文章编号：2096-1472(2022)-08-15-05
DOI:10.19644/j.cnki.issn2096-1472.2022.008.004

基于Tent映射的改进型人工蜂群算法

王贺琦，王 然，刘诗琳，唐浩桁，马云鹏

(天津商业大学信息工程学院，天津 300134)
✉384410733@qq.com; kad2001628@163.com; 1595996480@qq.com;
tanghaoheng@163.com; mayunpeng@tjcu.edu.cn

摘 要：人工蜂群(Artificial Bee Colony，ABC)算法是受蜜蜂的觅食行为启发而提出的群体智能算法，其利于解决高维连续函数的寻优问题。针对ABC算法的不足，分别对其种群个体初始化方式和更新机制进行改进。采用混沌映射和反向学习机制初始化种群个体位置，益于保持种群多样性；将新型惯性权值引入个体位置更新机制，以提高算法的收敛精度和平衡其探索能力与开发能力。通过13 个基准测试函数验证，结果表明：相比于原始ABC算法，改进的ABC算法在9 个函数上表现出良好的收敛精度和解的质量。

关键词：群体智能；人工蜂群算法；混沌映射；反向学习机制
中图分类号：TP181 **文献标识码**：A

www.rjgczz.com

Modified Artificial Bee Colony Algorithm based on Tent Mapping

WANG Heqi, WANG Ran, LIU Shilin, TANG Haoheng, MA Yunpeng

(*College of Information Engineering*, *Tianjin University of Commerce*, *Tianjin* 300134, *China*)
✉384410733@qq.com; kad2001628@163.com; 1595996480@qq.com;
tanghaoheng@163.com; mayunpeng@tjcu.edu.cn

Abstract: Artificial Bee Colony (ABC) is a kind of swarm intelligence algorithm inspired by the foraging behavior of honeybees, which is beneficial to solve the optimization problem of high-dimensional continuous function. Aiming at the shortcomings of ABC algorithm, this paper proposes to modify the initialization method and update mechanism of population individuals. The chaotic mapping and reverse learning mechanism are used to initialize individual positions, which is good for maintaining population diversity. The novel inertia weight is introduced in the individual position update mechanism to improve the convergence accuracy and balance its exploration and development ability. Through 13 benchmark functions verification, the experiment results show that compare with the conventional artificial bee colony algorithm, the modified artificial bee colony (MABC) algorithm represents better convergence accuracy and solution quality on 9 functions.

Keywords: swarm intelligence; artificial bee colony algorithm; chaotic mapping; reverse learning mechanism

1 引言(Introduction)

群体智能(Swarm Intelligence)是指具有简单智能的个体通过相互协作和组织表现出群体智能行为的特性，具有天然的分布式和自组织特征[1]。无论外界环境如何变化，蜜蜂总能找到蜜源和巢穴，对外界环境变化具有很好的适应性。蜜蜂属于群居性昆虫，其行为目标简单明确，各司其职，可以高效快速地找到蜜源。当外界环境变化时，蜜蜂可以灵活地调整劳动分工，如减少或增加采蜜的工蜂[2]。针对以上现象和行为，文献[3]提出了ABC算法，并将其应用于函数的数值优化问题上，且取得良好的效果。类比自然界中的蜜蜂采蜜行

基金项目：天津市自然科学基金(20JCQNJC00430)；大学生创新创业训练项目(202110069003，202110069034，202010069066).

为，ABC算法是将各蜜蜂的初始位置定义为搜索空间的初始点；把蜜蜂的进化和觅食行为模拟为种群个体更新机制；将蜜蜂对环境的适应性类比为解的适应度函数值；将蜜源的质量作为目标函数值。根据以上分析，ABC算法属于一种启发式迭代优化算法。

ABC算法具有控制参数少、全局优化能力强的优点，引起国内外研究学者的广泛关注，并将其成功应用于各个领域。然而，ABC算法仍存在一些不足，如局部搜索能力差、收敛精度低、收敛速度慢等。针对上述不足，文献[4]提出了分段式蜜源搜索半径公式，提高了算法的收敛精度和收敛速度，并采用改进的ABC算法对径向基神经网络的初始参数进行优化。文献[5]针对传统ABC算法收敛速度慢的问题，引入自适应参数，加快了算法的后期收敛能力，实验结果表明改进的算法收敛能力更强，收敛速度更快。文献[6]在ABC算法的跟随蜂阶段，使种群依概率进行反向学习代替跟随蜂搜索方式，雇佣蜂和侦察蜂阶段用以保证种群的探索能力以及种群的多样性。文献[7]运用最大最小距离积法处理初始蜜源，将K均值聚类算法与ABC算法相结合，提高了ABC算法的收敛速度，具有较好的解适应度。ZHU等人改进ABC算法的搜索模式来提高局部搜索能力，提出改进的ABC算法[8]。文献[9]将交叉机制与全局最优解信息融入ABC算法，增强了ABC算法的开发能力和探索能力。曾志文等人[10]提出一种双种群架构ABC算法，将交叉变异操作和最优解邻域搜索融入不同种群，提高了算法的寻优能力。

为进一步提高ABC算法的性能，本文提出一种改进的ABC算法(Modified Artificial Bee Colony，MABC)。首先，分别采用Tent混沌映射函数和反向学习机制初始化种群个体的位置，使初始解在搜索空间中的分布更加均匀，以增强种群的多样性，有助于提高算法的全局搜索能力；然后，在雇佣蜂阶段，将分组机制和惯性权值引入种群个体的更新机制中，以提高算法的收敛精度和平衡算法的探索能力与开发能力。

为验证MABC算法的有效性，本文应用13 个基准数学函数对MABC算法的收敛精度和全局搜索能力等性能进行测试。与原始ABC算法对比，实验结果表明MABC算法具有良好的收敛速度和收敛精度。

2 ABC算法简介(Introduction to ABC algorithm)

在标准ABC算法中，有三种角色：引领蜂、跟随蜂、侦察蜂。引领蜂又称为雇佣蜂，跟随蜂和侦查蜂被称为非雇佣蜂。ABC算法的搜索过程简要描述如下：首先，所有蜜蜂都是侦查蜂，在整个解空间随机搜索。当侦查蜂搜索到蜜源后，将蜜源质量信息带回蜂巢与其他蜜蜂共享，对侦察蜂带回的所有信息进行比较。当该蜜源的质量排名靠前时，该侦查蜂就转变为引领蜂，蜜源和引领蜂一一对应，另外招募到一些跟随蜂到引领蜂所在的蜜源附近搜索新的蜜源；当蜜源的质量排名居中时，该侦查蜂就转变为跟随蜂，然后根据一定的规则选择引领蜂，到引领蜂对应的蜜源附近进行搜索；当该蜜源的质量排名靠后时，该侦查蜂就放弃已经搜索到的蜜源，再次成为侦查蜂，在整个区域进行新一轮的随机搜索。

ABC算法在求解优化问题时，蜜源的位置代表优化问题的可行解，蜜源的质量代表优化问题的值，寻找蜜源的过程就是优化问题的过程，蜜源的最大质量即优化问题的最优解。在整个算法中，引领蜂和跟随蜂是为了加快算法的收敛，侦察蜂主要负责跳出局部最优解。

蜂群初始化时通过式(1)随机产生N个解，每个解代表一个蜜源：$X_i=\left|X_{ij}\right|\ (i=1,2,\cdots,N;j=1,2,\cdots,D)$，是一个$D$ 维向量。

$$X_{ij}=X_{ij}+rand(0,1)\times(X_{ij^{\max}}-X_{ij^{\min}}) \tag{1}$$

其中，$X_{ij^{\max}}$是X_{ij}的上限，$X_{ij^{\min}}$是X_{ij}的下限，$rand(0,1)$为0到1之间的一个随机数。

引领蜂和跟随蜂的位置根据式(2)进行更新。V_{ij}代表在X_{ij}附近产生的一个新解，k和j均是随机选取的，其中k不等于i，r是[−1,1]之间的一个随机数，此参数决定扰动幅度。

$$V_{ij}=X_{ij}+r\times\left(X_{ij}-X_{kj}\right) \tag{2}$$

跟随蜂对引领蜂的跟随依赖于解的适应度值和概率大小，适应度值f_it_i和选择概率p_i的计算式如式(3)、式(4)所示。其中，f_i为第i 个参数对应的函数值。

$$f_it_i=\begin{cases}1/(1+f_i),\ f_i<0\\ 1+\mathrm{abs}(f_i),\ f_i\geqslant 0\end{cases} \tag{3}$$

$$p_i=f_it_i/\sum_{i=1}^{N}f_it_i \tag{4}$$

在该算法中，控制参数的设置能有效平衡全局搜索与局部搜索。如果引领蜂在其蜜源邻域搜索次数达到最大未更新次数时，仍没有找到适应度值更高的新解，表明这个解为局部最优值，就放弃该蜜源；然后，根据式(1)再次随机初始化该引领蜂的位置。

3 MABC算法(MABC algorithm)

3.1 初始解产生方式的改进

在原始ABC算法中，种群初始化是经过伪随机序列生成的，种群多样性有非常大的不确定性，也不能保证解均匀地分布在解空间中。因此，有必要对种群初始化方式进行改进。由于混沌映射具有遍历性和随机性，若采用混沌映射函数产生混沌序列作为初始种群个体的位置，将增强初始解的

www.rjgcz.com

遍历均匀性。因此，本文采用Tent混沌映射函数初始化种群个体，如式(5)所示，其中$a=0.5$。种群个体混沌初始化后，将其转化到真实的解空间。除此之外，本文同时采用反向学习机制对转化后的解进行处理，如式(6)所示。每个随机产生的候选解相比它的反向解有50%的概率远离问题最优解[11]，选择两组解中适应度值大的作为初始种群成员，这将在很大程度上加速收敛，有利于提高全局搜索能力，更易接近最优解。

$$x_{i+1}=\begin{cases}\dfrac{x_i}{a} & 0\leqslant x_n\leqslant a\\ \dfrac{1-x_i}{1-a} & a\leqslant x_n\leqslant 1\end{cases} \tag{5}$$

$$X_{ij}'=X_{ij^{\max}}+X_{ij^{\min}}-X_{ij} \tag{6}$$

其中，X_{ij}'是与X_{ij}相反的位置点。

3.2 种群个体更新机制的改进

在原始ABC算法中，r是[−1,1]之间的一个随机数。解更新操作对原个体的影响较小，保留了很多特征到新解，极大影响了种群收敛的速度和精度。另一方面，系数r是一个随机数，不能动态地、自适应地改变搜索步长，造成了原始ABC算法随机性大的缺陷。本文采用一种新雇佣蜂更新机制，其基本思想如下：每次迭代更新前，找出当前解中最优秀的解向量，让较差解向最优解学习，让雇佣蜂根据式(7)进行更新，以便较快地提高算法的收敛精度和收敛速度。随着种群迭代的进行，在算法前期，较差解向最优解加速靠拢；在算法后期，较差解在其周围邻域内搜索。此外，引入惯性权值W，如式(8)所示。惯性权值是非线性收敛因子调整策略，它是逐级递增的，前期距离控制参数较大，保证较少的特性遗留到下一代，搜索范围较大，因此有利于全局寻优；在迭代后期距离控制参数大，较多保留了上一代优秀解的特性，在优秀解附近进行精细搜索，有利于局部寻优，能够有效平衡算法的全局搜索能力和局部搜索能力。三角函数权值系数也有助于提高算法的运行速度和全局搜索能力。

$$X_{ij}^{'}=X_{ij}\times W+r\times(X_{\text{best}}-X_{ij}) \tag{7}$$

$$W=\sin(\frac{\pi}{2}\times\frac{\text{iter}}{\text{MaxCycle}}) \tag{8}$$

其中，iter是当前迭代次数，MaxCycle为种群最大迭代次数，r是[−1,1]之间的随机数，X_{best}是种群最好的解位置。

4 MABC算法的性能测试(Performance test of MABC algorithm)

为验证MABC算法的性能，本文选用13个经典数学函数集测试MABC算法的收敛速度和收敛精度。表1给出13个函数的详细描述，包括函数表达式、解的取值范围、最优点的位置及最优值等。在表1中，F1—F7为单峰函数，用于测试MABC算法的收敛精度和求解能力；F8—F13为多峰函数，用于验证MABC算法的全局搜索能力。本文将MABC算法与原始ABC算法进行对比，两种算法的相关设置参数如表2所示。为验证算法的稳定性和对比公平性，每个算法独立运行20次，每次迭代1,000次，分别测试每个函数10维、30维、50维的函数值，并求20次最优函数值的平均值和均方差，实验结果记录于表3。平均值和均方差越小，说明算法的收敛精度和稳定性越好。表3中加粗的字体表示最优性能参数。本文采用的实验设备为：64位MateBook X Pro 2019款笔记本；处理器型号为：Intel(R) Core(TM) i5−8265U CPU @ 1.60 GHz 1.80 GHz；运行内存为8.00 GB；仿真软件为：MATLAB 2014b。由表3可知，相较于ABC算法，MABC算法在F1、F2、F3、F4、F7、F9、F10、F11、F12等9个函数上取得更小的最优解的平均值和均方根误差，说明MABC算法具有良好的收敛精度和稳定性。虽然ABC算法在F5、F6、F8及F13等4个函数上表现出较好的收敛精度和稳定性，但两种算法在收敛精度上差距不大。

表1 经典测试函数集

Tab.1 Classic test function set

函数	优化函数	迭代次数	取值范围	理论最优值	理论最优值点
F1	$f(x)=\sum_{i=1}^{n}(x_i)^2$	1,000	[−100,100]	0	$[0]^n$
F2	$f(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	1,000	[−10,10]	0	$[0]^n$
F3	$f(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$	1,000	[−100,100]	0	$[0]^n$
F4	$f(x)=\max\lvert x_i\rvert,1\leqslant i\leqslant n$	1,000	[−100,100]	0	$[0]^n$
F5	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i^2-1)]$	1,000	[−30,30]	0	$[1]^n$
F6	$f(x)=\sum_{1}^{n-1}(x_i+0.5)^2$	1,000	[−100,100]	0	$[0]^n$
F7	$f(x)=\sum_{i=1}^{n}(ix_i)^4-\text{random}[0,1)$	1,000	[−1.28,1.28]	0	$[0]^n$
F8	$f(x)=\sum_{1}^{n}-x_i\sin\sqrt{\lvert x_i\rvert}$	1,000	[−500,500]	$-418.9829\times n$	$[420.96]^n$
F9	$f(x)=\sum_{1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	1,000	[−5.12,5.12]	0	$[0]^n$
F10	$f(x)=-20\exp\left(-0.2\text{sqrt}\left(\frac{1}{n}\sum_{1}^{n}\cos(x_i^2)\right)\right)-\exp(\frac{1}{n}\sum_{1}^{n}\cos(2\pi x_i))+20+e$	1,000	[−32,32]	0	$[0]^n$
F11	$f(x)=\frac{1}{4000}\sum_{1}^{n}x_i^2-\prod_{1}^{n}\cos(x_i/\sqrt{i})+1$	1,000	[−600,600]	0	$[0]^n$
F12	$f(x)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i}^{n-1}(y_i-1)^2\times[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i}^{n}u(x_i,10,100,4)$	1,000	[−50, 50]	0	$[0]^n$
F13	$f(x)=0.1\{\sin(3\pi x_i)+\sum_{i}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})+(y_n-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i}^{n}u(x_i,5,100,4)$	1,000	[−50, 50]	0	$[0]^n$

表2 初始参数设置

Tab.2 Initial parameter setting

算法	种群个数	解未更新次数	迭代次数	维数
ABC	80	100	1,000	10、30、50
MABC	80	100	1,000	10、30、50

www.rjgczz.com

表3 测试结果

Tab.3 Test results

函数	维数	ABC		MABC	
		Mean	Std	Mean	Std
F1	10	1.2246×10^{-16}	6.2829×10^{-17}	**1.6911×10^{-20}**	**4.9224×10^{-21}**
	30	1.1901×10^{-9}	8.1206×10^{-10}	**7.1260×10^{-20}**	**2.8535×10^{-20}**
	50	1.3492×10^{-5}	1.4401×10^{-5}	**1.2019×10^{-19}**	**5.4009×10^{-20}**
F2	10	3.2414×10^{-16}	7.7894×10^{-17}	**5.4284×10^{-18}**	**3.8581×10^{-18}**
	30	2.0466×10^{-6}	7.4607×10^{-7}	**2.2376×10^{-18}**	**6.6570×10^{-19}**
	50	0.0023	6.8105×10^{-4}	**3.5109×10^{-18}**	**1.1083×10^{-18}**
F3	10	1.3776×10^{3}	1.2406×10^{3}	**4.2781×10^{-18}**	**8.1499×10^{-18}**
	30	3.1156×10^{4}	9.6316×10^{3}	**1.8958×10^{4}**	**2.9886×10^{-18}**
	50	9.0757×10^{4}	2.3104×10^{4}	**1.8360×10^{-18}**	**2.4619×10^{-18}**
F4	10	0.6207	0.3575	**4.6474×10^{-18}**	**2.9243×10^{-18}**
	30	52.5440	10.6779	**1.8130×10^{-18}**	**2.7418×10^{-19}**
	50	81.0938	3.7565	**1.8694×10^{-18}**	**1.8763×10^{-19}**
F5	10	0.2176	0.3629	7.4878	0.1071
	30	1.5749	1.5758	28.1719	0.1784
	50	23.2440	20.0386	48.3288	0.1278
F6	10	1.3019×10^{-16}	6.5969×10^{-17}	1.8190×10^{-16}	7.3676×10^{-17}
	30	1.3331×10^{-9}	1.2553×10^{-9}	2.2473×10^{-9}	5.3004×10^{-5}
	50	9.8019×10^{-6}	7.8062×10^{-6}	0.3513	0.1407
F7	10	0.0118	0.0041	**2.2135×10^{-5}**	**2.2584×10^{-5}**
	30	0.1517	0.0245	**1.9716×10^{-5}**	**3.5880×10^{-5}**
	50	0.5032	0.0736	**1.8531×10^{-5}**	**1.4699×10^{-5}**
F8	10	5263.5	231.2026	**421.86**	**128.7014**
	30	-1.2233×10^{4}	84.2260	-1.1739×10^{4}	155.1503
	50	$-1.9432.\times10^{4}$	197.0438	-1.8094×10^{4}	246.12152
F9	10	0	0	**0**	**0**
	30	0.0581	0.2220	**0**	**0**
	50	5.5662	2.0839	**0**	**0**
F10	10	7.9936×10^{-15}	1.6301×10^{-15}	**8.8818×10^{-16}**	**0**
	30	1.0906×10^{-5}	3.7931×10^{-6}	**8.8818×10^{-16}**	**0**
	50	0.0197	0	**8.8818×10^{-16}**	**0**
F11	10	0.0025	0.0055	**0**	**0**
	30	0.0018	0.0060	**0**	**0**
	50	0.0068	0.0162	**0**	**0**
F12	10	3.9796×10^{-4}	7.3455×10^{-4}	5.4435×10^{-4}	9.9046×10^{-4}
	30	39.3090	58.4399	**0.0711**	**0.1621**
	50	186.7767	62.6623	**0.2259**	**0.1932**
F13	10	1.9908×10^{-4}	2.4486×10^{-4}	0.0014	0.0043
	30	0.3476	0.4703	1.3425	1.3583
	50	12.8388	46.6625	**3.8715**	**2.0294**

为了更加直观地对比MABC算法与ABC算法的收敛性，图1—图8分别给出部分函数在两个算法中某个维度的仿真图，其中MABC算法用虚线表示，ABC算法用实线表示。由8个仿真图可知，无论是单峰函数还是多峰函数，MABC算法的收敛速度和收敛精度明显优于原始ABC算法。综上所述，MABC算法具有良好的收敛速度和全局搜索能力。

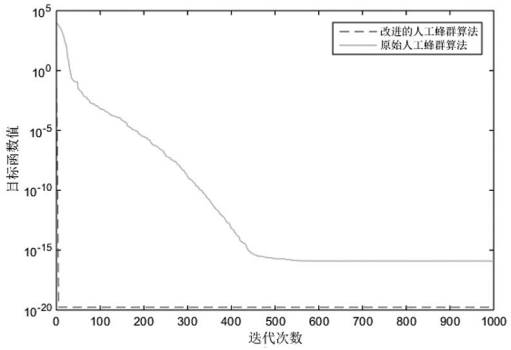

图1 F1=10适应度值对比

Fig.1 Comparison of fitness value of F1 with dim=10

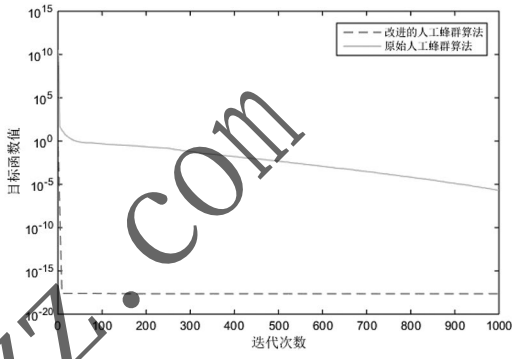

图2 F2=30适应度值对比

Fig.2 Comparison of fitness value of F2 with dim=30

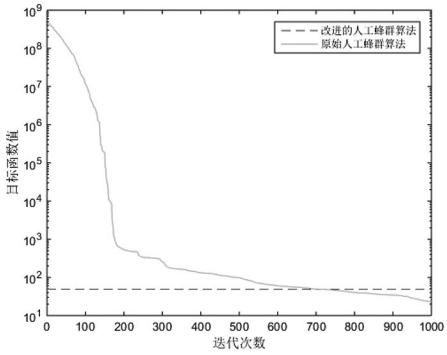

图3 F5=50适应度值对比

Fig.3 Comparison of fitness value of F5 with dim=50

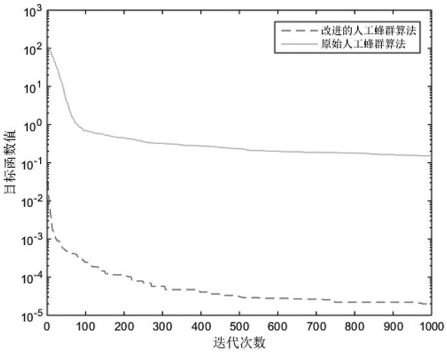

图4 F7=30适应度值对比

Fig.4 Comparison of fitness value of F7 with dim=30

www.rjgczz.com

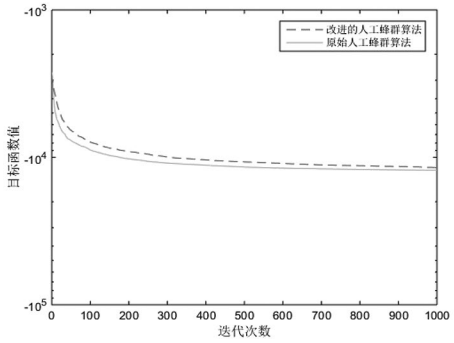

图5 F8=30适应度值对比

Fig.5 Comparison of fitness value of F8 with dim=30

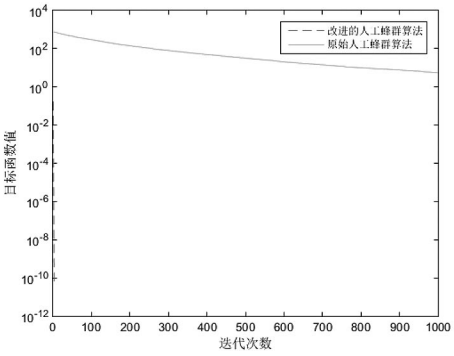

图6 F9=50适应度值对比

Fig.6 Comparison of fitness value of F9 with dim=50

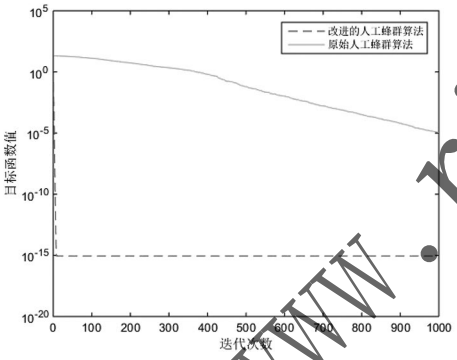

图7 F10=30适应度值对比

Fig.7 Comparison of fitness value of F10 with dim=30

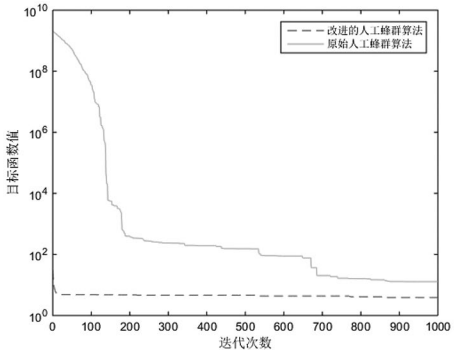

图8 F13=50适应度值对比

Fig.8 Comparison of fitness value of F13 with dim=50

5 结论(Conclusion)

为进一步提升ABC算法的性能，分别对ABC算法的种群个体初始化方式和雇佣蜂阶段的种群个体更新机制进行改进。应用Tent混沌映射函数和反向学习机制对种群个体的位置进行初始化，增强了初始个体在解空间的均匀分布性，提高了种群个体多样性。在雇佣蜂阶段，引入新型惯性权值和改进种群个体更新机制，有利于提高收敛速度和平衡算法的全局探索能力及局部开发能力。通过13 个基准测试函数的验证，实验结果显示MABC算法具有良好的收敛速度和全局搜索能力。

参考文献(References)

[1] SMITH A E. Swarm intelligence: From natural to artificial systems[J]. IEEE Transactions on Evolutionary Computation, 2000, 4(2):192−193.

[2] ROBINSON G E. Regulation of division of labor in insect societies[J]. Annual Review of Entomology, 1992, 37(1): 637−665.

[3] KARABOGA D, AKAY B. A comparative study of Artificial Bee Colony algorithm[J]. Applied Mathematics and Computation, 2009, 214(1):108−132.

[4] 管雪梅,黄青龙,黄靖一,等.基于改进人工蜂群算法的大青杨晚材率预测[J].科学技术与工程,2021,21(26):11118−11124.

[5] 王训豪,李树江.基于改进人工蜂群算法的中央空调冷却水系统优化研究[J].电脑知识与技术:学术版,2021,17(33):86−90.

[6] 王剑,王冰,葛孟珂.基于反向学习的人工蜂群算法[J].牡丹江师范学院学报(自然科学版),2022(01):23−30.

[7] 贺井然,何广军,于学生.基于改进蜂群算法的无人机路径规划[J].火力与指挥控制,2021,46(10):103−106.

[8] ZHU G, KWONG S. Gbest−guided artificial bee colony algorithm for numerical function optimization[J]. Applied Mathematics and Computation, 2010, 217(7):3166−3173.

[9] 任金霞,杜增正,王兴康.改进人工蜂群算法的云任务调度[J].河南科技大学学报(自然科学版),2022,43(04):55−60,6−7.

[10] 曾志文,陈晓,郭冬,等.双种群人工蜂群算法及其在MT和重力联合反演中的应用[J].石油地球物理勘探,2021,56(06): 1400−1408,1447,1203.

[11] 吴昱,李元香,徐星.基于群智能的新型反向混合差分进化算法[J].小型微型计算机系统,2009,30(05):903−907.

作者简介：

王贺琦(1998−)，男，本科生.研究领域：仿生智能计算，机器学习.

王　然(2001−)，女，本科生.研究领域：仿生智能计算.

刘诗琳(2000−)，女，本科生.研究领域：仿生智能计算.

唐浩桁(2000−)，男，本科生.研究领域：模式识别，特征提取.

马云鹏(1989−)，男，博士，讲师.研究领域：复杂过程控制，机器学习，仿生智能计算.本文通讯作者.

www.rigcz.com