

多策略融合的改进麻雀搜索算法研究

王玲玲¹, 孙磊¹, 丁光平², 王加刚³, 段誉¹

¹盐城工学院机械优集学院, 江苏 盐城

²重庆望江工业有限公司, 重庆

³清华大学精密仪器系, 北京

收稿日期: 2022年10月9日; 录用日期: 2022年11月25日; 发布日期: 2022年12月8日

摘要

针对基本麻雀搜索算法(sparrow search algorithm, SSA)存在路径规划时间长、非最优路径且收敛速度慢、容易陷入局部最优解等问题, 提出了一种基于Cubic映射和正余弦搜索策略的SSA。首先, 在麻雀搜索算法的基础上, 引入Cubic混沌映射, 对麻雀种群进行初始化策略, 加速种群间的信息交流, 提高了物种的多样性; 再运用正余弦搜索策略对发现者位置进行更新, 有效地降低了产生局部最优的概率, 加快收敛的速度; 最后通过6个基准测试函数对多策略融合的算法和单一策略的算法进行对比仿真实验。结果表明, 该多策略的改进算法在寻优搜索能力上得到了显著的提高, 迭代次数少, 收敛精度提高且具有较高的实时性和稳定性。

关键词

麻雀搜索算法, Cubic映射, 正余弦搜索策略, 寻优能力

Research on Improved Sparrow Search Algorithm Based on Multi Strategy Fusion

Lingling Wang¹, Le Sun¹, Guangping Ding², Jiagang Wang³, Yu Duan¹

¹Mechanical Excellence College, Yancheng Institute of Technology, Yancheng Jiangsu

²Chongqing Wangjiang Industry Co., Ltd., Chongqing

³Department of Precision Instrument, Tsinghua University, Beijing

Received: Oct. 9th, 2022; accepted: Nov. 25th, 2022; published: Dec. 8th, 2022

Abstract

To address the problems of long path planning time, non-optimal path and slow convergence

文章引用: 王玲玲, 孙磊, 丁光平, 王加刚, 段誉. 多策略融合的改进麻雀搜索算法研究[J]. 软件工程与应用, 2022, 11(6): 1182-1190. DOI: 10.12677/sea.2022.116120

speed, and easy to fall into local optimal solutions in the basic sparrow search algorithm (SSA), a SSA based on Cubic mapping and sine cosine search strategy is proposed. First, based on the sparrow search algorithm, Cubic chaotic mapping is introduced to speed up information exchange between populations and improve species diversity. Initialization strategy for the sparrow population, which accelerates the information exchange between populations and improves the species diversity; Then the sine cosine search strategy is applied to update the discoverer position, which effectively reduces the probability of generating a local optimum and speeds up the convergence; Finally, the simulation experiments are conducted to compare the algorithm of multi-strategy fusion and the algorithm of single strategy by six benchmark test functions. The results show that the multi-strategy improved algorithm has significantly improved the search capability, decreased iterations, improved convergence accuracy and it has higher real-time and stability.

Keywords

Sparrow Search Algorithm, Cubic Mapping, Sine and Cosine Search Strategy, Merit Search Capability

Copyright © 2022 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

近年来, 仿生群体智能优化算法层出不穷, 并广泛应用于各个工业和生产领域, 在解决实际工程中的优化问题时尤为重要, 如粒子群优化(Particle Swarm Optimization, PSO)算法[1]、遗传算法[2]、蚁群优化(Ant Colony Optimization, ACO)算法[3], 布谷鸟搜索算法[4]、萤火虫算法[5]、人工蜂群优化(Artificial Bee Colony, ABC)算法[6]等。麻雀搜索(Sparrow Search Algorithm, SSA)算法[7]是由 Xue 等人 2020 年提出的一种新的群集优化方法, SSA 是根据麻雀觅食和隐蔽天敌的行为来创建优化模型, 该算法具有较好的鲁棒性。但在解决复杂问题时, 和其他群体智能算法一样, 在解决实际工程优化问题时, 初始化种群的随机性分布不能达到绝对均匀, 局部搜索的性能较差, 容易忽略最优解。针对此缺陷, 不同学者对麻雀搜索算法进行了改进[8] [9] [10] [11] [12]。

针对 SSA 算法目前存在的不足, 本文提出了一种基于 Cubic 映射和正余弦搜索策略的麻雀搜索算法。该算法首先通过 Cubic 混沌映射的方法将初始麻雀种群进行优化, 再运用正、余弦搜索策略对发现者位置进行更新, 寻找更优质的可行解, 来提高收敛精度和寻优能力。对多策略融合与单独一种策略的 SSA 进行对比, 仿真结果表明, 多策略融合的算法不论是在寻优能力还是收敛精度和收敛速度上, 都有着显著的优势。

2. 麻雀搜索算法

自然界中, 麻雀种群在觅食和反捕食的过程中有着明确的分工。SSA 算法将其具体分为发现者、追随者和预警者这三类角色。其中, 一般会选种群中每次迭代中位置信息最佳的麻雀担任发现者, 主要为整个麻雀群体搜寻高质量的食物资源, 并指明区域和引导目标。除了发现者外, 剩余个体都是追随者, 数量与发现者比重一致, 跟随发现者来完成觅食, 并不断监视发现者来争抢食物资源来增加自身的捕食效率, 进而成为发现者。预警者作为麻雀群体中比较特殊的部分个体, 一般来说数量占整个麻雀种群的 20% 左右, 若发生危险, 会立即提醒其他麻雀群体进行撤离, 从而规避危险。麻雀搜索算法的步骤如下:

1) 初始化种群位置, 在 D 维解空间内 n 只麻雀种群的位置可表示为

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \quad (1)$$

2) 初始麻雀种群的觅食位置的适应度值可表示为

$$F_x = \begin{bmatrix} f[(x_{1,1} \ x_{1,2} \ \cdots \ x_{1,d})] \\ f[(x_{2,1} \ x_{2,2} \ \cdots \ x_{2,d})] \\ \vdots \\ f[(x_{n,1} \ x_{n,2} \ \cdots \ x_{n,d})] \end{bmatrix} \quad (2)$$

式中, F_x 为麻雀种群内个体位置的适应度值。

3) 麻雀种群中的发现者的位置更新公式为

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot iter_{\max}}\right) & R_2 < ST \\ X_{i,j}^t + Q \cdot L & R_2 \geq ST \end{cases} \quad (3)$$

其中, t 表示当前迭代次数, $j=(1,2,\dots,d)$, $X_{i,j}^t$ 表示在第 t 代中第 i 个麻雀在 j 维的位置信息; $\alpha \in (0,1)$ 范围的一个服从均匀分布的随机数; $iter_{\max}$ 表示最大迭代次数; $R_2 (R_2 \in [0,1])$ 表示预警阈值; $ST (ST \in [0.5,1])$ 表示安全阈值; Q 是一个随机数且服从 $[0,1]$ 正态分布; $L=1 \times d$, 且里面每个元素都为 1。当 $R_2 < ST$ 时, 当前环境安全, 发现者继续觅食; 当 $R_2 \geq ST$ 时, 当前环境危险, 预警者就会开始鸣叫向其他麻雀发出危险信号, 麻雀种群转移飞往安全区。

4) 麻雀种群中的跟随者的位置更新公式为

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst}^t - X_{i,j}^t}{t^2}\right) & i > \frac{n}{2} \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L & i \leq \frac{n}{2} \end{cases} \quad (4)$$

其中: X_{worst}^t 表示发现者在第 t 次迭代在第 j 维的最差位置信息, X_p^{t+1} 表示发现者在第 $t+1$ 次迭代在第 j 维处于最优位置信息。 A 是一个 $1 \times d$ 的一行多维向量, 且内部元素随机选择是 1 或 -1。其中, $A^+ = A^T (AA^T)^{-1}$ 。当 $i > n/2$ 时, 表示当前位置的追随者处于饥饿状态, 适应度值比较差, 需要跳出当前位置去其他位置觅食; 当 $i \leq n/2$ 时, 表示当前位置的追随者有较好的适应度值, 可以在该区域范围内继续觅食。

5) 麻雀种群中的预警者的位置更新公式为

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^{t+1}| & f_i > f_g \\ X_{i,j}^t + K \cdot \frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon} & f_i = f_g \end{cases} \quad (5)$$

其中: X_{best}^t 表示全局最优的位置信息; β 为控制步长的参数, 服从正态分布(均值为 0、方差为 1); $K \in [-1,1]$ 范围内的一个均匀随机数, 用来控制麻雀的运动方向; f_i 是当前个体麻雀的适应度值, f_g 和 f_w

分别为当前种群的局部最优和最差适应度； ε 为一个极小的常数，用来防止发生分母为 0 的情况，这里本文取 $10E-8$ 。当 $f_i > f_g$ 时，表示麻雀觅食区域处于较差的位置，很容易受到攻击；当 $f_i = f_g$ 时，表明部分麻雀察觉到了危险，需要靠近其他同伴以减少被抓的风险。

3. 算法分析与改进

3.1. Cubic 混沌映射优化初始种群

Cubic 混沌映射是混沌映射中比较常用的方法之一。在解决优化问题时，混沌映射可以生成 0 到 1 之间的随机性序列，且具有较好的遍历性和随机性。

原 Cubic 映射函数可以表示如下：

$$x_{n+1} = bx^3 - cx_n \quad (6)$$

式中， b 和 c 表示混沌影响因子。一般在 $c \in (2.3, 3)$ 时，Cubic 映射可生成混沌序列。当 $b = 1$ 时，Cubic 映射序列值 $x_n \in (-2, 2)$ ；当 $b = 4$ 时，序列值 $x_n \in (-1, 1)$ 。

Cubic 映射表达式也可以表示为：

$$x_{n+1} = \rho x_n (1 - x_n^2) \quad (7)$$

式中， ρ 为 Cubic 映射因子即控制参数，序列值 $x_n \in (0, 1)$ 。

实验结果表明，当初值 $x_0 = 0.3$ ， $\rho = 2.595$ 时生成的混沌序列会有良好的随机性，处于完全混沌的状态。设置迭代次数均为 1000，其仿真结果如下图 1 所示。

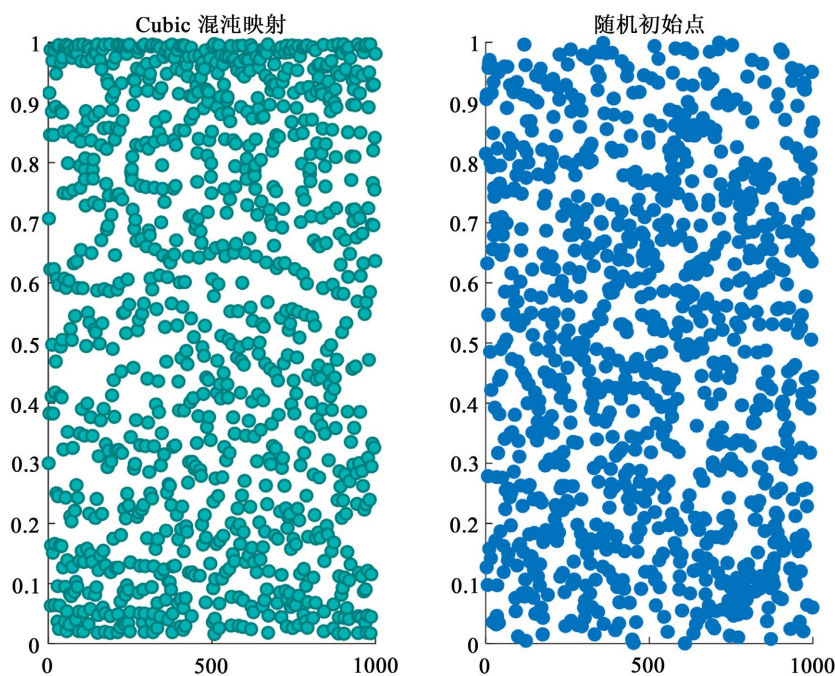


Figure 1. Cubic chaotic sequence simulation distribution

图 1. Cubic 混沌序列仿真分布图

由图 1 可知，当 $x_0 = 0.3$ ， $\rho = 2.595$ 时，使用 Cubic 映射来初始化种群后，可以使分布更加均匀，扩大了寻优的区域，从而增加了种群的复杂性。

3.2. 正余弦策略更新发现者位置

正、余弦搜索策略的数学表达式为

$$X(t+1) = \begin{cases} X(t) + r_3 \times \sin r_4 \times D & r_5 < 0.5 \\ X(t) + r_3 \times \cos r_4 \times D & r_5 \geq 0.5 \end{cases} \quad (8)$$

$$D = \{r_6 \times X_i(t) - X_i(t) \mid i = 1, 2, \dots, d\} \quad (9)$$

$$r_3 = a - a \times \frac{t}{M} \quad (10)$$

式中： a 为正常数，本文设置为 1； t 为迭代次数； M 表示最大迭代次数；表示控制下一次迭代的方向且呈现线性递减趋势； $r_4 \in (0, 2\pi)$ ，表示下一次迭代的移动距离； r_5 和 r_6 表示两个在 $[0, 1]$ 上服从均匀分布的随机数， r_5 用来判别是服从正弦函数还是余弦函数， r_6 用来判别当前最优个体的影响程度即权重系数。正余弦搜索策略的作用是使得算法有效防止发现者落入局部最优值，大大地提高了收敛精度，同时加快了迭代效率。

3.3. CSSSA 算法的流程图

基于 Cubic 混沌映射和正余弦搜索策略改进的麻雀搜索算法步骤流程图见图 2。

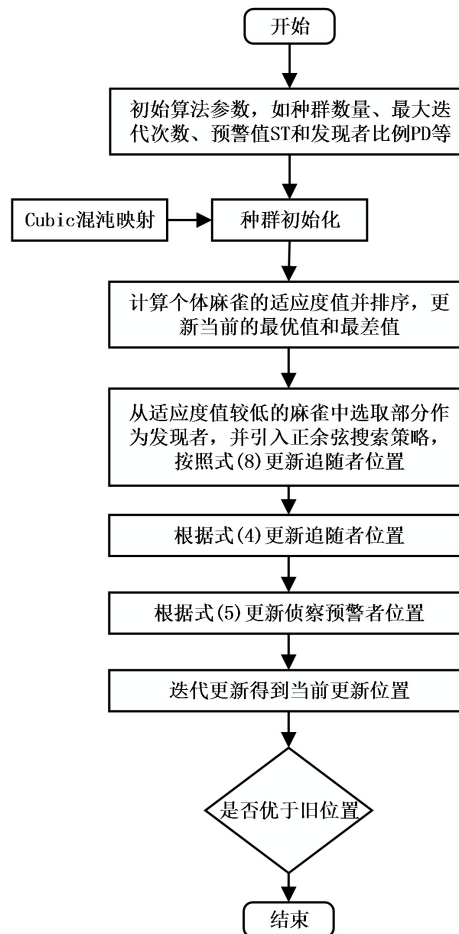


Figure 2. CSSSA flow chart
图 2. CSSSA 流程图

4. 仿真实验与结果分析

4.1. 仿真实验环境

为了保证实验结果的严谨和公平,本次实验的操作系统为 Windows10 操作系统专业版 64 位,实验的集成开发环境为 MATLAB 2018b,计算机硬件条件 Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz,运行内存为 8 G。

4.2. 比较对象和参数设置

为了验证改进算法的有效性,本文选取 7 个基准测试函数来验证改进算法与 SSA 算法、GWO 算法、PSO 算法来对比,前 4 个为单峰函数,后 3 个为多峰函数。各个基准测试函数信息如表 1 所示。所有算法的种群数量为 30,迭代次数为 1000,空间维度为 30,其它各参数设置如表 2 所示。分别统计各算法的最优值、平均值、方差及运行时间四项指标,用来综合衡量算法的寻优能力。4 项指标用来检验算法的寻优能力、寻优速度及稳定性。算法优化对比实验结果见表 3。

Table 1. Benchmark function test table

表 1. 基准函数测试表

序号	函数	维度	区间	极值
f_1	$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
f_2	$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10, 10]	0
f_3	$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)$	30	[-100, 100]	0
f_4	$F_4(x) = \max \{ x_i , 1 \leq i \leq n \}$	30	[-100, 100]	0
f_5	$F_5(x) = \frac{1}{4000} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
f_6	$F_6(x) = -20 \exp\left[-0.2 \sqrt{\sum_{i=1}^n (x_i)^2}\right] - \exp\left[\sum_{i=1}^n \cos(2\pi x_i)\right] + 20 + e$	30	[-100, 100]	0
f_7	$F_7(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) + 10$	30	[-5.12, 5.12]	0

Table 2. The parameter settings for each algorithm

表 2. 各算法的参数设置

算法	参数设置
SSA	$PD = 0.7$, $SD = 0.2$, $ST = 0.6$
SSSA	$PD = 0.7$, $SD = 0.2$, $ST = 0.6$
CSSSA	$PD = 0.7$, $SD = 0.2$, $ST = 0.6$
GWO	a 从 2 递减到 0
PSO	$C_1 = C_2 = 1$, $\omega = 0.3$

Table 3. Algorithm optimization compares experimental results
表 3. 算法优化对比实验结果

函数	算法	最优值	平均值	标准差	运行时间
F1	SSA	0	0	0	0.730337
	SSSA	0	0	0	2.729055
	CSSSA	0	0	0	0.331297
	GWO	3.25E-61	7.06E-59	1.49E-58	0.730337
	PSO	2.73E-17	6.65E-12	3.16E-11	0.068184
F2	SSA	0	5.64E-290	0	0.708496
	SSSA	0	0	0	2.837528
	CSSSA	0	0	0	0.337507
	GWO	6.67E-36	9.30E-35	9.11E-35	0.708496
	PSO	1.07E-07	0.001926	0.005565	0.068286
F3	SSA	0	6.55E-205	0	1.114254
	SSSA	0	3.92E-317	0	4.552729
	CSSSA	0	0	0	0.536326
	GWO	1.37E-20	6.64E-16	1.47E-15	1.114254
	PSO	0.355402	2.999262	2.196039	0.274216
F4	SSA	0	2.19E-180	0	0.699407
	SSSA	0	1.72E-169	0	2.713909
	CSSSA	0	0	0	0.33663
	GWO	2.05E-16	1.01E-14	1.69E-14	0.699407
	PSO	0.102354	0.450567	0.193799	0.067621
F5	SSA	0	0	0	0.774075
	SSSA	0	0	0	3.015265
	CSSSA	0	0	0	0.364194
	GWO	0	0.001272	0.004877	0.774075
	PSO	0	0.009108	0.010698	0.098076
F6	SSA	8.88E-16	8.88E-16	0	0.741164
	SSSA	8.88E-16	8.88E-16	0	2.890346
	CSSSA	8.88E-16	8.88E-16	0	0.346387
	GWO	20.60683	20.8176	0.102596	0.741164
	PSO	1.501747	18.15033	5.459149	0.100515
F7	SSA	0	0	0	0.741869
	SSSA	0	0	0	2.823418
	CSSSA	0	0	0	0.348516
	GWO	0	0	0	0.741869
	PSO	9.949591	20.09817	4.863061	0.08366

4.3. 仿真结果分析

从表 2 可知, 与其他群体智能算法比较, 本文的采用 Cubic 混沌映射和正余弦策略, 对于单峰测试函数 F1 到 F4 的寻优值, CSSSA 算法的寻优性能均优于 SSA、SSSA、GWO 和 PSO, 且 CSSSA 标准差都为 0, 说明 CSSSA 较其他群体智能算法而言, 增强了算法的稳定性。在求解多峰函数 F5 和 F7 方面, CSSSA、SSSA 和 SSA 的求解精度最高, 并且寻优精度都达到最优值, 且最优值、平均值和标准差都是 0, 但 CSSSA 的运行时间最快。在求解多峰函数 F6 时, CSSSA 的改进效果并不显著。因此在实际应用的时间, 可以根据不同的情况来不同的改进算法。

为了更加直观地比较各个算法的寻优能力, 图 3 为 1000 次迭代次数下各群体智能算法的寻优的仿真结果。

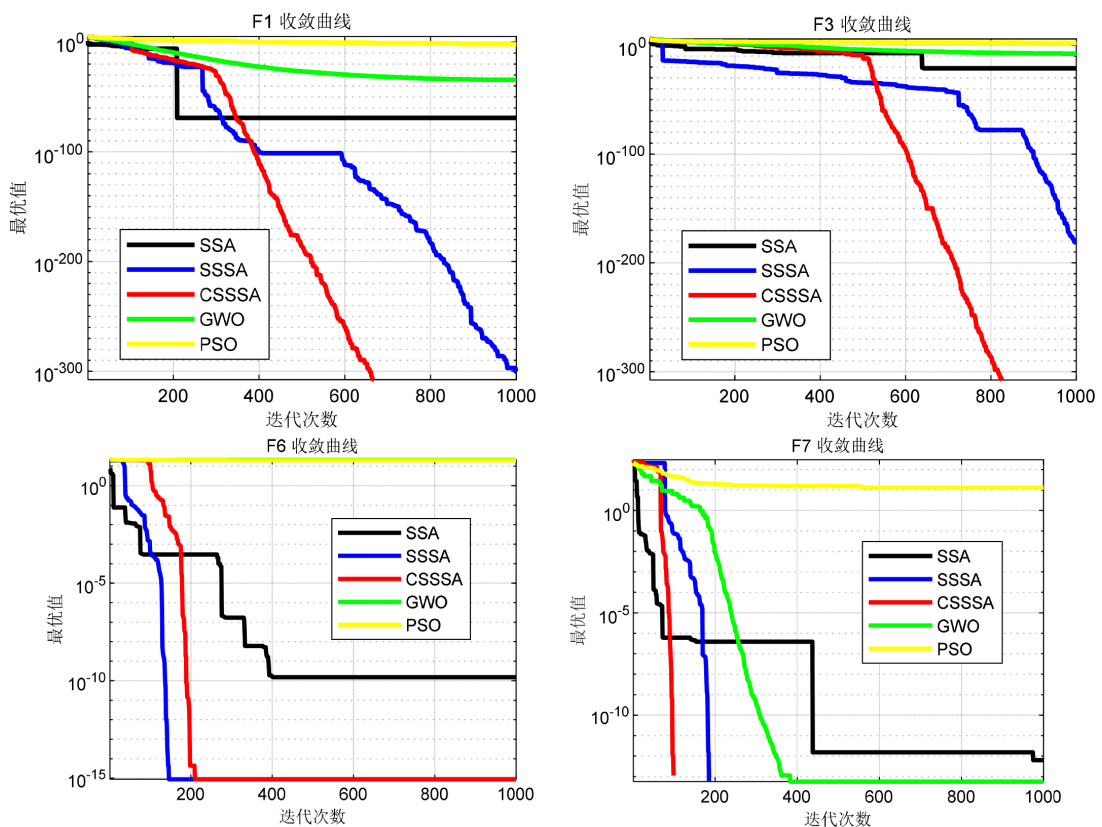


Figure 3. Optimization test results of different algorithms

图 3. 不同算法的寻优测试结果

从上述四种不同的基准测试函数的收敛曲线图中, 前两张为单峰函数 F1 和 F3 的收敛曲线图, 后两张为多峰函数 F6 和 F7 的收敛曲线图。在迭代次数上, CSSSA 有着明显的优势, 可以在最短的迭代中找到最优值, 增强了迭代的效率, 说明了 CSSSA 在改善局部最优解上优明显的增强。尽管在 F6 多峰函数的仿真结果不太理想, 但总体来说, CSSSA 算法在相比其他算法在收敛精度上仍有显著的优势, 这也说明了策略的有效性。

综上所述, Cubic 混沌映射的引入使得麻雀种群在前期的搜索区域变大且分布更加均匀, 而正余弦搜索策略的加入使得算法在寻优精度上得到极大的提升。

5. 结语

本文从混沌初始化、正余弦搜索策略 2 个方面改进了 SSA 算法, 提出了一种融合 Cubic 混沌映射和正余弦搜索策略的 SSA 算法。首先采用 Cubic 混沌映射使初始种群的分布更加均匀且更具多样性, 增大了搜索范围, 然后采用正余弦搜索策略减少了算法的迭代次数提高了搜索的效率, 同时改善算法收敛精度, 增强了算法跳出局部最优值的能力, 最后通过对 7 种基准测试函数进行寻优实验。仿真结果表明, 该改进算法与其他群体智能算法相比, 寻优速度和收敛精度得到大幅度的提升, 且具有较强的稳定性。未来将继续改进该算法, 分析混沌映射麻雀搜索算法解决多约束条件优化问题的能力, 并将其应用到路径规划、神经网络等方向, 提高工程应用价值。

参考文献

- [1] 冯茜, 李擎, 全威, 裴轩墨. 多目标粒子群优化算法研究综述[J]. 工程科学学报, 2021, 43(6): 745-753.
<https://doi.org/10.13374/j.issn2095-9389.2020.10.31.001>
- [2] Tang, K.S. and Man, K.F. (1996) Genetic Algorithms and their applications. *Signal Processing Magazine IEEE*, **13**, 22-37.
- [3] 史春天, 曾艳阳, 侯守明. 群体智能算法在图像分割中的应用综述[J]. 计算机工程与应用, 2021, 57(8): 36-47.
- [4] 郑洪清, 周永权. 一种自适应步长布谷鸟搜索算法[J]. 计算机工程与应用, 2013, 49(10): 68-71.
- [5] Yang, X.S. and He, X. (2013) Firefly Algorithm: Recent Advances and Applications. *International Journal of Swarm Intelligence*, **1**, 36-50.
- [6] Karaboga, D. and Basturk, B. (2007) Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J. and Pedrycz, W., Eds., *Foundations of Fuzzy Logic and Soft Computing*, Springer, Berlin, 789-798.
- [7] Xue, J.K. and Shen, B. (2020) A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. *Systems Science & Control Engineering*, **8**, 22-34.
- [8] 吕鑫, 慕晓冬, 张钧. 基于改进麻雀搜索算法的多阈值图像分割[J]. 系统工程与电子技术, 2021, 43(2): 318-327.
- [9] 汤安迪, 韩统, 徐登武, 谢磊. 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(7): 2128-2136.
- [10] 欧阳城添, 朱东林. 融合 K-means 的多策略改进麻雀搜索算法研究[J]. 电光与控制, 2021, 28(12): 11-16.
- [11] 吕鑫, 慕晓冬, 张钧, 王震. 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.
<https://doi.org/10.13700/j. bh.1001-5965.2020.0298>
- [12] 毛清华, 张强. 融合柯西变异和反向学习的改进麻雀算法[J]. 计算机科学与探索, 2021, 15(6): 1155-1164.