

文章编号: 2095-2163(2022)05-0037-06

中图分类号: TP301

文献标志码: A

融合鲸鱼泡泡网机制的 SSA 算法

冉琳, 潘大志, 冯世强

(西华师范大学 数学与信息学院, 四川 南充 637009)

摘要: 针对基本麻雀搜索算法(Sparrow Search Algorithm, SSA)在迭代寻优过程中,种群多样性减少、易陷入局部极值等问题,提出一种改进的麻雀搜索算法(WSSA)。首先,基于 Tent 混沌映射策略初始化种群,提高初始解的质量;其次,结合鲸鱼优化算法(WOA)中的鲸鱼泡泡网捕食数学模型,改进发现者的位置更新策略,增强算法全局搜索;然后,采用自适应 t 分布变异,提升算法跳出局部最优能力;最后,在仿真实验中,结果表明改进麻雀搜索算法具有更好的收敛性和求解精度。

关键词: 麻雀搜索算法; Tent 混沌映射; 鲸鱼优化算法; 泡泡网捕食; 自适应 t 分布变异; 函数优化

SSA algorithm with whale bubble net mechanism

RAN Lin, PAN Dazhi, FENG Shiqiang

(College of Mathematics and Information, China West Normal University, Nanchong Sichuan 637009, China)

[Abstract] Aiming at the problems that the basic Sparrow Search Algorithm (SSA) is in the iterative optimization process, the population diversity is reduced, and it is easy to fall into local extremum. An improved Sparrow Search Algorithm (WSSA) is proposed. Firstly, the population is initialized based on the Tent chaotic mapping strategy to improve the quality of the initial solution; secondly, combined with the whale bubble net predation mathematical model in the whale optimization algorithm (WOA), the position update strategy of the discoverer is improved, and the global search of the algorithm is enhanced; then, The adaptive t distribution variation is used to improve the algorithm's ability to jump out of the local optimum. Finally, in the simulation experiment, the results show that the improved sparrow search algorithm has better convergence and solution accuracy.

[Key words] Sparrow Search Algorithm; Tent chaotic map; whale optimization algorithm; bubble net hunting; adaptive t distribution variation; function optimization

0 引言

受大自然的启发,群智能优化算法(Intelligence Optimization Algorithm, IOA)一般是通过对自然界中各种生物的进化过程与物理现象进行研究而形成的一种迭代寻优搜索算法。近年来,涌现了一些新的群智能优化算法,比如:蛾群算法(moth swarm algorithm, MSA)^[1]、哈里斯鹰优化算法(harris hawks optimization, HHO)^[2]、布谷鸟搜索算法(cuckoo search algorithm, CSA)^[3]、萤火虫算法(firefly algorithm, FA)^[4]、蝙蝠算法(bat algorithm, BA)^[5]等。麻雀搜索算法是受自然界中麻雀搜索食物与反捕食行为的启发,于2019年由薛建凯^[6]提出。为使麻雀搜索算法寻优效果更好,学者们主要通过2个方面对其进行改进。一方面,根据麻雀搜索算法本身的特性进行改进;另一方面,通过将2种或多种智能算法与麻雀搜索算法进行融合,从而达到改进

SSA算法的目的。目前,已有许多学者对麻雀搜索算法展开了进一步的研究:王海瑞等人^[7]将Levy飞行策略和柯西高斯变异引入麻雀搜索算法,增加算法搜索方向的多样性,同时也加强了局部搜索能力;吕鑫等人^[8]将高斯变异方法引入SSA算法,提高了搜索精度以及加强了算法跳出局部的能力;柳长安等人^[9]利用萤火虫机制,对最优解进行扰动变异,从而增加种群多样性;汤安迪等人^[10]通过引入正余弦算法和线性递减策略,协调局部和全局搜索能力;李敦桥^[11]将模拟退火算法的Metropolis准则融入SSA算法中,使算法跳出局部最优。针对麻雀搜索算法迭代过程中,种群多样性减少,以及容易陷入局部最优解等缺点,本文根据Tent混沌映射初始化麻雀种群,增加种群多样性,融合鲸鱼优化算法中的鲸鱼泡泡网捕食数学模型,在搜索过程中,更新麻雀种群中发现者的位置,最后按自适应 t 分布变异方式,加强了算法跳出局部的能力。

基金项目: 国家自然科学基金(11871059); 四川省教育厅自然科学基金(18ZA0469)。

作者简介: 冉琳(1996-),女,硕士研究生,主要研究方向:智能算法;潘大志(1974-),男,博士,教授,主要研究方向:智能计算、算法设计;冯世强(1980-),男,硕士,副教授,主要研究方向:数学模型、优化理论及应用。

通讯作者: 潘大志 Email: pdzzj@126.com

收稿日期: 2021-12-01

1 麻雀搜索算法

在麻雀种群中,麻雀分为发现者、加入者和警戒者三种类型。其中,发现者引导种群觅食,加入者通过发现者得到食物,发现者和加入者都有可能是警戒者。

在麻雀搜索算法(SSA)算法中,麻雀个体的适应度值决定能源高低,发现者具有较多能源,引领加入者搜寻食物。发现者有如下位置更新:

$$X_i^{m+1} = \begin{cases} X_i^m \cdot \exp\left(\frac{-i}{\alpha \cdot T_{\max}}\right) & \text{if } R_2 < ST \\ X_i^m + Q \cdot L & \text{if } R_2 \geq ST \end{cases} \quad (1)$$

其中, X_i^m 表示在第 m 次迭代时,第 i 只麻雀的位置信息; $X_i = [X_{i,1}, \dots, X_{i,j}, \dots, X_{i,d}]$, d 是变量的维数; T_{\max} 是最大迭代次数; $\alpha \in (0,1)$ 是随机数; $R_2 \in [0,1]$ 是预警值; $ST \in [0.5,1]$ 是安全值; Q 是一个服从正态分布的随机数, L 是 $1 \times d$ 且元素全为 1 的矩阵。当 $R_2 < ST$ 时,表示觅食环境较安全; $R_2 \geq ST$ 时,说明有部分麻雀已经发现了捕食者,并发出警报信号,种群快速飞向其他安全区域。

发现者与加入者的身份可以根据拥有能源的多少而改变,但每种类型麻雀的占比不变。加入者的位置更新公式为:

$$X_i^{m+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^m - X_i^m}{i^2}\right) & \text{if } i > \frac{n}{2} \\ X_p^{m+1} + |X_i^m - X_p^{m+1}| \cdot A^+ \cdot L & \text{if } i \leq \frac{n}{2} \end{cases} \quad (2)$$

其中, X_{worst}^m 表示目前全局最差位置; X_p^{m+1} 表示第 $m+1$ 次迭代发现者的最优位置; A 是一个元素为 1 或 -1 的 $1 \times d$ 的矩阵, $A^+ = A^T (AA^T)^{-1}$; n 为麻雀总数,当 $i > \frac{n}{2}$ 时,表示第 i 个加入者未得到食物,处于非常饥饿状态,需前往其他区域觅食。

警戒者由整个种群随机产生 10% 到 20%, 位置更新的数学表达式:

$$X_i^{m+1} = \begin{cases} X_{\text{best}}^m + \beta \cdot |X_i^m - X_{\text{best}}^m| & f_i > f_g \\ X_i^m + k \cdot \frac{|X_i^m - X_{\text{worst}}^m|}{(f_i - f_w) + \varepsilon} & f_i = f_g \end{cases} \quad (3)$$

其中, X_{best}^m 表示目前全局最优位置; β 是服从 $(0,1)$ 正态分布的步长控制参数; $k \in [-1,1]$ 是随机数; ε 是防止当 $f_i = f_g$ 时,分母出现为 0 的情况,将其设置为 10^{-8} ; f_i 表示个体适应度值,而 f_g 、 f_w 则分别表示最佳适应度值和最差适应度值。

2 改进麻雀算法

2.1 Tent 混沌映射

基本 SSA 算法利用随机初始化种群,这种方法会导致迭代过程中种群多样性减少。由于混沌映射具有全局遍历性、容易实现和随机性等特点,故本文采用基于 Tent 混沌映射初始化种群,以提高种群多样性和麻雀搜索的遍历性。Tent 映射数学公式如下:

$$X_i^{m+1} = \begin{cases} X_i^m & 0 < X_i^m < a \\ \frac{X_i^m}{a} & \\ \frac{1 - X_i^m}{1 - a} & a \leq X_i^m \leq 1 \end{cases} \quad (4)$$

其中, $a \in (0,1)$, $X_i^m \in (0,1)$ 。

2.2 基于鲸鱼泡泡网捕食机制更新麻雀发现者位置

鲸鱼优化算法(Whale Optimization Algorithm, WOA)是近几年来群智能优化算法中出现的一种新算法,本文结合 WOA 算法中泡泡网捕食中的收缩包围机制和螺旋式位置更新机制,用以更新麻雀发现者的位置,以提高 SSA 算法的全局搜索能力。

融合后的麻雀发现者的更新位置公式为:

$$X_i^{m+1} = \begin{cases} X_{\text{best}}^m - A \cdot |C \cdot X_{\text{best}}^m - X_i^m| & R_2 < ST \\ X_{\text{best}}^m + D \cdot e^{kl} \cdot \cos(2\pi l) & R_2 \geq ST \end{cases} \quad (5)$$

其中, $D = |X_{\text{best}}^m - X_i^m|$, $l \in [-1,1]$; k 表示对数螺旋形状的常数; $A = 2e \cdot r_1 - e$; $C = 2 \cdot r_2$; $e = 2 - 2m/T_{\max}$; r_1, r_2 是 $[0,1]$ 的随机数。

2.3 自适应 t 分布变异

t 分布的曲线形态与自由度参数 n 的大小有关。当 $n = 1$ 时, t 分布是标准的柯西分布;当 $n \rightarrow \infty$ 时, t 分布是标准的高斯分布。 t 分布的概率密度函数为:

$$P_t(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi} \Gamma\left(\frac{n}{2}\right)} \cdot \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}} \quad (6)$$

在算法的初始阶段, t 分布近似于柯西分布变异,此时 t 分布的全局搜索能力较强;在算法运行中期, t 分布变异逐渐由柯西分布变异向高斯分布变异过渡,提高了算法平衡全局搜索与局部开发的能力;在算法运行的后期阶段,由于 $n \rightarrow \infty$, t 分布近似于高斯分布,此时 t 分布的局部开发能力较强。

本文对麻雀第 i 个个体原来位置 X_i^m 变异如下:

$$X_i^{m+1} = X_i^m + X_i^m \cdot t(m) \quad (7)$$

其中, X_i^{m+1} 表示 t 分布变异后的麻雀个体位置, $t(m)$ 表示基于迭代次数 m 为参数的 t 分布。

2.4 改进麻雀算法

针对麻雀优化算法的不足,首先利用 Tent 混沌映射初始化种群,提高初始解的质量;在改进麻雀算法(WSSA)中,为提高寻优效果,利用鲸鱼泡泡网机制扰动发现者位置更新,使得改进后算法更有利于平衡局部开发和全局搜索;最后,利用自适应 t 分布变异更新位置,提高算法跳出局部最优的能力。具体实现步骤如下:

Step 1 设置参数,预警值 ST 、发现者比例 PD 、意识到危险麻雀的比重 SD 。

Step 2 基于 Tent 混沌映射初始化种群。

Step 3 计算麻雀个体适应度值并排序。

Step 4 利用式(5)更新麻雀发现者位置。

Step 5 利用式(2)对加入者位置进行更新。

Step 6 利用式(3)对警戒者位置进行更新。

Step 7 对种群中的越界个体做越界处理。若新适应度值优于旧值,则更新位置和适应度值,反之保留。

Step 8 利用式(7)对麻雀个体的位置进行 t 分布变异,对越界个体做越界处理,并更新麻雀个体位置、适应度值以及对适应度值排序。

Step 9 判断当前迭代次数 m 是否达到 T_{max} ,

若已达到,则循环结束并输出结果,反之返回 Step 3。

3 实验与仿真

3.1 标准测试函数

表 1 分别给出了标准测试函数的公式、维数以及搜索区间。这些标准测试函数被用于测试算法的可行性、稳定性。

3.2 参数设置

在本文中,将各个算法的最大迭代次数设置为 1 000,种群规模设定为 100,即 $n = 100$ 。在灰狼优化算法(Grey Wolf Optimizer, GWO)中, a 是从 2 线性递减至 0, $r_1 \in [0, 1], r_2 \in [0, 1]$; 将引力搜索算法(Gravitation Search Algorithm, GSA)的参数设置为 $G_0 = 100, \alpha = 20$; 将麻雀搜索算法(Sparrow Search Algorithm, SSA)以及本文的改进麻雀算法(WSSA)的参数设置为:发现者 PD 占麻雀种群的 0.2,警戒者 SD 占 0.1,安全阈值 $ST = 0.8$ 。避免程序运行一次出现误差,本文在实验仿真阶段运行 30 次来获得标准测试函数的最优值、均值以及标准差,以反映算法的收敛性、精确度和稳定性。

实验环境:Windows 10, CPU 为 Intel(R) Core(TM) i5-10210U CPU @ 1.60 GHz 2.11 GHz,内存 16 G, Matlab 2018a。

表 1 标准测试函数

Tab. 1 Standard test functions

函数	维数	搜索范围	最优值
$f_1 = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
$f_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0
$f_3 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100, 100]$	0
$f_4 = \max_i \{ x_i , 1 \leq i \leq n\}$	30	$[-100, 100]$	0
$f_5 = \sum_{i=1}^n i x_i^4 + \text{random}[0, 1]$	30	$[-1.28, 1.28]$	0
$f_6 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
$f_7 = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	$[-32, 32]$	0
$f_8 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]$	0

3.3 与其他算法对比

因为单峰测试函数有且仅有一个全局最优解,并没有局部最优解,故用单峰测试函数检验算法的

局部开发能力。与单峰测试函数不同的是,多峰测试函数具有多个局部极值点,搜索全局最优值的难度较大,但是能够测试出算法的搜索能力。表 2 给

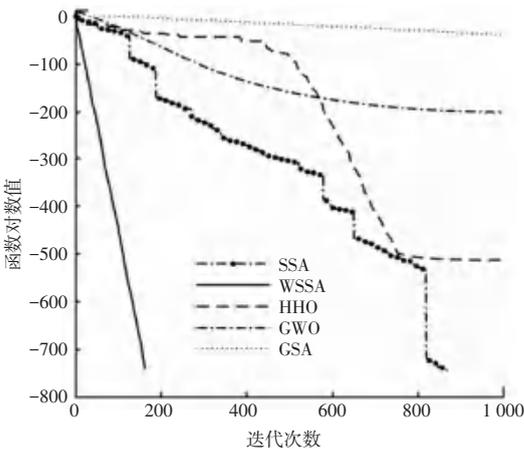
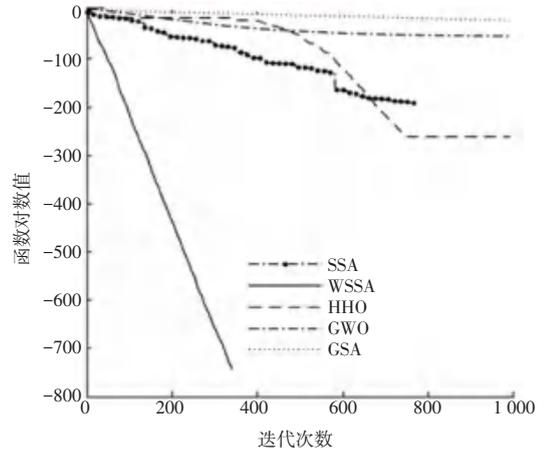
出了引力搜索算法(GSA)、灰狼优化算法(GWO)、哈里斯鹰算法(HHO)、麻雀搜索算法(SSA)以及本文的改进麻雀搜索算法(WSSA),在独立运行了30

次后的实验结果。图1~图8则分别给出了在测试函数上5种算法的收敛曲线。

表2 测试函数优化结果

Tab. 2 Test function optimization results

函数	指标	GSA	GWO	HHO	SSA	WSSA
f_1	Best	2.115 0E-18	3.066 5E-87	2.336 7E-225	0	0
	Average	4.049 2E-18	5.984 4E-85	2.782 6E-202	6.707 5E-169	0
	Std	8.225 1E-19	1.800 6E-84	0	0	0
f_2	Best	6.670 0E-9	2.292 1E-50	2.442 8E-119	0	0
	Average	9.778 3E-9	3.376 8E-49	2.740 5E-107	8.492 8E-119	0
	Std	1.480 1E-9	4.589 5E-49	1.426 6E-106	4.465 7E-118	0
f_3	Best	47.496 3	3.255 2E-32	1.563 0E-209	0	0
	Average	114.929 0	2.362 2E-26	4.233 7E-184	1.201 7E-115	0
	Std	43.940 1	6.989 8E-26	0	6.582 2E-115	0
f_4	Best	6.718 0E-10	2.426 3E-23	7.146 6E-112	0	0
	Average	1.008 6E-9	1.188 9E-21	5.177 4E-102	1.325 9E-63	0
	Std	1.671 0E-10	2.840 3E-21	1.938 2E-101	7.243 8E-63	0
f_5	Best	0.001 9	6.496 2E-5	1.066 3E-6	2.930 5E-6	7.229 4E-7
	Average	0.005 7	2.508 4E-4	2.029 2E-5	8.140 5E-5	1.909 9E-5
	Std	0.002 3	1.257 2E-4	2.762 3E-5	7.594 1E-5	1.768 4E-5
f_6	Best	2.984 9	0	0	0	0
	Average	7.230 0	3.789 6E-15	0	0	0
	Std	2.491 6	1.442 2E-14	0	0	0
f_7	Best	1.210 1E-9	7.993 6E-15	8.881 8E-16	8.881 6E-16	8.881 8E-16
	Average	1.556 0E-9	1.107 3E-14	8.881 8E-16	8.881 6E-16	8.881 8E-16
	Std	1.641 2E-10	3.195 9E-15	0	0	0
f_8	Best	1.064 2	0	0	0	0
	Average	1.831 7	4.279 6E-4	0	0	0
	Std	0.540 8	0.002 3	0	0	0

图1 函数 f_1 的对比曲线Fig. 1 Comparison curve of function f_1 图2 函数 f_2 的对比曲线Fig. 2 Comparison curve of function f_2

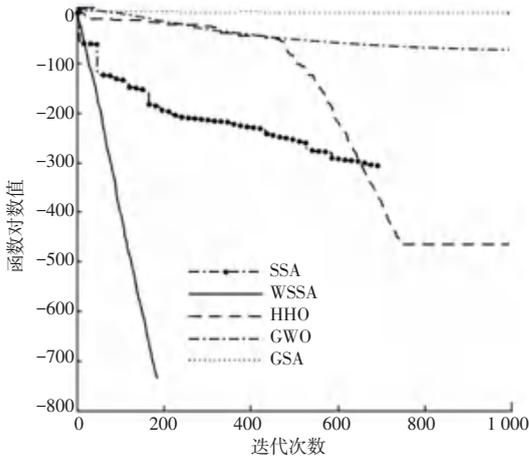


图3 函数 f_3 的对比曲线

Fig. 3 Comparison curve of function f_3

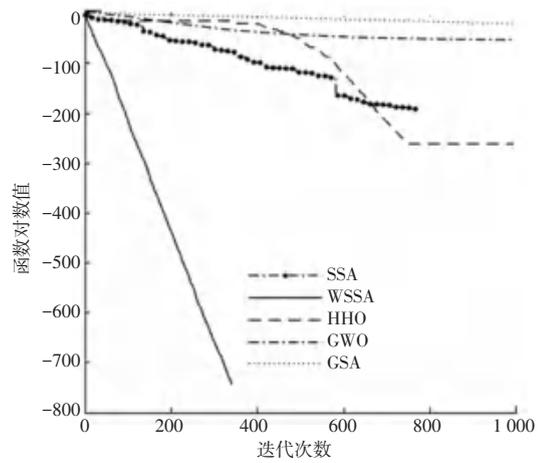


图4 函数 f_4 的对比曲线

Fig. 4 Comparison curve of function f_4

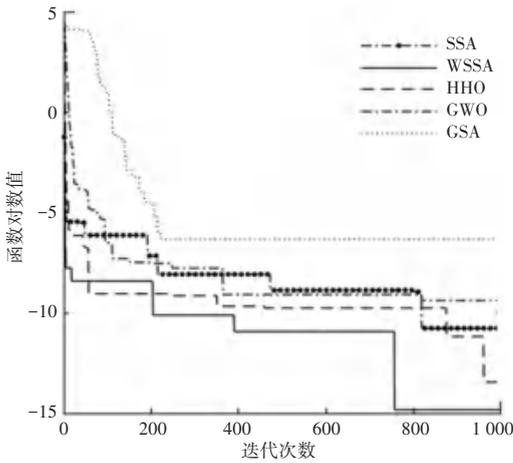


图5 函数 f_5 的对比曲线

Fig. 5 Comparison curve of function f_5

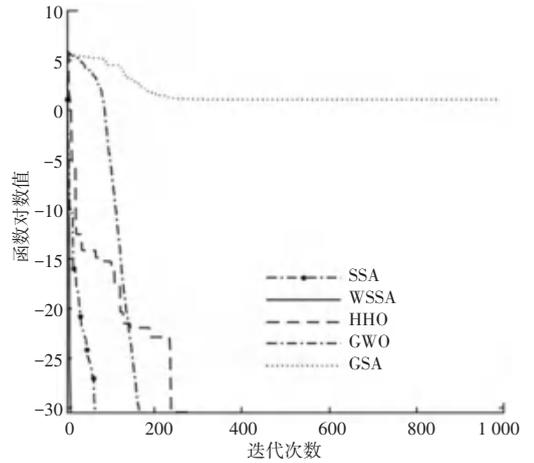


图6 函数 f_6 的对比曲线

Fig. 6 Comparison curve of function f_6

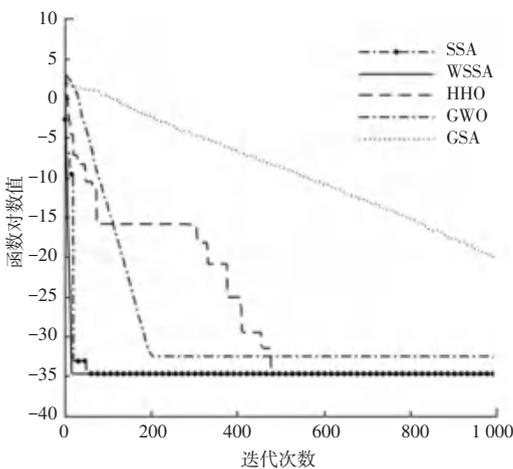


图7 函数 f_7 的对比曲线

Fig. 7 Comparison curve of function f_7

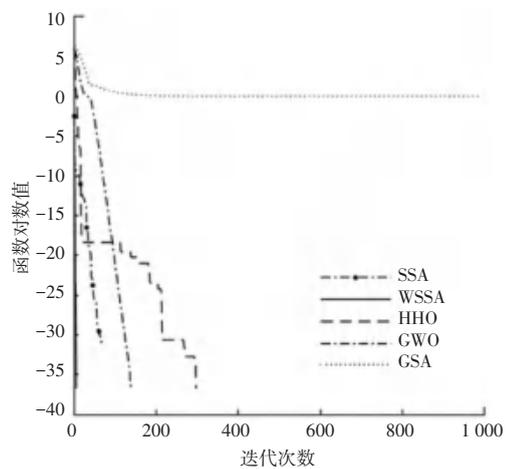


图8 函数 f_8 的对比曲线

Fig. 8 Comparison curve of function f_8

3.4 结果分析

表2中给出了灰狼优化算法(GWO)、引力搜索算法(GSA)、麻雀搜索算法(SSA)、哈里斯鹰算法(HHO)以及本文的改进麻雀搜索算法(WSSA),对于8个标准测试函数在独立运行了30次后的优化结果,*Best*、*Average*、*Std*分别表示运行30次后所得到的最优解、均值以及标准差,最优解在表2中已加粗。

从表2可以得出,引力搜索算法(GSA)的解的质量较差,灰狼优化算法(GWO)对于函数 f_6 、 f_8 可获得最优解,就收敛速度而言,GWO算法对于函数 f_6 、 f_7 、 f_8 ,优于HHO算法和GSA算法,次于SSA算法和WSSA算法。哈里斯鹰算法(HHO)相较于GSA算法、GWO算法的寻优结果更好,对于函数 f_5 ,HHO算法寻优结果比SSA算法的好,说明此时HHO算法的局部开发能力更优,且对于函数 $f_6 \sim f_8$,虽然HHO算法的收敛速度次于SSA算法和WSSA算法,但也能搜索到最优解。SSA算法相比于GSA算法、GWO算法和HHO算法,总体来说,收敛速度更快、寻优结果更好,这说明SSA算法在跳出局部极值点和全局搜索能力方面都要强于GWO算法、GSA算法和HHO算法。WSSA算法对于函数 $f_1 \sim f_8$,成功地找到了全局最优解,且最优解、最优解均值、标准差都比GSA算法、GWO算法、HHO算法以及SSA算法好,表现出了很好的平衡局部与全局、跳出局部极值的能力。

为说明算法的收敛性,图1~图8中给出了5种算法对于标准测试函数在独立运行30次后所得最优结果的收敛性能对比图。在基本麻雀搜索的基础上,因为改变了麻雀发现者的位置更新和加入了自适应 t 分布变异,改进后的算法不仅能够平衡局部和全局搜索,在收敛速度方面也优于其他4种算法。综上所述,WSSA算法具有稳定性较强、收敛速度较快等特点。

4 结束语

本文针对基本SSA算法易陷入局部极值、后期种群多样性减少等缺陷,提出了一种融合鲸鱼优化算法和自适应 t 分布变异的麻雀搜索算法(WSSA)。在SSA算法中,基于Tent混沌映射初始化种群,对发现者的位置融入鲸鱼优化算法的泡泡网捕食机制进行更新,以平衡局部开发与全局搜索能力;加入自适应 t 分布变异,增强算法跳出停滞的能力。但对于发现者、加入者、意识到危险麻雀的位置更新公式可做进一步改进。还可以将算法运用到实际问题中,如背包问题等,这也是接下来要解决的问题。

参考文献

- [1] 崔东文. 蚁群算法与投影寻踪耦合模型在区域水资源脆弱性评价中的应用[J]. 三峡大学学报(自然科学版), 2017, 39(04): 10-18.
- [2] 贾鹤鸣, 康立飞, 孙康健, 等. 哈里斯鹰算法优化脉冲耦合神经网络的图像自动分割[J]. 应用科技, 2019, 46(04): 16-20, 25.
- [3] SHEN Dili. A Cuckoo search algorithm using improved Beta distributing and its application in the process of EDM [J]. Crystals, 2021, 11(8): 916.
- [4] GOLDANLOO M J, GHAREHCHOPOGH F S. A hybrid OBL-based firefly algorithm with symbiotic organisms search algorithm for solving continuous optimization problems [J]. The Journal of Supercomputing, 2022, 78: 3998-4031.
- [5] RAUF H T, ALMADHOR A, ARIF M, et al. Enhanced bat algorithm for COVID-19 short-term forecasting using optimized LSTM [J]. Soft Computing, 2021, 25: 12989-12999.
- [6] 薛建凯. 一种新型的群智能优化技术的研究与应用[D]. 上海: 东华大学, 2019.
- [7] 王海瑞, 鲜于建川. 改进麻雀搜索算法在分布式电源配置中的应用[J]. 计算机工程与应用, 2021, 57(20): 245-252.
- [8] 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(08): 1712-1720.
- [9] 柳长安, 冯雪菱, 孙长浩, 等. 基于改进麻雀算法的最大二维熵分割方法[J/OL]. 激光技术: 1-15 [2021-04-23]. <http://kns.cnki.net/kcms/detail/51.1125.tn.20210423.1018.004.html>.
- [10] 汤安迪, 韩统, 徐登武, 等. 基于混沌麻雀搜索算法的无人机航迹规划方法[J]. 计算机应用, 2021, 41(07): 2128-2136.
- [11] 李敦桥. 一种混合麻雀搜索算法[J]. 电脑知识与技术, 2021, 17(05): 232-234.