

doi: 10.12068/j.issn.1005-3026.2020.02.003

# 珊瑚礁算法的改进研究

孔 芝, 李事成, 赵 杰  
(东北大学秦皇岛分校 控制工程学院, 河北 秦皇岛 066004)

**摘 要:** 珊瑚礁算法易于陷入局部最优且寻优精度低, 因此提出一种改进的珊瑚礁算法. 此算法借鉴粒子群算法、高斯变异和模拟退火算法的思想改进珊瑚礁算法的内部有性繁殖、无性繁殖和更替机制, 提高了算法的寻优精度并可跳出局部最优. 在仿真实验中, 将改进珊瑚礁算法与基本珊瑚礁算法和粒子群算法等 10 种算法分别在高维和低维测试函数下进行比较. 实验结果表明, 改进的珊瑚礁算法不仅较其他算法具有更好的收敛速度和精度, 而且在高维测试函数中, 仍然可以保持良好的性能.

**关 键 词:** 珊瑚礁算法; 内部有性繁殖; 无性繁殖; 改进粒子群算法; 模拟退火算法

**中图分类号:** TP 18      **文献标志码:** A      **文章编号:** 1005-3026(2020)02-0163-07

## Improved Coral Reef Algorithm

KONG Zhi, LI Shi-cheng, ZHAO Jie  
(School of Control and Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China.  
Corresponding author: KONG Zhi, E-mail: kongz@neuq.edu.cn)

**Abstract:** The coral reef algorithm has the disadvantages of being easy to fall into local optimum and low precision. In view of the shortcomings of coral reef algorithm, this paper proposes an improved coral reef algorithm. By referring to particle swarm optimization, Gaussian mutation and simulated annealing algorithm, this algorithm improves the broadcast spawning, asexual reproduction and setting(replacement) mechanisms of coral reef algorithm, which greatly improves the optimization precision of the algorithm and can jump out of the local optimum. In the simulation experiment, the improved coral reef algorithm is respectively compared with ten algorithms such as basic coral reef algorithm and particle swarm algorithm in high-dimensional and low-dimensional test functions. The experimental results show that the improved coral reef algorithm has better convergence rate and accuracy than other algorithms, which can still be maintained in the high-dimensional test functions.

**Key words:** coral reef algorithm; internal sexual reproduction; asexual reproduction; particle swarm optimization; simulated annealing algorithm

珊瑚礁优化(coral reef optimization, CRO)算法是 Salcedo-Sanz 等<sup>[1]</sup>在 2013 年, 受到珊瑚虫繁衍生存过程的启发而提出的一种智能优化算法, 用于处理多峰值函数优化问题. 2014 年, Pastor-Sanchez 等<sup>[2]</sup>提出一种基于混合珊瑚礁-极限学习机的方法, 并将其应用于风速预测系统中的特征选择, 体现出优异的性能. 2015 年, Salce-Sanz 等<sup>[3]</sup>提出一种具有和声搜索算子的珊瑚礁优化算法, 并将其应用于风速预测中, 得到良好结果. 2016 年, Pastor-Sanchez 等<sup>[4]</sup>将珊瑚礁优化算法应用于多目标优化问题中, 在计算资源有限的情况下表现出优异性能. 同年, Salcedo-Sanz 等<sup>[5]</sup>将珊瑚礁算法应用于微电网最佳电池优化调度问题, 大大节省了总电费. 2017 年, Magdaleno 等<sup>[6]</sup>利用珊瑚礁优化算法对调谐质量阻尼器进行结构振动控制, 产生了极好的效果. 同年, Jimenez-Fernandez 等<sup>[7]</sup>提出一种物种珊瑚礁优化算法, 并将其有效地应用于太阳能全球辐射重要参数缩减中. 2018 年, Duran-Rosal 等<sup>[8]</sup>将珊瑚礁优化算法应用于缩减时间序列最佳尺寸中, 显示出强大的

分割能力.

目前为止,人们已经提出了许多改进的珊瑚礁优化算法<sup>[9-12]</sup>,但这些算法在精度、收敛速度与避免陷入局部最优能力方面仍有待进一步提高. 本文提出一种改进的珊瑚礁(improved coral reef optimization, ICRO)算法,通过对原珊瑚礁优化算法内部有性繁殖、无性繁殖和更替机制的改进,可以有效避免算法陷入局部最优,提高精度,并通过具体的测试函数证明了算法的优越性.

1 珊瑚礁算法

珊瑚礁优化算法是受到珊瑚虫繁衍生存过程的启发而提出的一种元启发式算法. 珊瑚礁优化算法基本执行过程如下<sup>[13]</sup>:

①初始化:假设存在大小为  $M$  的珊瑚礁,即礁上存在  $M$  个节点空间供珊瑚虫生存;开始时,珊瑚礁上已有比例为  $\rho$  的空间被珊瑚虫占据,即礁上珊瑚虫的个数为  $M \times \rho$ .

②外部有性繁殖和内部有性繁殖:珊瑚虫产生子代的方式分为两种,即外部有性繁殖和内部有性繁殖;其比例分别为  $\xi$  和  $1 - \xi$ ,即外部有性繁殖的珊瑚虫个数为  $M \times \rho \times \xi$ ,内部有性繁殖的珊瑚虫个数为  $M \times \rho \times (1 - \xi)$ . 外部有性繁殖的子代由雌雄异体的两个亲代产生,如式(1)所示;内部有性繁殖的子代由雌雄同体产生,如式(2)所示.

$$\left. \begin{aligned} c_{1,\alpha} &= [(1 + \varphi)C_{1,\alpha} + (1 - \varphi)C_{2,\alpha}] / 2, \\ c_{2,\alpha} &= [(1 - \varphi)C_{1,\alpha} + (1 + \varphi)C_{2,\alpha}] / 2, \\ \alpha &= 1, \dots, \psi. \end{aligned} \right\} \quad (1)$$

$$c_\alpha = C_\alpha + \text{rand}(-1, 1) \times (C_\alpha^{\max} - C_\alpha^{\min}). \quad (2)$$

式中:  $C_1$  和  $C_2$  表示两个雌雄异体珊瑚虫,  $c_1$  和  $c_2$  表示其产生的两个子代;  $C$  表示雌雄同体珊瑚虫,  $c$  表示其产生的子代;  $\alpha$  为迭代次数;  $\varphi$  为随机变量,其产生方式如下:

$$\varphi = \begin{cases} (2\tau)^{\frac{1}{\kappa+1}}, & \tau < 0.5; \\ [2(1 - \tau)]^{-\frac{1}{\kappa+1}}, & \tau \geq 0.5. \end{cases} \quad (3)$$

式中:  $\kappa$  为交叉常数;  $\tau$  为 0 到 1 的随机数.

③更替机制:新产生的子代珊瑚虫需要占据珊瑚礁生存,此时珊瑚礁上有未被占据和已被占据的空间,其数量分别为  $M \times (1 - \rho)$  和  $M \times \rho$ . 若子代选择未被占据珊瑚礁,则可直接占据;若子代选择已被占据的珊瑚礁,则需与其上原有珊瑚虫进行适应度  $f(p)$  比较;若子代适应度高则占据该处,否则失败,继续尝试其他空间;若在最大占据

次数  $\mu$  内未完成占据,则子代珊瑚虫死亡.

④无性繁殖:当所有子代珊瑚虫完成珊瑚礁占据后,对现有珊瑚礁上的珊瑚虫进行适应度排序,选出比例为  $\gamma$  的高适应度珊瑚虫,让其通过无性繁殖(复制)的方式产生子代,并进行珊瑚礁的占据.

⑤毁灭机制:再一次进行适应度排序,选出比例为  $\delta$  的低适应度珊瑚虫,将其毁灭.

进入循环,重复步骤②~⑤,直至满足终止条件,即达到最大迭代次数  $\psi$ . 得出最优解.

2 改进的珊瑚礁算法

为了克服珊瑚礁优化算法的不足,本文对珊瑚礁优化算法进行了改进,主要涉及三个方面:内部有性繁殖方式、无性繁殖方式和更替方式.

2.1 内部有性繁殖方式改进

原始珊瑚礁算法的内部有性繁殖方式为遗传算法等经典算法传统意义上的变异方式,其受自身的影响较大,且变优趋势是随机的,不利于收敛. 粒子群优化(PSO)算法是受到飞鸟集群活动规律性的启发,进而利用群体智能建立的一个简化模型,它通过追随当前搜索到的最优个体来寻找全局最优;而改进粒子群优化(IPSO)算法<sup>[14]</sup>是对 PSO 算法的更替方式进行了改进,具有向优收敛效果好、收敛速度快的优点. 因此在 CRO 算法内部有性繁殖部分加入 IPSO 算法的改进以提高其向优收敛效果和收敛速度. 本文受 IPSO 算法的启发,将内部有性繁殖公式(2)改为公式(4)和公式(5).

$$x_{i,j}(t+1) = x_{i,j}(t) + 2r_1[p_{\text{best}}(t) - x_{i,j}(t)], \quad (4)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + 2r_2[g_{\text{best}}(t) - x_{i,j}(t)]. \quad (5)$$

式中:  $x$  为珊瑚虫;  $i$  为珊瑚虫的序列号;  $j$  为维数;  $t$  为迭代次数; 2 为学习因子;  $r$  为分布于  $[0, 1]$  间的随机数;  $p_{\text{best}}$  为个体最优珊瑚虫;  $g_{\text{best}}$  为群体最优珊瑚虫. 式(4)和式(5)由  $r$  和  $\gamma$  来判断,  $\gamma = \sqrt{(1 - t) / T_{\text{all}}}$ ,  $T_{\text{all}}$  为总代数.

改进后的珊瑚礁算法以自身最优珊瑚虫和全局最优珊瑚虫为参考对象,减少了差解的占比,并且设置了与代数相关的选择概率;前期有利于局部搜索,后期有利于全局搜索,不仅保持种群多样性,还提高了向优收敛速度.

2.2 无性繁殖方式改进

原无性繁殖部分是将当前所有珊瑚虫中最好的珊瑚虫按比例  $\gamma$  进行完全复制,产生幼虫. 这样虽然增加了优秀珊瑚虫的比例,但使种群更容易

陷入局部最优,不利于后期进一步搜索和提高精度. 高斯变异是重点搜索原个体附近的某个局部区域,具有提高精度的优点;因此在 CRO 无性繁殖部分加入高斯变异,以提高搜索精度. 本文受高斯变异<sup>[15]</sup>的启发,将无性繁殖方式改进为式(6).

$$x_i = g_{\text{best},i} + g_{\text{best},i} \cdot N(0,1). \tag{6}$$

式中: $i$  为比例为  $\gamma$  的高适应度珊瑚虫的序号; $g_{\text{best},i}$  表示从当前最优到次优依次排列的解; $N(0,1)$  为正态分布.

改进后的无性繁殖部分不是单纯地复制最优珊瑚虫,而是按高斯变异的方式在最优解附近根据正态分布变异,这种变异方式在搜索后期种群趋于稳定后,仍可以对最优解进行微调,进一步提高搜索精度.

### 2.3 更替方式改进

原始珊瑚礁算法的更替方式为比较子代珊瑚虫与父代珊瑚虫的适应度大小,若子代适应度更好则代替父代,否则不代替. 这种单纯的优势代替劣势,虽然使种群向优势种群收敛,但大大缩减了种群的多样性,未考虑到劣势种群的潜力,对于多峰值函数问题,极易陷入局部最优. 模拟退火算法来源于固体退火原理,具有保留种群多样性、易于跳出局部最优的优点. 因此在 CRO 更替方式部分加入模拟退火算法改进以保留其种群多样性,提高其跳出局部最优的能力. 本文受到模拟退火算法<sup>[16]</sup>的启发对其进行改进,将更替方式改为式(7).

$$P = \begin{cases} 1 & , f_{i+1} < f_i; \\ \exp(- (f_{i+1} - f_i)/T) & , f_{i+1} \geq f_i. \end{cases} \tag{7}$$

式中: $f_i$  为第  $i$  个珊瑚虫的适应度; $i+1$  为第  $i$  个珊瑚虫的子代; $T$  表示温度,温度随着代数的增长而下降,即  $T=0.7T$ . 适用度的值越小,表示适应度越好,当子代适应度好于父代时,子代替父代(即代替概率  $P$  为 1);当子代适用度未好于父代时,以代替概率  $P$  代替父代.

改进后珊瑚礁算法的更替方式,不再是单纯地优势代替劣势,而是以一定的概率保留了适应度差的珊瑚虫,使其可以在下一代中作为父代产生新的子代珊瑚虫,增加了前期种群的多样性,避免种群因收敛过快而陷入局部最优;即使在后期种群陷入某个峰值,也有一定概率跳出局部最优.

改进珊瑚礁算法具体步骤如下:

- ①初始化:同 CRO 算法相同.
- ②外部有性繁殖和内部有性繁殖:外部有性繁殖方式与 CRO 算法相同,内部有性繁殖方式采

- 用式(4)和式(5).
- ③更替机制:更替方式采用式(7).
- ④无性繁殖:无性繁殖方式采用式(6).
- ⑤毁灭机制:与 CRO 算法相同.
- ⑥进入循环,重复步骤②~⑤,直至满足终止条件,即达到最大迭代次数. 得出最优解.

## 3 仿真实验与结果分析

### 3.1 实验参数设置

实验环境为 64 位操作系统,内存为 4 GB, CPU 为 Inter ( R ) Core ( TM ) i5 - 3337U @ 1.80 GHz,程序采用 MATLAB 2014a 编程实现.

将本文算法与原始珊瑚礁优化算法 (CRO)<sup>[17]</sup>、高斯变异珊瑚礁优化算法 (CRO (G))、柯西变异珊瑚礁优化算法 (CRO (C))、高斯与柯西变异珊瑚礁优化算法 (CRO (GC))<sup>[18]</sup>、粒子群优化算法 (PSO)<sup>[19]</sup>、改进权重粒子群优化算法 (IWPSO)、改进粒子群优化算法 (IPSO)<sup>[14]</sup>、飞蛾扑火优化算法 (MFO)<sup>[20]</sup>、多神经优化算法 (MVO)<sup>[21]</sup> 和正余弦优化算法 (SCA)<sup>[22]</sup> 在测试函数下进行比较.

在仿真实验中,统一设置种群规模为 30,最大迭代次数为 1 000. 低维测试函数见表 1,高维测试函数见表 2. 各算法均实验 30 次,求其平均值与标准差. 针对 15 个低维和 5 个高维测试函数,算法结果见表 3,对应的平均值曲线见图 1. 由于 CRO (GC) 已在文献[18]中被证明优于 CRO (G) 和 CRO (C),因此本文表中未列出 CRO (G) 和 CRO (C) 的数据.

### 3.2 实验结果分析

针对低维测试函数,从表 3 可以看出,ICRO 在测试函数  $f_1$  上要明显好于 CRO, PSO, IWPSO, IPSO, MFO, MVO 和 SCA 算法,并且从图 1 的  $f_1$  可以看出其收敛速度极快,仅仅比 CRO (C) 差一点. 在测试函数  $f_2$  上, ICRO 要明显好于 CRO, IWPSO 和 MFO 算法,比 CRO (G), CRO (C) 和 MVO 精确度更高,虽与 SCA 算法在平均值上相等,但标准差要比 SCA 算法更精确,比 IPSO 要稍差,但从图 1 的  $f_2$  可以看出其收敛速度比 IPSO 要快得多,在第 12 代即达到稳定. 而在测试函数  $f_3$  上,虽然 CRO (G), CRO (GC), IWPSO, MVO 和 SCA 可以达到较好效果, CRO (C) 能够达到 8.3526E-15,但 ICRO 可以达到平均值和标准差都为 0,且从图 1 的  $f_3$  可以看出, ICRO 稳定后在



表 1 低维测试函数  
Table 1 Low-dimensional test functions

函数	函数名	维数	上限	下限	函数	函数名	维数	上限	下限
$f_1$	Ackley	30	- 32. 768	32. 768	$f_9$	Rastrigin	30	- 5. 12	5. 12
$f_2$	Dixon-price	10	- 10	10	$f_{10}$	Rotated Hyper-ellipsoid	10	- 65. 536	65. 536
$f_3$	Griewan	30	- 600	600	$f_{11}$	Sphere	30	- 5. 12	5. 12
$f_4$	Levy	30	- 10	10	$f_{12}$	Styblinski-tang	10	- 5	5
$f_5$	Michalewicz	2	0	$\pi$	$f_{13}$	Sum of Different Powers	10	- 1	1
$f_6$	Perm	10	- 10	10	$f_{14}$	Sum Squares	10	- 10	10
$f_7$	Powell	10	- 4	5	$f_{15}$	Zakharov Function	10	- 5	10
$f_8$	Power Sum	4	0	4					

表 2 高维测试函数  
Table 2 High-dimensional test functions

函数	函数名	维数	上限	下限
$f_{16}$	Ackley	500	- 32. 768	32. 768
$f_{17}$	Griewan	500	- 600	600
$f_{18}$	Levy	500	- 10	10
$f_{19}$	Rastrigin	500	- 5. 12	5. 12
$f_{20}$	Ackley	1 000	- 32. 768	32. 768

780 代时跳出局部最优,进一步搜索最优值. 针对测试函数 $f_4$ 和 $f_9$ 这种很难得到满意最优值的函数,ICRO 对于 $f_4$ 可以达到 $3.986\ 6\text{E}-06$ ,且从图 1 的 $f_4$ 可以明显看出 ICRO 曲线的波动,很好地展示了其跳出局部最优的能力,对于 $f_9$ ,甚至可以达到平均值与标准差都为 0,且图 1 的 $f_9$ 进一步证明其跳出局部最优的能力. 针对测试函数 $f_5$ ,PSO,IWPSO,MFO,MVO 和 ICRO 都可以达到理论最优值 $-1.801\ 3$ ,而 ICRO 算法与效果最好的 IPSO 和 MFO 达到了相同的平均值和标准差,且从图 1 的 $f_5$ 可以看出其收敛速度明显好于这两种算法,其在第 10 代达到稳定,而 IPSO 和 MVO 在 100 代左右才达到稳定. 针对测试函数 $f_6$ ,虽然从图 1 的 $f_6$ 可以看到 ICRO 具有极高的收敛速度,但其精确度较差,与大部分对比算法一样未能达到 1 以下的效果,而 IPSO 可以精确到 0.04. 在测试函数 $f_7,f_{11},f_{13},f_{14}$ 和 $f_{15}$ 上,ICRO 算法精确度要明显好于其他算法. 针对测试函数 $f_8$ ,ICRO 可以达到 0.039 51,且具有较高的收敛速度,比大部分算法要好,但 SCA 达到了 $1.525\ 1\text{E}-5$ . 在测试函数 $f_{10}$ 上,虽然 SCA 达到了 $2.903\ 0\text{E}-245$ ,但

ICRO 更加精确,达到了 $1.510\ 3\text{E}-251$ ,且标准差为 0. 在测试函数 $f_{12}$ 上,ICRO 能与对比算法中效果最好的 IPSO 达到相同的平均值,在标准差上稍微差一点,但从图 1 的 $f_{12}$ 可以看出其收敛速度明显好于 IPSO.

针对高维函数,从表 3 可看出,在测试函数 $f_{16}$ 上,ICRO 也能达到最优的效果. 在测试函数 $f_{17}$ 上,虽然 CRO (G) 可以达到 $7.401\ 5\text{E}-18$ ,但 ICRO 在平均值和标准差上都达到 0. 且从图 1 的 $f_{17}$ 可以看出其收敛速度极快. 在测试函数 $f_{18}$ 上可以看出,ICRO 虽然收敛速度极快,但收敛精度效果有待提高. 在测试函数 $f_{19}$ 上可以看出大部分算法都达不到一个较好的平均值,虽然 CRO (C) 平均值达到了 $5.199\ 3\text{E}-11$ ,但 ICRO 可以在平均值和标准差上都达到 0. 在测试函数 $f_{20}$ 上,即使维度提高到 1 000,ICRO 仍然能达到算法中的最优效果 $8.881\ 8\text{E}-16$ .

总的来说,在所有测试函数中 ICRO 都具有极高的收敛速度,这要归功于 ICRO 在内部有性繁殖部分的改进,既保留了种群多样性,又极大地提高了收敛速度. 在测试函数 $f_1,f_3,f_4,f_5,f_7,f_9,f_{10},f_{11},f_{13},f_{14},f_{15},f_{16},f_{17},f_{18}$ 和 $f_{20}$ 中,ICRO 的精度都是最高的,这要归功于其无性繁殖部分的改进,大大提高了搜索精度. 此外,在测试函数 $f_3,f_4$ 和 $f_9$ 上,ICRO 很好地展示了其跳出局部最优的能力,这要归功于其更替方式的改进,一定程度上丰富了种群的多样性,避免因种群收敛过快而陷入局部最优,即使其陷入了某个峰值,也有很高的概率跳出局部最优.

表 3 测试函数的实验结果  
Table 3 Experimental results of test function

函数	CRO		CRO(GC)		PSO	
	平均值	标准差	平均值	标准差	平均值	标准差
$f_1$	15.447 0	1.001 7	6.809 4E-15	1.674 8E-15	11.724 8	2.5470
$f_2$	2.178 7E+3	2.262 2E+3	0.426 3	0.046 8	0.633 8	0.149 7
$f_3$	53.189 2	37.936 3	0.064 3	0.240 7	1.271 3	0.793 5
$f_4$	32.189 9	11.212 1	5.350 0	10.527 3	10.846 1	3.870 8
$f_5$	0.143 8	-1.774 6	-1.620 5	0.301 2	-1.801 3	4.558 8E-6
$f_6$	55.240 7	49.120 3	425.695 6	175.781 8	7.639 7	7.830 0
$f_7$	55.240 7	49.120 3	1.484 0E-44	2.423 2E-44	0.007 8	0.004 4
$f_8$	0.930 4	1.754 9	2.295 6	3.101 8	0.001 7	9.601 5E-4
$f_9$	227.439 4	28.594 3	228.927 5	24.220 0	90.441 9	24.905 0
$f_{10}$	4.063 8E+3	2.705 5E+3	1.428 4E-41	5.002 3E-41	0.007 3	0.002 4
$f_{11}$	12.891 9	11.486 9	7.151 2E-44	1.503 9E-43	0.032 3	0.008 5
$f_{12}$	-255.077 8	28.376 9	-290.490 7	55.483 0	-359.560 5	14.599 3
$f_{13}$	0.004 3	0.007 3	4.637 6E-49	12.48 7E-48	7.862 1E-7	6.002 1E-7
$f_{14}$	109.184 2	74.704 3	3.301 9E-43	7.820 0E-43	0.006 5	0.002 8
$f_{15}$	101.736 8	79.063 3	1.093 7E-43	2.072 6E-43	0.004 4	0.001 5
$f_{16}$	16.379 8	0.490 5	7.283 1E-15	1.421 1E-15	15.230 5	0.980 5
$f_{17}$	829.187 8	534.454 1	2.220 4E-17	4.440 9E-17	805.041 4	150.839 3
$f_{18}$	5.196 8E+3	130.921 1	3.880 0E+3	2.143 6E+3	3.220 3E+3	138.918 0
$f_{19}$	838.429 9	203.310 6	2.857 0	0.242 6	202.102 4	37.914 4
$f_{20}$	16.390 0	0.518 6	7.046 2E-15	1.511E-15	14.669 3	1.053 9

函数	IWPSO		IPSO		MFO	
	平均值	标准差	平均值	标准差	平均值	标准差
$f_1$	3.888 4	0.144 6	0.006 1	0.001 7	10.300 4	8.276 1
$f_2$	14.303 7	9.156 2	0.022 2	0.119 7	28.634 1	64.462 9
$f_3$	0.479 1	0.081 6	3.050 8E-4	1.383 8E-4	6.045 1	22.509 7
$f_4$	1.903 7	0.280 1	6.178 4E-6	2.725 2E-6	33.966 5	17.136 5
$f_5$	-1.801 3	8.170 5E-7	-1.801 3	8.881 8E-16	-1.801 3	8.881 8E-16
$f_6$	1.271 8E+3	1.063 2E+3	0.040 0	0.037 1	3.722 6	5.779 9
$f_7$	8.767 8	7.875 7	2.570 9E-5	1.708 2E-5	24.259 9	37.680 6
$f_8$	0.241 6	0.256 1	0.001 8	0.002 3	0.042 0	0.176 5
$f_9$	238.630 7	24.502 1	238.630 7	24.502 1	160.161 5	30.653 5
$f_{10}$	7.778 4	3.197 6	8.597 4E-17	2.585 1E-16	1.138 1E-29	3.450 5E-29
$f_{11}$	9.668 7	1.681 6	4.009 9E-6	2.030 5E-6	7.864 3	12.012 9
$f_{12}$	-294.085 2	22.283 54	-391.661 7	2.398 2E-13	-360.560 9	16.077 0
$f_{13}$	1.956 5E-4	2.305 0E-4	1.080 7E-43	3.249 6E-43	1.077 4-63	5.146 8E-8
$f_{14}$	7.746 6	3.189 7	4.819 7E-18	7.787 1E-18	9.937 2E-32	1.599 9E-31
$f_{15}$	2.100 1	1.319 5	4.213 9E-10	7.614 7E-10	13.108 6	18.796 9
$f_{16}$	15.230 5	0.980 5	3.893 4	0.045 1	3.893 4	0.045 1
$f_{17}$	0.735 0	0.062 2	0.700 1	0.065 9	8.496 5E+3	341.146 4
$f_{18}$	5.110 3E+3	160.862 6	4.206 9E+3	99.350 4	6.479 2E+3	159.554 2
$f_{19}$	28.697 7	0.705 2	0.705 2	0.765 0	3.425 1E+3	194.053 1
$f_{20}$	3.891 9	0.028 1	3.896 5	0.032 0	20.848 9	0.028 4

续表 3

函数	MVO		SCA		ICRO	
	平均值	标准差	平均值	标准差	平均值	标准差
$f_1$	1.053 9	0.616 2	0.012 0	0.014 7	$8.881\ 8\text{E}-16$	0
$f_2$	0.667 6	0.001 0	0.666 7	$2.756\ 3\text{E}-5$	0.666 7	$4.942\ 7\text{E}-6$
$f_3$	0.059 54	0.104 2	0.303 5	0.281 1	0	0
$f_4$	14.389 9	8.939 3	2.309 7	0.391 8	$3.986\ 6\text{E}-06$	$9.350\ 6\text{E}-06$
$f_5$	-1.801 3	$2.331\ 7\text{E}-7$	-1.638 7	0.319 4	-1.801 3	$8.881\ 8\text{E}-16$
$f_6$	2.265 4	4.167 0	18.198 3	10.497 9	7.975 1	8.221 7
$f_7$	0.001 6	0.001 6	$1.525\ 1\text{E}-5$	$7.568\ 5\text{E}-5$	$1.989\ 5\text{E}-218$	0
$f_8$	$6.505\ 5\text{E}-4$	$5.356\ 7\text{E}-4$	$1.525\ 1\text{E}-5$	$7.568\ 5\text{E}-5$	0.0395 1	0.0599 7
$f_9$	111.065 1	26.275 2	19.654 0	23.225 0	0	0
$f_{10}$	0.011 7	0.006 5	$2.903\ 0\text{E}-245$	$1.360\ 1\text{E}-24$	$1.510\ 3\text{E}-251$	0
$f_{11}$	$7.801\ 0\text{E}-4$	$2.515\ 6\text{E}-4$	$83\ 703\text{E}-5$	$2.509\ 3\text{E}-4$	$2.502\ 9\text{E}-230$	0
$f_{12}$	-345.952 8	22.702 0	-312.042 1	24.758 4	-391.661 7	$2.238\text{E}-07$
$f_{13}$	$1.118\ 0\text{E}-8$	$1.422\ 8\text{E}-8$	$1.118\ 0\text{E}-8$	$1.422\ 8\text{E}-8$	0	0
$f_{14}$	$2.870\ 5\text{E}-4$	$1.493\ 4\text{E}-4$	$1.090\ 8\text{E}-28$	$4.552\ 3\text{E}-28$	$7.631\ 7\text{E}-253$	0
$f_{15}$	$1.374\ 1\text{E}-4$	$7.815\ 5\text{E}-5$	$4.622\ 9\text{E}-13$	$2.478\ 5\text{E}-12$	$1.483\ 0\text{E}-237$	0
$f_{16}$	20.810 9	0.101 1	13.726 3	2.063 7	$8.881\ 8\text{E}-16$	0
$f_{17}$	258.166 0	17.215 0	$1.346\text{E}+3$	529.646 7	0	0
$f_{18}$	$5.844\ 8\text{E}+3$	189.699 1	$1.080\ 3\text{E}+3$	400.272 5	0	0
$f_{19}$	$1.477\ 2\text{E}+3$	91.817 1	$1.529\ 4\text{E}+3$	413.390 0	44.056 1	0.304 7
$f_{20}$	21.006 7	0.065 8	14.784 0	1.534 6	$8.881\ 8\text{E}-16$	0

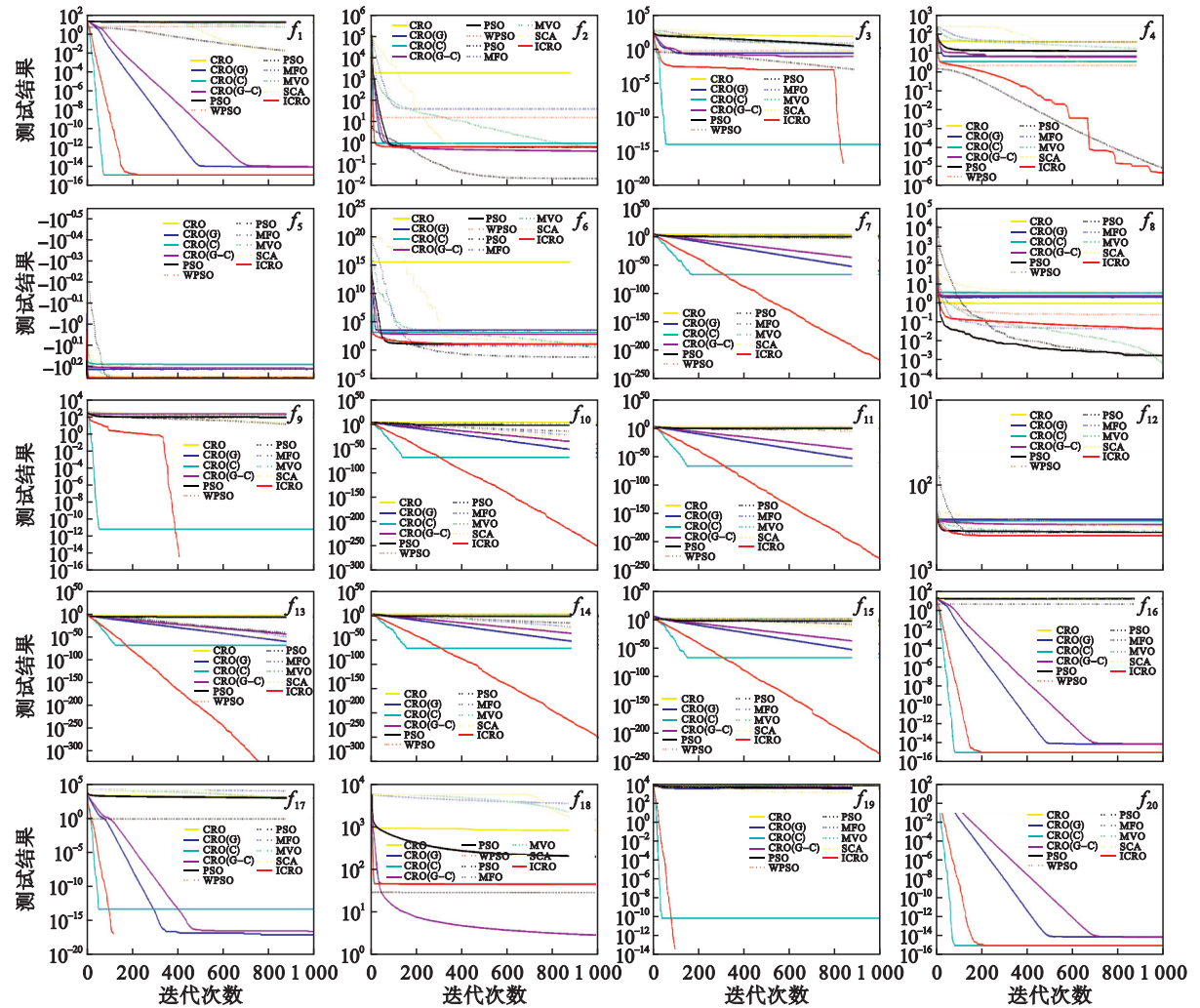


图 1 测试函数的实验结果图  
Fig. 1 Experimental results of the test function

## 4 结 语

本文针对现有珊瑚礁算法存在的不足,提出了改进的珊瑚礁算法,在测试函数下证明了其相比其他算法可以更好地跳出局部最优,有效地改善了其原本易于陷入局部最优的缺点. 相比其他算法,改进的珊瑚礁算法还具有更快的收敛速度和更高的搜索精度,即使在高维测试函数下仍然可以保持良好的性能.

## 参考文献:

- [ 1 ] Salcedo-Sanz S, Del Ser J, Gil-Lopez S, et al. The coral reefs optimization algorithm; an efficient meta-heuristic for solving hard optimization problems [ C ] // 15th Applied Stochastic Models and Data Analysis International Conference. Barcelona, 2014: 751 – 758.
- [ 2 ] Salcedo-Sanz S, Pastor-Sanchez A, Prieto L, et al. Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization-extreme learning machine approach [ J ]. *Energy Conversion and Management*, 2014, 87: 10 – 18.
- [ 3 ] Salcedo-Sanz S, Pastor-Sanchez A, Del Ser J, et al. A coral reefs optimization algorithm with harmony search operators for accurate wind speed prediction [ J ]. *Renewable Energy*, 2015, 75: 93 – 101.
- [ 4 ] Salcedo-Sanz S, Pastor-Sanchez A, Portilla-Figueras J A, et al. Effective multi-objective optimization with the coral reefs optimization algorithm [ J ]. *Engineering Optimization*, 2016, 48(6): 966 – 984.
- [ 5 ] Salcedo-Sanz S, Camacho-Gomez C, Mallol-Poyato R, et al. A novel coral reefs optimization algorithm with substrate layers for optimal battery scheduling optimization in micro-grids [ J ]. *Soft Computing*, 2016, 20(11): 4287 – 4300.
- [ 6 ] Salcedo-Sanz S, Camacho-Gomez C, Magdaleno A, et al. Structures vibration control via tuned mass dampers using a co-evolution coral reefs optimization algorithm [ J ]. *Journal of Sound and Vibration*, 2017, 393: 62 – 75.
- [ 7 ] Salcedo-Sanz S, Jimenez-Fernandez S, Aybar-Ruiz A, et al. A CRO-species optimization scheme for robust global solar radiation statistical downscaling [ J ]. *Renewable Energy*, 2017, 111: 63 – 76.
- [ 8 ] Duran-Rosal A M, Gutierrez P A, Salcedo-Sanz S, et al. A statistically-driven coral reef optimization algorithm for optimal size reduction of time series [ J ]. *Applied Soft Computing*, 2018, 63: 139 – 153.
- [ 9 ] Salcedo-Sanz S, Gallo-Marazuela D, Pastor-Sanchez A, et al. Offshore wind farm design with the coral reefs optimization algorithm [ J ]. *Renewable Energy*, 2014, 63(1): 109 – 115.
- [ 10 ] Salcedo-Sanz S, Casanova-Mateo C, Pastor-Sanchez A, et al. Daily global solar radiation prediction based on a hybrid coral reefs optimization-extreme learning machine approach [ J ]. *Solar Energy*, 2014, 105: 91 – 98.
- [ 11 ] Li M, Miao C, Leung C. A coral reef algorithm based on learning automata for the coverage control problem of heterogeneous directional sensor networks [ J ]. *Sensors*, 2015, 15(12): 30617 – 30635.
- [ 12 ] 朱重龙, 何登旭. 珊瑚礁算法及其应用研究 [ D ]. 南宁: 广西民族大学, 2016.  
(Zhu Zhong-long, He deng-xu. Coral reef algorithm and its application [ D ]. Nanning: Guangxi University for Nationalities, 2016. )
- [ 13 ] 全亚威, 田娜, 纪志成, 等. 基于珊瑚礁算法的永磁同步电机参数辨识 [ J ]. 系统仿真学报, 2016, 28(4): 927 – 939.  
(Quan Ya-wei, Tian Na, Ji Zhi-cheng, et al. Coral reefs optimization for solving parameter identification in permanent magnet synchronous motor [ J ]. *Journal of System Simulation*, 2016, 28(4): 927 – 939. )
- [ 14 ] Wu P F, Gao L Q, Zou D X, et al. An improved particle swarm optimization algorithm for reliability problems [ J ]. *ISA Transactions*, 2011, 50(1): 71 – 81.
- [ 15 ] 曲良东, 何登旭. 基于自适应高斯变异的人工鱼群算法 [ J ]. 计算机工程, 2009, 35(15): 182 – 189.  
(Qu Liang-dong, He Deng-xu. Artificial fish-school algorithm based on adaptive gauss mutation [ J ]. *Computer Engineering*, 2009, 35(15): 182 – 189. )
- [ 16 ] Dupanloup I, Schneider S, Excoffier L. A simulated annealing approach to define the genetic structure of populations [ J ]. *Molecular Ecology*, 2002, 11(12): 2571 – 2581.
- [ 17 ] Yang Y K, Yang B T, Niu M Q. Parameter identification of Jiles-Atherton model for magnetostrictive actuator using hybrid niching coral reefs optimization algorithm [ J ]. *Sensors and Actuators A: Physical*, 2017, 261: 184 – 195.
- [ 18 ] Salcedo-Sanz S, Del Ser J, Landa-Torres I, et al. The coral reefs optimization algorithm: a novel metaheuristic for efficiently solving optimization problems [ J/OL ]. *Scientific World Journal*, 2014 [ 2018 – 10 – 09 ]. [http://dx. doi. org/ 10. 1155/2014/739768](http://dx.doi.org/10.1155/2014/739768).
- [ 19 ] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization [ J ]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 256 – 279.
- [ 20 ] Mirjalili S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm [ J ]. *Knowledge-Based Systems*, 2015, 89: 228 – 249.
- [ 21 ] Mirjalili S, Mirjalili S M, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization [ J ]. *Neural Computing & Applications*, 2016, 27(2): 495 – 513.
- [ 22 ] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems [ J ]. *Knowledge-Based Systems*, 2016, 96: 120 – 133.