

一种改进的猴群算法^①

徐小平, 张东洁

(西安理工大学 理学院, 西安 710049)

摘 要: 猴群算法是一种新的群体智能优化算法, 该算法可以有效地求解线性、非线性、非凸和复杂高维函数的优化问题, 目前已得到了许多学者的研究和关注. 为了进一步提高猴群算法的求解精度, 给出了一种改进的猴群算法. 首先, 采用均匀分布的 Kent 混沌映射产生猴群算法的初始可行解. 然后, 在该算法的爬过程中采用递减的因子作为爬步长. 最后, 在仿真实验中, 与已有方法进行比较, 结果显示了所给改进猴群算法的求解精度明显得到改善, 即所提算法是可行的.

关键词: 智能算法; 猴群算法; Kent 混沌映射; 递减的因子

Improved Monkey Algorithm

XU Xiao-Ping, ZHANG Dong-Jie

(School of Sciences, Xi'an University of Technology, Xi'an 710049, China)

Abstract: Monkey algorithm is a new optimization algorithm of swarm intelligent. The algorithm can effectively solve the optimization problems of functions such as linear, nonlinear, nonconvex and complex high dimensional function, etc. Currently, it has been widely studied and concerned by many scholars. In order to further improve the solution accuracy of monkey algorithm, this paper puts forward an improved monkey algorithm. Firstly, uniformly distributed Kent chaotic map is adopted as the initial feasible solution of the algorithm. Then, the descending factor is used as the step size in the climb process of the algorithm. Finally, in the simulation experiment, compared with the existing methods, the results show that the solution accuracy of the proposed monkey algorithm is significantly improved, namely, the proposed algorithm is feasible.

Key words: intelligent algorithm; monkey algorithm; Kent chaotic map; descending factor

随着科学技术和信息时代的发展, 优化理论不断得到完善, 同时也面临实际生活中, 所需解决的问题复杂度的无限增高、空间维数的不断增大^[1]. 解决这些问题要求的目标是: 在最短时间内、更加准确、更加快速地搜索到最优解. 往往利用传统的方法难以解决这种复杂的、大规模的多维问题. 随着智能优化算法的兴起, 对于求解这种复杂问题给出了新的思路, 借用计算机对其进行求解已成为一种新趋势. 例如, 差分进化算法^[2]、粒子群算法^[3]和遗传算法^[4]等. 它们是由人类受到自然界的生物生存法则的启发后, 通过了解与研究它们种群间的各种行为规则、觅食习惯及群

体中个体间的联系与区别, 设计出类似生物界各种生物的智能优化算法.

猴群算法(Monkey Algorithm, MA)是近来被公认的一种群智能优化算法, 它是 2008 年由 Ruiqing Zhao 和 Wansheng Tang 首次提出的^[5]. 其突出的优点在于求解高维的优化问题时, 无需考虑函数是否可导, 只需要计算当前位置的伪梯度, 就可以确定爬过程中的搜索的方向. 并且该算法需调整的参数少、实现过程简单. 因此, 该算法被提出不久, 便受到了许多学者的研究和关注. 郝士鹏在 2010 年, 将混沌搜索方法与猴群算法结合, 提出了一种混沌猴群算法^[6]. 2013 年, 齐

① 基金项目: 陕西省自然科学基金研究计划(2014JM8325); 陕西省教育厅专项科研计划项目(14JK1538); 西安理工大学科技创新计划项目(2016CX013)

收稿时间: 2016-09-26; 收到修改稿时间: 2016-11-14 [doi:10.15888/j.cnki.csa.005822]

艳玉等^[7]引用了一种混沌映射算子初始化,且根据一类动态优化问题的模型,提出了一种求解动态优化问题的优化算法.同年,陈信等^[8]结合单纯形法思想,设计出了一种基于猴群算法和单纯形法的混合算法.但是,这些算法仍有不足之处,比如,算法的求解精度还需要作进一步的研究改进.

本文通过对基本猴群算法的分析,在猴群算法的初始化中,采用 Kent 混沌映射,设置参数,使得其值在区间上满足均匀分布.在算法的爬过程中,利用递减的因子作为爬步长给出了一种改进方法.

1 基本猴群算法

基本猴群算法的思想是^[5]:在猴子爬山过程中,模仿猴子的爬、望和跳几个有特点的动作,设计出相对应的搜索过程.其中,爬过程是通过逐步迭代,寻找猴子所在山峰的最高点,也就是局部最优解;望过程是通过张望来查看邻近区域是否存在比当前位置更优的解;跳过程主要是以全体猴子的重心为支点,从当前的搜索区域跳到新的搜索区域,从而使猴群跳出局部区域,避免在搜索的过程中陷入局部最优.

基本猴群算法主要分为解的表示和初始化,爬过程,望过程和跳过程四个过程^[5].

2 改进的猴群算法

在对基本猴群算法的应用中,我们发现算法的解初始化和爬过程对算法优化性能影响比较明显,因此这里在算法的解初始化和爬过程中分别给出了改进方法来提高算法的求解精度,其具体如下:

(1) 解初始化的改进

猴群初始化位置的设置往往会对最优结果产生一定的影响,初始化位置分布越均匀,越有利于搜索到最优结果^[9].混沌猴群算法^[6]在初始化时引用的是 Logistic混沌映射的方法,它的Lyapunov指数是 0.691. Lyapunov指数是一个衡量随着时间变化的不确定性的发散比率,发散率越大,分布越均匀^[10].为了使初始化位置分布更均匀,这里采用Kent混沌映射的方法来构造猴群的初始位置,它的Lyapunov指数是 0.696,大于Logistic混沌映射.而且在Kent混沌映射中,当参数 $\beta = 0.4$ 时, h_j 为 (0,1) 上的均匀分布.因此, Kent混沌映射比Logistic混沌映射有更好的遍历性和无重复性特点,其具体由如下函数来产生^[11]:

$$h_{j+1} = \begin{cases} h_j / \beta & 0 < h_j \leq \beta \\ 1 - h_j / (1 - \beta) & \beta < h_j \leq 1 \end{cases} \quad (1)$$

其中, h_j 为混沌变量, $h_j \in [0,1]$, $j = 1, 2, \dots, n$, β 为参数,当 $\beta = 0.4$ 时, h_j 为 (0,1) 上的均匀分布, x_{ij} 可由下式产生:

$$x_{ij} = x_{\min} + h_j \cdot (x_{\max} - x_{\min}) \quad (2)$$

其中, x_{ij} 为第 i 只猴子在第 j 维的具体位置, $[x_{\min}, x_{\max}]$ 为 x_{ij} 的取值范围.

因为 h_j 为 (0,1) 上的均匀分布,所以由式(2)可以使 x_{ij} 均匀分布于区间 $[x_{\min}, x_{\max}]$ 中.而初始化位置为均匀分布,有利于猴子进行下一步仔细地搜索,这在很大程度上避免了算法逃逸最优值.从而节省了搜索时间,减少了计算量,将会很快找到最优位置.因此有利于提高整个算法的精度,同时也加快了算法收敛到全局最优值的速度.

(2) 爬过程的改进

在基本猴群算法的爬过程中,爬步长决定算法搜索的精度和需要CPU的时间,爬步长值在一定范围内越小,算法搜索精度越高,相应需要CPU时间越长,反之亦然^[8].鉴于Sigmoid函数连续、可导、有界、光滑和具有严格单调性,是一类激励函数,它可将变量映射到[0,1]范围内,并且变量是在区间内逐渐变小^[12].因此,在该过程中引入Sigmoid函数得到渐变的爬步长,从而可在搜索初期保持种群多样性,在后期随着爬步长变小,局部搜索增强,加快收敛,这样就可使得求解精度与搜索速度达到一种平衡^[6].这里构造的递减因子具体如下:

$$a_{G+1} = \frac{1}{1 + e^{\frac{2ln100 - G}{nMax} - ln100}} \cdot \frac{nMax - G + 1}{nMax} \cdot a_G \quad (3)$$

其中, G 为当前的迭代次数, $nMax$ 为最大的迭代次数, a_G 为第 G 次迭代的爬步长.

综上,下面给出改进猴群算法求解步骤.

Step 1. 设定猴群算法的参数值,如种群大小 M ,总的迭代次数 $nMax$ 等参数.

Step 2. 利用(2)式给出种群中所有猴子的初始位置.计算出猴群中个体当前位置的目标函数值,找出当前最优位置及函数值并记录.

Step 3. 执行爬过程,利用(3)式将爬过程的爬步长由固定值改为可变的步长因子.计算爬过程后,如果有更好的目标函数值,则更新猴群位置到较好目标

函数值相对应的位置。

Step 4. 执行望过程, 在 $(x_{ij} - b, x_{ij} + b)$ 产生变量 y_{ij} , $j=1,2,\dots,n$, 搜索到新的猴群个体位置 $Y_i = (y_{i1}, y_{i2}, \dots, y_{in})$. 接着, 执行 Step 3.

Step 5. 执行跳过程, 计算 $x'_{ij} = x_{ij} + \text{round}(\theta |p_j - x_{ij}|)$, 其中 $j=1,2,\dots,n$, round 为取整函数, 计算猴群中个体当前位置的目标函数值, 如果有更好的目标函数值, 则更新猴群位置到较好目标函数值相对应的位置。

Step 6. 重复 Step 3-Step 5, 直至找到猴群中个体寻找到最高山峰或达到预定值 $nMax$, 算法结束, 输出最优位置和最优函数值。

3 数值仿真

为了说明本文所给的改进猴群算法(Improved Monkey Algorithm, IMA)的有效性, 这里选取如下典型测试函数进行求解。

1) Rastrigin 函数

$$f_1(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), \quad |x_i| \leq 5.12$$

2) Griewank 函数

$$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad |x_i| \leq 600 \quad (3)$$

3) Schwefel 函数

$$f_3(x) = \max\{|x_i|, i=1,2,\dots,n\}, \quad |x_i| \leq 100$$

以上这三个测试函数的理论最小值都为 0。在仿真中, 受到文献[5]的算法参数设置方法的启发, 并且经过多次试验, 将 IMA 的参数设置为: 种群规模 $M=10$, 总的迭代次数 $nMax=1000$, 爬步长 $a=0.1$, 爬步长的精度 1×10^{-6} , 爬的迭代次数 $Nc=10$, 视野长度 $b=0.5$, 望的迭代次数 $Nw=10$, 跳区间 $[c,d]=[-1,1]$ 。在函数均为 10 维的情况下, 对三个测试函数分别求解一次, 相应寻优过程曲线分别为图 1、图 2 和图 3 所示。

接着, 对上述三个测试函数分别利用基本猴群算法(Monkey Algorithm, MA)^[5]、差分进化算法(Differential Evolution Algorithm, DE)^[2]、粒子群算法(Particle Swarm Optimization, PSO)^[3]、遗传算法(Genetic Algorithm, GA)^[4]、混沌猴群算法(CMA)^[6]、改进混沌猴群算法(DCMA)^[7]和基于猴群算法和单纯形法的混合算法(SMMA)^[8]进行一次求解, 寻优曲线仍

分别如图 1、图 2 和图 3 所示。

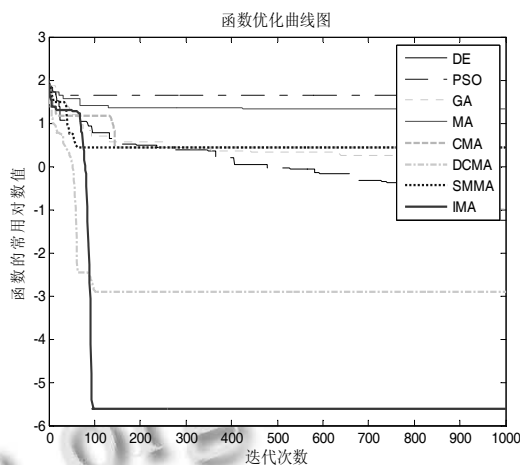


图1 对Rastrigin函数10维的寻优过程曲线

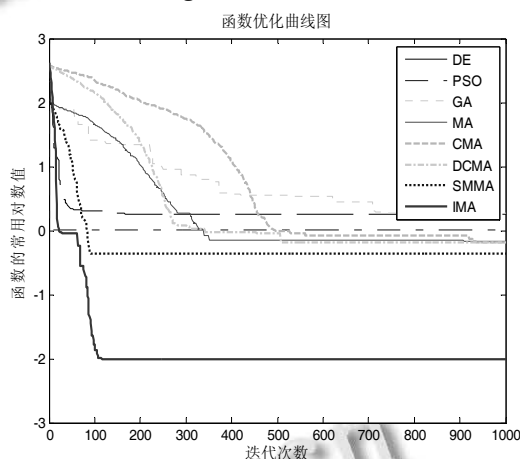


图2 对Griewank函数10维的寻优过程曲线

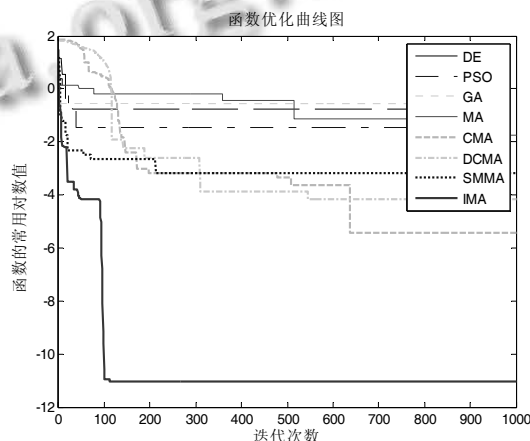


图3 对Schwefel函数10维的寻优过程曲线

现进一步利用上述八种算法对以上三个测试函数在 10 维的情况下, 分别进行 50 次求解, 它们的最差值、最优值、平均值和方差均罗列在表 1、表 2 和表 3 中。

表1 对 Rastrigin 函数 10 维的寻优结果

算法	最优解	最差解	平均值	方差
DE	1.01×10^{-2}	1.4775	0.2900	0.2270
PSO	11.285	52.554	35.899	181.38
GA	5.54×10^{-3}	0.1032	4.30×10^{-2}	1.85×10^{-3}
MA	11.673	24.263	17.976	20.617
CMA	0.03519	21.222	9.1158	90.406
DCMA	1.25×10^{-3}	3.54×10^{-3}	2.48×10^{-3}	7.64×10^{-7}
SMMA	2.7103	14.626	7.0101	25.022
IMA	1.35×10^{-7}	1.46×10^{-5}	5.49×10^{-6}	3.21×10^{-11}

表2 对 Griewank 函数 10 维的寻优结果

算法	最优解	最差解	平均值	方差
DE	0.7599	4.3888	2.063	1.319
PSO	1.0031	1.4921	1.1316	2.56×10^{-2}
GA	1.0687	2.886	1.7371	0.3484
MA	0.6076	0.8653	0.7506	4.59×10^{-3}
CMA	0.4193	0.8197	0.6932	1.75×10^{-2}
DCMA	0.4539	0.7814	0.6210	9.10×10^{-3}
SMMA	0.1695	0.7011	0.4673	2.88×10^{-2}
IMA	9.85×10^{-3}	9.83×10^{-2}	4.37×10^{-2}	8.68×10^{-4}

表3 对 Schwefel 函数 10 维的寻优结果

算法	最优解	最差解	平均值	方差
DE	2.15×10^{-4}	1.9755	0.3328	0.3437
PSO	8.84×10^{-3}	0.2788	8.92×10^{-2}	6.56×10^{-3}
GA	1.60×10^{-3}	4.61×10^{-2}	1.26×10^{-2}	2.71×10^{-4}
MA	4.16×10^{-4}	0.2067	3.78×10^{-2}	3.09×10^{-3}
CMA	2.65×10^{-5}	2.18×10^{-4}	7.09×10^{-5}	3.95×10^{-9}
DCMA	6.09×10^{-6}	8.28×10^{-3}	2.21×10^{-3}	7.60×10^{-6}
SMMA	8.42×10^{-6}	9.29×10^{-4}	4.70×10^{-4}	1.00×10^{-7}
IMA	7.72×10^{-15}	9.07×10^{-11}	8.13×10^{-13}	7.51×10^{-21}

为了进一步验证所给改进算法的有效性,在函数均为 30 维的情况下,做同上述一样的求解,求解一次的寻优过程曲线分别如图 4、图 5 和图 6 所示。接着对它们进行 50 次求解,所得结果均罗列在表 4、表 5 和表 6 中。

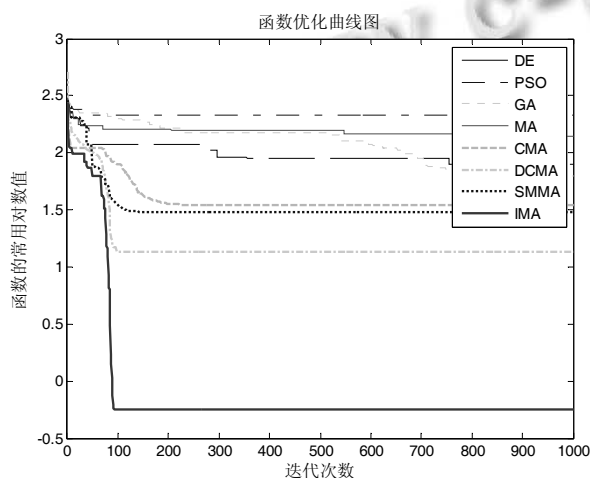


图4 对 Rastrigin 函数 30 维的寻优过程曲线

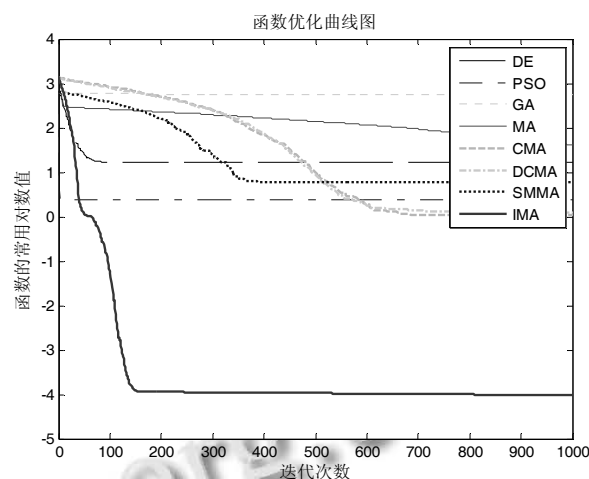


图5 对 Griewank 函数 30 维的寻优过程曲线

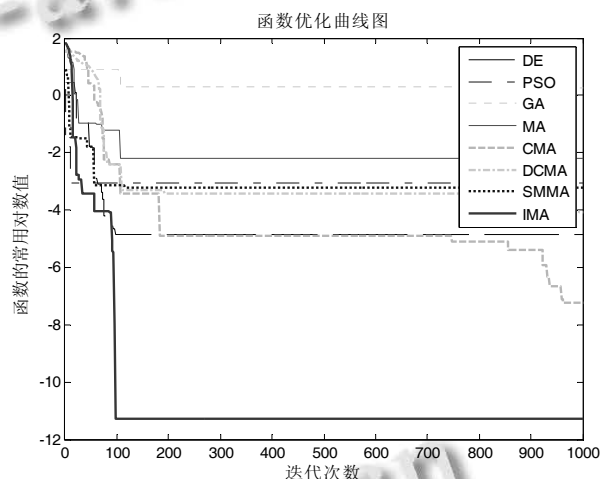


图6 对 Schwefel 函数 30 维的寻优过程曲线

表4 对 Rastrigin 函数 30 维的寻优结果

算法	最优解	最差解	平均值	方差
DE	53.532	79.792	67.24	101.59
PSO	36.671	214.25	141	5273.7
GA	32.504	80.857	59.645	366.63
MA	109.62	143.56	133.18	172.63
CMA	4.255	34.861	15.221	98.255
DCMA	0.565	13.619	6.082	21.085
SMMA	29.008	58.795	39.83	81.177
IMA	0.196	1.024	0.560	7.90×10^{-2}

表5 对 Griewank 函数 30 维的寻优结果

算法	最优解	最差解	平均值	方差
DE	13.684	43.597	27.674	102.53
PSO	0.9225	4.8599	1.7652	1.3761
GA	389.4	616.66	521.33	9443.5
MA	10.762	101.34	47.347	694.83

CMA	4.90×10^{-2}	1.102	1.004	7.58×10^{-2}
DCMA	1.185	6.833	1.746	2.230
SMMA	4.127	6.773	5.310	0.698
IMA	1.58×10^{-5}	1.30×10^{-4}	3.56×10^{-5}	1.66×10^{-7}

表 6 对 Schwefel 函数 30 维的寻优结果

算法	最优解	最差解	平均值	方差
DE	1.78×10^{-3}	0.4654	0.1251	2.34×10^{-2}
PSO	5.60×10^{-4}	0.3270	0.1467	1.01×10^{-2}
GA	4.97×10^{-3}	1.3663	0.3280	0.1318
MA	1.20×10^{-5}	3.33×10^{-4}	1.37×10^{-4}	8.58×10^{-9}
CMA	1.19×10^{-7}	6.85×10^{-7}	3.81×10^{-7}	4.36×10^{-14}
DCMA	5.40×10^{-5}	5.12×10^{-3}	2.15×10^{-3}	3.54×10^{-6}
SMMA	2.55×10^{-5}	1.23×10^{-3}	5.06×10^{-4}	1.68×10^{-7}
IMA	3.66×10^{-16}	3.68×10^{-11}	6.51×10^{-14}	1.65×10^{-26}

从以上结果可看出, 本文所给算法的优化性能明显得到改善, 求解精度得到提高.

4 结论

鉴于基本猴群算法易陷入局部最优, 求解精度不高的缺点, 文中改进了该算法解的初始化和爬过程阶段, 得到一种改进猴群算法. 然后利用对典型测试函数的寻优结果显示了所给改进算法的性能有明显的改善.

参考文献

1 王辉, 钱锋. 群体智能优化算法. 化工自动化及仪表, 2007, 34(5): 7-13.

2 周艳平, 顾幸生. 差分进化算法研究进展. 化工自动化及仪表, 2007, 34(3): 1-6.

3 Kennedy J, Eberhart RC. Particle swarm optimization. IEEE International Conference on Neural Networks. Perth, Australia. 1995, 4. 1942-1948.

4 Holland J. Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press, 1975.

5 Zhao RQ, Tang WS. Monkey algorithm for global numerical optimization. Journal of Uncertain Systems, 2008, 2(3): 165-176.

6 郝士鹏. 混沌猴群算法及其应用[硕士学位论文]. 天津: 天津大学, 2010.

7 齐艳玉. 离散猴群算法和混沌猴群算法及其应用[硕士学位论文]. 天津: 天津大学, 2012.

8 陈信, 周永权. 基于猴群算法和单纯法的混合优化算法. 计算机科学, 2013, 40(11): 248-254.

9 周刘喜, 张兴华, 李伟. 一种改进的多目标粒子群优化算法. 计算机工程与应用, 2009, 45(33): 38-41.

10 张伟伟, 武静, 马宏伟. 基于 Lyapunov 指数的超声导波检测技术. 振动、测试与诊断, 2015, 35(2): 250-257.

11 陶菊春, 吴建民. 可线性化非线性回归预测模型的剖析与改进. 数学的实践与认识, 2003, 33(2): 7-12.

12 张萧, 黄晞, 仲伟汉, 张亮. Sigmoid 函数及其导函数的 FPGA 实现. 福建师范大学学报(自然科学版), 2011, 27(2): 62-65.