \times 10^{+00}$	1.214×10^{-03}	8.648×10^{-07}	$0.000 \times 10^{+00}$	$0.000 \times 10^{+00}$
	标准差	$3.683 \times 10^{+01}$	$0.000 \times 10^{+00}$	$0.000 \times 10^{+00}$	1.372×10^{-03}	2.577×10^{-07}	$0.000 \times 10^{+00}$	$0.000 \times 10^{+00}$
F_{10}	平均值	4.421×10^{-15}	8.882×10^{-16}	8.881×10^{-16}	6.385×10^{-03}	5.136×10^{-05}	8.881×10^{-16}	8.881×10^{-16}
	标准差	5.272×10^{-15}	$0.000 \times 10^{+00}$	$0.000 \times 10^{+00}$	5.043×10^{-03}	1.433×10^{-04}	0.517×10^{-02}	$0.000 \times 10^{+00}$
F_{11}	平均值	$-3.263 \times 10^{+00}$	$-3.304 \times 10^{+00}$	$-2.736 \times 10^{+00}$	$-3.154 \times 10^{+00}$	$-3.069 \times 10^{+00}$	$-2.132 \times 10^{+00}$	$-3.528 \times 10^{+00}$
	标准差	5.985×10^{-02}	4.116×10^{-02}	$4.132 \times 10^{+00}$	4.786×10^{-02}	1.263×10^{-01}	5.308×10^{-01}	7.472×10^{-02}
F_{12}	平均值	$-7.165 \times 10^{+00}$	$-1.047 \times 10^{+01}$	$-8.584 \times 10^{+00}$	$-6.747 \times 10^{+00}$	$-6.071 \times 10^{+00}$	$-1.833 \times 10^{+00}$	$-7.808 \times 10^{+00}$
	标准差	$2.783 \times 10^{+00}$	1.278×10^{-05}	$2.496 \times 10^{+00}$	$1.325 \times 10^{+00}$	$2.126 \times 10^{+00}$	$1.011 \times 10^{+00}$	$2.608 \times 10^{+00}$
F_{13}	平均值	$3.461 \times 10^{+03}$	$3.407 \times 10^{+03}$	$3.452 \times 10^{+03}$	$3.435 \times 10^{+03}$	$3.431 \times 10^{+03}$	$3.413 \times 10^{+03}$	$3.402 \times 10^{+03}$
	标准差	$5.214 \times 10^{+01}$	$4.658 \times 10^{+01}$	$1.353 \times 10^{+01}$	$3.867 \times 10^{+01}$	$5.300 \times 10^{+01}$	$4.254 \times 10^{+01}$	$3.474 \times 10^{+01}$
F_{14}	平均值	$3.493 \times 10^{+03}$	$3.548 \times 10^{+03}$	$3.607 \times 10^{+03}$	$3.551 \times 10^{+03}$	$3.522 \times 10^{+03}$	$3.513 \times 10^{+03}$	$3.466 \times 10^{+03}$
	标准差	$5.642 \times 10^{+01}$	$6.952 \times 10^{+01}$	$3.662 \times 10^{+02}$	$4.540 \times 10^{+01}$	$7.216 \times 10^{+01}$	$4.832 \times 10^{+01}$	$5.342 \times 10^{+01}$
F_{15}	平均值	$3.614 \times 10^{+03}$	$3.487 \times 10^{+03}$	$3.528 \times 10^{+03}$	$3.593 \times 10^{+03}$	$3.586 \times 10^{+03}$	$3.532 \times 10^{+03}$	$3.487 \times 10^{+03}$
	标准差	$7.694 \times 10^{+01}$	$9.413 \times 10^{+01}$	$3.689 \times 10^{+02}$	$3.119 \times 10^{+02}$	$7.966 \times 10^{+01}$	$5.228 \times 10^{+01}$	$8.616 \times 10^{+01}$
F_{16}	平均值	$2.932 \times 10^{+03}$	$2.953 \times 10^{+03}$	$6.191 \times 10^{+03}$	$3.158 \times 10^{+03}$	$3.294 \times 10^{+03}$	$3.031 \times 10^{+03}$	$2.946 \times 10^{+03}$
	标准差	$2.863 \times 10^{+01}$	$3.274 \times 10^{+01}$	$6.516 \times 10^{+02}$	$1.596 \times 10^{+02}$	$3.206 \times 10^{+02}$	$2.857 \times 10^{+01}$	$2.536 \times 10^{+01}$
F_{17}	平均值	$3.534 \times 10^{+03}$	$3.511 \times 10^{+03}$	$3.543 \times 10^{+03}$	$3.681 \times 10^{+03}$	$3.624 \times 10^{+03}$	$3.516 \times 10^{+03}$	$3.502 \times 10^{+03}$
	标准差	$4.103 \times 10^{+01}$	1.447×10^{-04}	1.463×10^{-04}	$4.541 \times 10^{+02}$	$1.509 \times 10^{+01}$	1.343×10^{-01}	1.430×10^{-04}
F_{18}	平均值	$3.585 \times 10^{+03}$	$3.524 \times 10^{+03}$	$3.508 \times 10^{+03}$	$3.722 \times 10^{+03}$	$3.763 \times 10^{+03}$	$3.593 \times 10^{+03}$	$3.508 \times 10^{+03}$
	标准差	$2.850 \times 10^{+01}$	1.713×10^{-04}	1.502×10^{-04}	$4.568 \times 10^{+02}$	$2.624 \times 10^{+02}$	$3.362 \times 10^{+00}$	1.502×10^{-04}
Total	最优平均值个数	1	4	3	0	0	3	14
	最优标准差个数	0	5	7	1	0	4	10

复合函数的求解精度与稳定性分析 复合函数需要求解出多个问题的最优交解, 考验算法不仅具有较高全局勘探能力, 还应具备局部极值的逃逸能力进而实现局部的精细搜索, 以获得对多个问题的最优求解. 在 6 个复合函数 $F_{13} \sim F_{18}$ 中, MSFSSA 得到了 5 个函数的最优均值和 3 个函数的最优标准差. 性能稍弱的 ISSA 取得 1 个最优均值和 2 个最优标准差. 而 CSSOA 与 SSA 分别在函数 F_{15} 与 F_{16} 中取得了最优均值. HCOPSO 与

IGOA 仅在一个函数上获得了最优标准差, WOA 表现最弱, 均未获得.

综上所述, MSFSSA 在三类函数上都取得了优异表现, 最优均值个数远远超出其他对比算法, 最优标准差虽没有最优平均值优势巨大, 但也排在所有算法第一位. 在单峰、多峰及复合函数的优异表现表明该算法拥有较强的全局搜索与局部开发的自适应平衡能力, 具有较高寻优精度的同时收敛速度更快.

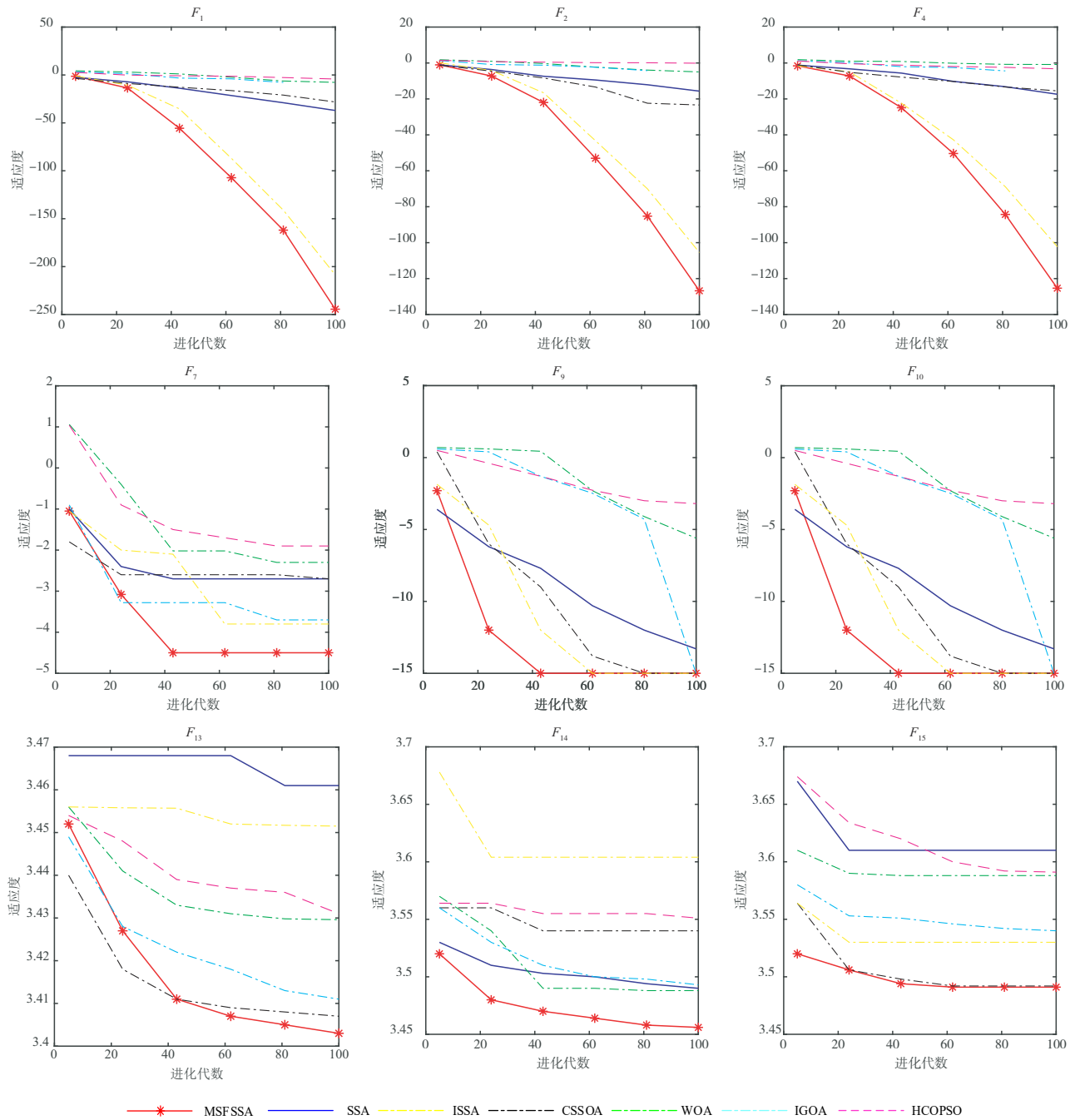


图4 MSFSSA 与对比算法在单峰、多峰、复合函数上的收敛曲线

3.2.2 收敛曲线对比分析

为直观体现算法性能,本文提供了MSFSSA与6种对比算法在部分函数上的收敛曲线,如图4所示.主要包括:单峰函数(F_1, F_2, F_4)、多峰函数(F_7, F_9, F_{10})、复合函数(F_{13}, F_{14}, F_{15}).为了收敛图更加清晰显著,图中纵轴采用对数坐标表示适应度值,横坐标表示迭代次数.

图4可知,无论在单峰、多峰、复合函数上,MSFSSA均具有较快的收敛速度.

进一步分析,在单峰函数 F_1, F_2, F_4 上,MSFSSA的收敛速度远优于对比算法.这是由于算法前期采用最佳适配机制与随机迁移策略,提升种群多样性的同时加大步长因子,增强算法的全局搜索能力进而指导算法后期能够快速找到全局最优解,提升了算法的寻优效率与收敛速度.在多峰函数 F_7, F_9, F_{10} 的收敛曲线图中可知,MSFSSA在整个迭代过程中皆出于深度搜索与广度搜索不断交替执行的寻优过程,尤其在函数 F_7, F_{10} 中的表现最为明显,而在函数 F_9 上则能快速找到全局最优解.此外,在函数 F_7, F_{10} 上MSFSSA在42代均获得最优值,而在函数 F_{10} 上ISSA与CSSOA分别第60代与80代获得最优值,可见MSFSSA具有更快的收敛速度与更强的寻优能力.在复合函数 F_{13}, F_{14}, F_{15} 中,MSFSSA也表现出优异性能.在函数 F_{13}, F_{15} 中,CSSOA在算法前期收敛速度优于MSFSSA,但在算法

后期在函数 F_{13} 上并未找到最优解,而在函数 F_{15} 上虽然与MSFSSA均获得了最优解,但迭代次数为60,而MSFSSA迭代次数为40.在函数 F_{14} 上,MSFSSA相较于其他算法收敛速度则更为显著.这要归功于MSFSSA在算法前期收敛速度较快,在算法中期则相较平缓,这是由于算法通过碰撞反弹算子,实现个体多样性扰动并生成邻域空间,指导种群进行局部开发实现广度搜索提升多样性,当搜索到多个局部最优时进入算法后期,通过模拟预警机制避免陷入局部极值,并利用蝗虫算法实现多路径深度寻优的同时加快收敛速度,获得最优解.

3.2.3 Friedman 检验

本文对7种算法在18个函数上进行Friedman排名,如表4所示.Friedman排名是每个算法在18个测试函数上排名的平均值,排名越小代表算法的综合能力越好,ST代表卡方统计量,如果卡方值大于5.99,则认为各算法存在明显差异.从表3可知,在测试函数 $F_1 \sim F_{18}$ 上,ST值为44.278,远大于5.99,表明7个算法性能存在显著差异.同时MSFSSA在所有测试函数上秩值为1.68,排名第一,且数值远低于其他对比算法.此外,在单峰、多峰与复合函数中,MSFSSA也名列榜首,表明该算法相较于其他6种算法具有更优的寻优效率与收敛速度,综合性能优势显著.

表4 MSFSSA与其他对比算法在均值上的Friedman检验结果

rank	所有函数($F_1 \sim F_{18}$)		单峰函数($F_1 \sim F_6$)		多峰函数($F_7 \sim F_{12}$)		复合函数($F_{13} \sim F_{18}$)	
	算法	秩	算法	秩	算法	秩	算法	秩
1	MSFSSA	1.68	MSFSSA	1.40	MSFSSA	2.25	MSFSSA	1.33
2	CSSOA	2.82	ISSA	2.00	CSSOA	2.42	CSSOA	2.75
3	ISSA	3.35	CSSOA	3.40	ISSA	2.92	IGOA	3.67
4	IGOA	4.15	SSA	4.40	IGOA	4.42	SSA	4.17
5	SSA	4.47	IGOA	4.40	SSA	4.83	ISSA	4.92
6	WOA	5.47	WOA	5.80	WOA	5.33	WOA	5.33
7	HCOPSO	6.06	HCOPSO	6.60	HCOPSO	5.83	HCOPSO	5.83
ST	51.301		22.971		17.106		19.132	

4 结论

针对麻雀种群的不同行为阶段,构建了基于多阶段调度框架的麻雀搜索优化算法.在该框架下,针对种群的不同阶段设计了不同机制与策略.对初始随机产生个体使用Halton序列和Tent映射提高种群的质量与分布均匀性;最佳适配机制调控发现者与加入者比例关系,提高了算法迭代效率;结合Levy飞行与指数分布构造了随机迁移策略,生成潜在的全局最优解区域,扩大了全局搜索范围;预警模拟机制保障了算法的稳定性和持续开发能力,并结合蝗虫算法实现多路径寻优与精细搜索.不同策略与机制间的交替运行、

协同调度使MSFSSA算法在单峰、多峰与复合函数上的求解精度、收敛速度及稳定性上反映出了良好的综合性能,实验结果验证了MSFSSA具有较强的寻优效率与收敛速度.

参考文献

- [1] XUE J K, SHEN B. A novel swarm intelligence optimization approach: Sparrow search algorithm[J]. Systems Science & Control Engineering, 2020, 8(1): 22-34.
- [2] FANG Q C, SHEN B, XUE J K. A new elite opposite sparrow search algorithm-based optimized LightGBM ap-

- proach for fault diagnosis[J]. *Journal of Ambient Intelligence and Humanized Computing*, 2023, 14(8): 10473-10491.
- [3] LI X J, GU J N, SUN X H, et al. Parameter identification of robot manipulators with unknown payloads using an improved chaotic sparrow search algorithm[J]. *Applied Intelligence*, 2022, 52(9): 10341-10351.
- [4] XU P L, LUO W J, XU J F, et al. An alternative way of evolutionary multimodal optimization: Density-based population initialization strategy[J]. *Swarm and Evolutionary Computation*, 2021, 67: 100971.
- [5] ZHANG Z, HAN Y. Discrete sparrow search algorithm for symmetric traveling salesman problem[J]. *Applied Soft Computing*, 2022, 118: 108469.
- [6] 李大社, 孙元威, 阮俊虎. 基于 GOSSA 和 HMM 的时间序列预测算法[J]. *电子学报*, 2023, 51(9): 2492-2503.
- LI D S, SUN Y W, RUAN J H. Time series prediction algorithm based on GOSSA and HMM[J]. *Acta Electronica Sinica*, 2023, 51(9): 2492-2503. (in Chinese)
- [7] 黄鹤, 李文龙, 杨澜, 等. 跳跃跟踪 SSA 交叉迭代 AP 聚类算法[J]. *电子学报*, 2024, 52(3): 977-990.
- HUANG H, LI W L, YANG L, et al. Jump tracking ssa hybrid iterative ap clustering algorithm[J]. *Acta Electronica Sinica*, 2024, 52(3): 977-990. (in Chinese)
- [8] MENG K, CHEN C, XIN B. MSSSA: A multi-strategy enhanced sparrow search algorithm for global optimization[J]. *Frontiers of Information Technology & Electronic Engineering*, 2022, 23(12): 1828-1847.
- [9] 殷礼胜, 刘攀, 孙双晨, 等. 基于互补集合经验模态分解和改进麻雀搜索算法优化双向门控循环单元的交通流组合预测模型[J]. *电子与信息学报*, 2023, 45(12): 4499-4508.
- YIN L S, LIU P, SUN S C, et al. Traffic flow combined prediction model based on complementary ensemble empirical mode decomposition and bidirectional gated recurrent unit optimized by improved sparrow search algorithm[J]. *Journal of Electronics & Information Technology*, 2023, 45(12): 4499-4508. (in Chinese)
- [10] 韩统, 汤安迪, 周欢, 等. 基于 LASSA 算法的多无人机协同航迹规划方法[J]. *系统工程与电子技术*, 2022, 44(1): 233-241.
- HAN T, TANG A D, ZHOU H, et al. Multiple UAV cooperative path planning based on LASSA method[J]. *Systems Engineering and Electronics*, 2022, 44(1): 233-241. (in Chinese)
- [11] CHEN Y F, LI J, ZHANG L. Learning sparrow algorithm with non-uniform search for global optimization[J]. *International Journal of Swarm Intelligence Research*, 2023, 14(1): 1-31.
- [12] ZHANG J N, XIA K W, HE Z P, et al. Semi-supervised ensemble classifier with improved sparrow search algorithm and its application in pulmonary nodule detection[J]. *Mathematical Problems in Engineering*, 2021, 2021: 6622935.
- [13] XIAN S D, LEI H L, CHEN K Y, et al. A novel fuzzy time series model based on improved sparrow search algorithm and CEEMDAN[J]. *Applied Intelligence*, 2023, 53(9): 11300-11327.
- [14] 付华, 刘昊. 多策略融合的改进麻雀搜索算法及其应用[J]. *控制与决策*, 2022, 37(1): 87-96.
- FU H, LIU H. Improved sparrow search algorithm with multi-strategy integration and its application[J]. *Control and Decision*, 2022, 37(1): 87-96. (in Chinese)
- [15] 刘成汉, 何庆. 融合多策略的黄金正弦黑猩猩优化算法[J]. *自动化学报*, 2023, 49(11): 2360-2373.
- LIU C H, HE Q. Golden sine chimp optimization algorithm integrating multiple strategies[J]. *Acta Automatica Sinica*, 2023, 49(11): 2360-2373. (in Chinese)
- [16] 张娜, 赵泽丹, 包晓安, 等. 基于改进的 Tent 混沌万有引力搜索算法[J]. *控制与决策*, 2020, 35(4): 893-900.
- ZHANG N, ZHAO Z D, BAO X A, et al. Gravitational search algorithm based on improved Tent chaos[J]. *Control and Decision*, 2020, 35(4): 893-900. (in Chinese)
- [17] 王毅, 王佩琦, 张茂省, 等. 基于直觉模糊 Memetic 框架的双粒子群混合优化算法[J]. *电子学报*, 2021, 49(6): 1041-1049.
- WANG Y, WANG K Q, ZHANG M S, et al. Hybrid double particle swarm optimization algorithm based on intuitionistic fuzzy memetic framework[J]. *Acta Electronica Sinica*, 2021, 49(6): 1041-1049. (in Chinese)
- [18] 尹德鑫, 张达敏, 蔡朋宸, 等. 改进的麻雀搜索优化算法及其应用[J]. *计算机工程与科学*, 2022, 44(10): 1844-1851.
- YIN D X, ZHANG D M, CAI P C, et al. An improved sparrow search optimization algorithm and its application[J]. *Computer Engineering & Science*, 2022, 44(10): 1844-1851. (in Chinese)
- [19] REYES J, GÓMEZ-DÉNIZ E, GÓMEZ H W, et al. A bimodal extension of the exponential distribution with applications in risk theory[J]. *Symmetry*, 2021, 13(4): 679.
- [20] SAREMI S, MIRJALILI S, LEWIS A. Grasshopper optimisation algorithm: Theory and application[J]. *Advances*

in Engineering Software, 2017, 105: 30-47.

- [21] 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, ZAHNG J, et al. Chaos sparrow search optimization algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2021, 47(8): 1712-1720. (in Chinese)
- [22] 孙辉, 邓志诚, 赵嘉, 等. 混合均值中心反向学习粒子群优化算法[J]. 电子学报, 2019, 47(9): 1809-1818.
SUN H, DENG Z C, ZHAO J, et al. Hybrid mean center opposition-based learning particle swarm optimization[J]. Acta Electronica Sinica, 2019, 47(9): 1809-1818. (in Chinese)
- [23] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [24] 刘亮, 何庆. 基于改进蝗虫优化算法的特征选择方法[J]. 南京大学学报(自然科学), 2020, 56(1): 41-50.
LIU L, HE Q. An feature selection method based on improved grasshopper optimization algorithm[J]. Journal of Nanjing University (Natural Science), 2020, 56(1): 41-50. (in Chinese)



洪国栋 男, 1998 年出生, 河南濮阳人. 西北大学信息科学与技术学院硕士研究生, 主要研究方向为群体智能优化算法.



闫小婕 女, 1999 年 5 月出生, 陕西宝鸡人. 西北大学信息科学与技术学院硕士研究生, 主要研究方向为群体智能优化算法.
E-mail: xiaojie_yxj@163.com

作者简介



王 毅 男, 1979 年 2 月生, 上海人, 博士(后). 现为西北大学信息科学与技术学院副教授, 主要从事智能信息处理、深度学习与群体智能优化算法.
E-mail: wangyi@nwu.edu.cn



郑宏志 男, 1997 年 8 月生, 山东济南人, 西北大学信息科学与技术学院硕士研究生, 主要从事深度学习, 群体智能与多目标优化.
E-mail: hzzheng27189@163.com



黄 欣 女, 1979 年 6 月生, 陕西人, 博士. 现为西安交通大学第一附属医院副教授, 主要从事血管腔内影像处理与优化研究.
E-mail: hearthx@126.com