

一种改进算的鸟群算法

李延延, 万仁霞

(北方民族大学 数学与信息科学学院, 宁夏 银川 750021)

摘 要: 针对鸟群算法(BSA)在警戒行为和飞行行为的模型设计中存在的不足, 本文在原算法的基础上, 为了增加鸟群警戒行为的靶向性, 在选取目标个体时, 采用了当前最优个体替代原算法中随机选取的方法; 为了克服飞行行为中生产者迭代步长过大而容易导致过度跳跃的不足, 采用步长加权平均思想予以改进. 通过对 12 种典型的不同类型基准函数进行实验测试, 结果表明改进策略对算法的全局搜索能力和优化精度都有较大的提高, 能达到更好的收敛速度和寻优精度.

关键词: 鸟群算法; 警戒行为; 飞行行为; 优化

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2018)09-0079-06

An Improved Algorithm for Bird Swarm Optimization

LI Yan-yan, WAN Ren-xia

(North MinZu University, Yinchuan, 750021 China)

Abstract: In this paper, two strategies are employed to overcome the shortcomings of the bird swarm algorithm (BSA): (1) in order to improving the search targeting of BSA vigilance behavior, the current optimal individual is adopted instead of random selection; (2) the weighted average step takes the place of the t-step in order to overcoming of sharp jump which originates from the iteration step-size of BSA flight behavior. Experimental results show that the proposed algorithm can improve both the global search ability and the optimization accuracy of BSA, and achieves a better convergence speed.

Key words: bird swarm; vigilance behavior; flight behavior; optimization

1 引言

优化问题是当今社会应用越来越广泛的一类问题, 如时间、空间、分类等的优化. 这些问题有连续的、离散的、线性的、非线性的和非凸的等. 对于一些连续可微的问题已经有传统且成熟的方法, 而对复杂问题, 如非凸的或不可微的问题传统的方法很难有效的解决. 对于这些很难解决的复杂问题, 近年来人们模仿自然生物进化提出了很多的仿生群体智能优化算法, 如遗传算法(GA)^[1]、粒子群算法(PSO)^[2]、布谷鸟搜索算法(CS)^[3]、磷虾群算法(KH)^[4]、鸡群算法(CSO)^[5]等. 这些经典的群体智能算法为求解全局最优化问题提供了有效的方法

途径.

鸟群算法(Bird Swarm Algorithm, BSA)是由 Xian-Bing Meng 等人^[6]于 2015 年提出来的一种基于鸟群行为的群体智能算法, 文献[6]中的实验表明鸟群算法在解决最优化问题时的性能明显优于粒子群算法和差分进化算法, 然而, 由于鸟群算法在警戒行为和飞行行为模型设计中存在的不足, 在解决高维多极值优化问题时容易陷入局部最优和过早收敛的问题.

本文针对原鸟群算法存在的问题进行改进, 对于鸟群的警戒行为迭代过程中, 通过选取当前时间段中的全局最优个体来改进原文中的随机选取策略; 对飞行行为中生产者的迭代步长采用加权平均

思想加以改进,避免原文中直接选取当前位置作为下一步迭代步长,后期步长跳跃过大而导致寻优不精确的缺点,改进后的算法更加逼近鸟群寻优行为,从而提高算法性能。

2 鸟群算法

鸟群算法是一种新型的启发式算法,是对鸟群的群体行为和群体智慧互动的一种简化,该算法模仿鸟群的三种基本行为:觅食行为、警戒行为和飞行行为,从而通过这种群体智能解决最优化问题。

觅食行为:每只鸟在觅食时借助自身及整个种群的经验进行觅食行为,可以通过以下更新公式表示:

$$x_{i,j}^{t+1} = x_{i,j}^t + (p_{i,j} - x_{i,j}^t) \times C \times \text{rand} + (g_j - x_{i,j}^t) \times S \times \text{rand} \quad (1)$$

式中, $j \in [1, \dots, D]$; rand 表示 $(0, 1)$ 之间独立均匀分布的随机数; C 和 S 分别是感知系数和社会加速系数;均为正数, $p_{i,j}$ 表示第 i 只鸟以往最佳的位置; g_j 表示整个群体共享的以往最佳位置。

警戒行为:鸟群都试图往种群中心移动,这时彼此之间发生不可避免的竞争都会对其移动产生影响和阻碍,因此每只鸟都不会顺利的直接往种群中心移动.警戒行为的更新公式如下:

$$x_{i,j}^{t+1} = x_{i,j}^t + A_1 (\text{mean}_j - x_{i,j}^t) \times \text{rand} + A_2 (p_{k,j} - x_{i,j}^t) \times \text{rand} \quad (2)$$

$$A_1 = a_1 \times \exp(-\frac{p\text{Fit}_i}{\text{sumFit} + \epsilon} \times N)$$

$$A_2 = a_2 \times \exp((\frac{p\text{Fit}_i - p\text{Fit}_k}{|p\text{Fit}_k - p\text{Fit}_i| + \epsilon}) \times \frac{N \times p\text{Fit}_k}{\text{sumFit} + \epsilon})$$

式中, $k(k \neq i)$ 为 $[1, N]$ 之间的一个随机正整数; a_1 和 a_2 是 $[0, 2]$ 之间的两个常数; $p\text{Fit}_i$ 表示第 i 只鸟的最佳适应度值; sumFit 表示整个种群的最佳适应度值总和; ϵ 为一个很小的常量用来避免零分误差; mean_j 表示整个种群平均位置的第 j 个元素. 当一只鸟往种群中心移动时会不可避免的和其他鸟发生竞争,通过种群的平均适应度值刻画当鸟往种群中心移动时由周围环境引起的间接影响,用 A_1 来刻画这种间接影响. A_2 表示鸟往种群中心移动时由某个特定的干扰引起的直接影响。

若个体 $k(k \neq i)$ 适应度值优于第 i 个个体的适应度,这将意味着第 i 个个体可能遭受了比第 k 个个体更大的干扰. 尽管存在一定的随机性和不确定性,个体 k 也将更加可能的往种群中心移动。

飞行行为:鸟群由于受外界的掠夺、觅食或其他

干扰而飞行到其他地方. 当到达一个新环境后鸟群可能会再次寻找食物,一些鸟扮演生产者的角色来搜寻食物块,另外一些会消费这些有生产者寻觅而来的食物块. 这里的生产者和乞食者通过规则 4 来划分,生产者和乞食者行为可以通过一下迭代公式描述:

$$x_{i,j}^{t+1} = x_{i,j}^t + \text{randn}(0, 1) \times x_{i,j}^t \quad (3)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + (x_{k,j}^t - x_{i,j}^t) \times \text{FL} \times \text{rand} \quad (4)$$

这里 $\text{randn}(0, 1)$ 表示均值为 0, 标准偏差为 1 的高斯分布, $k \in [1, \dots, N], k \neq i$, FL ($\text{FL} \in [0.5, 0.9]$) 表示乞食者跟随生产者寻找食物. 为了方便,假设鸟群飞行到其他地方的频率为 FQ , FQ 是一个正整数。

其算法流程如下

```

1: 输入:  $N$  种群规模;  $M$  最大迭代次数;  $\text{FQ}$  飞
2:   行频率;  $P$  觅食概率;  $C, S, a_1, a_2, \text{FL}$  为五个常
3:   量参数;  $t = 0$ ; 初始化种群和定义相关参数; 计
4:   算  $N$  个个体的适应度值并找出最优解;
5: While ( $t < M$ )
6:   If ( $t \% \text{FQ} \neq 0$ )
7:     For  $i = 1: N$ 
8:       If  $\text{rand} < P$ 
9:         鸟采用公式(1) 觅食;
10:      Else
11:        鸟采用公式(2) 警戒;
12:      End if
13:    End for
14:  Else
15:    将鸟群分为两部分: 生产者和乞食者;
16:    For  $i = 1: N$ 
17:      If  $i$  为生产者
18:        采用公式(3);
19:      Else
20:        采用公式(4);
21:      End if
22:    End for
23:  End if
24:    计算新的适应度值;
25:    如果新的适应度值比以往的优, 则更新掉;
26:    找出当前最优值;
27:  $t = t + 1$ ;
28: End while
29: 输出: 目标函数最优值的个体

```

3 改进的鸟群算法

3.1 算法改进

原鸟群算法在设计过程中,对于警戒行为的位置更新公式,其直接影响部分中个体 k 的选取过于

随机,若选取的个体 k 比当前个体 i 的适应度差,即选取成乞食者,这样反而会将降低算法的寻优能力,增加寻优过程的震荡性. 对于飞行行为中直接采用当前个体的位置作为下一步迭代的步长,没有考虑之前步长的影响,这时容易导致在下一次迭代时跳跃过大,因此不能严格而准确的寻找最优位置. 对于原算法存在的这些不足,本文对其做以下两方面改进.

3.1.1 警戒行为中随机个体 k 的选取

为了克服警戒行为中因为随机选取个体可能带来的震荡性,本文以寻优为靶向,设计选取当前时间段全局最优个体加以改进. 即对公式(2) 改进如下:

$$\begin{aligned} x_{i,j}^{t+1} &= x_{i,j}^t + A_1(\text{mean}_j - x_{i,j}^t) \times \text{rand} \\ &\quad + A_2(p_{\text{gFit},j} - x_{i,j}^t) \times \text{rand} \\ A_1 &= a_1 \times \exp(-\frac{p\text{Fit}_i}{\text{sumFit} + \epsilon} \times N) \\ A_2 &= a_2 \times \exp((\frac{p\text{Fit}_i - p\text{Fit}_{\text{gFit}}}{|p\text{Fit}_{\text{gFit}} - p\text{Fit}_i| + \epsilon}) \frac{N \times p\text{Fit}_{\text{gFit}}}{\text{sumFit} + \epsilon}) \end{aligned} \tag{5}$$

这里 gFit 表示当前时间段全局最优适应度值个体.

3.1.2 飞行行为中生产者步长的改进

为了克服原算法因选择前期步长可能带来的位置变化的跳跃性,本文折中历史步长的影响,通过对历史步长加权取均值的方法来代替原算法的前期步

长的选择,即对原算法位置公式(3) 做如下改进:

$$\begin{aligned} x_{i,j}^{t+1} &= x_{i,j}^t + \text{randn}(0,1) \\ &\quad \times \frac{\sum_{i=1}^t |x_{i,j}^t - x_{i-1,j}^t| (x_{i,j}^t - x_{i-1,j}^t)}{\sum_{i=1}^t |x_{i,j}^t - x_{i-1,j}^t|} \end{aligned} \tag{6}$$

通过加权平均的思想对迭代步长加以改进,避免直接选取当前个体的位置作为下一步迭代的步长而导致跳跃过大出现的寻优误差.

3.2 算法步骤

改进后的算法依然借助 2.3 算法流程,在第 11 行鸟群的警戒行为中采用改进后的公式(5),第 18 行鸟群飞行行为中生产者步长迭代公式采用改进后的公式(6),其他步骤不变.

4 算法实验与比较

4.1 测试函数

为验证本文提出的改进鸟群算(IBSA))的有效性 and 可靠性,我们选取 12 个具有不同特点的基准函数进行测试,如表 1 所示,这些函数包含单峰、多模态、高维和低维等情况,并将 IBSA 算法实验结果与经典的启发式算法 PSO(粒子群算法)和 BSA(鸟群算法)的实验结果进行比较.

表 1 测试函数

测试函数	取值范围	全局最优值
$F_1 = \sum_{i=1}^n x_i^2$	$[-100,100]$	0
$F_2 = \sum_{i=1}^{n-1} (x_i^2)(x_{i+1}^2+1) + (x_{n+1}^2)(x_i^2+1)$	$[-1,4]$	0
$F_3 = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$	$[-10,10]$	0
$F_4 = \sum_{i=1}^n (e^{-0.2} \sqrt{x_i^2 + x_{i+1}^2} + 3(\cos(2x_i) + \sin(2x_{i+1})))$	$[-35,35]$	-3.917 275
$F_5 = \sum_{i=1}^{n-1} [-(x_{i+1} + 47) \sin \sqrt{ x_{i+1} + x_i/2 + 47 } - x_i \sin \sqrt{ x_i - (x_{i+1} + 47) }]$	$[-512,512]$	959.64
$F_6 = -\sin^3(5\pi x) \exp(-(\frac{5(x-0.1)}{4})^2)$	$[-1,1]$	-1
$F_7 = [\frac{1}{n} \sum_{i=1}^n x_i - (\prod_{i=1}^n x_i \frac{1}{n})^2]$		0
$F_8 = -\frac{1}{n} \sum_{i=1}^n x_i \sin \sqrt{ x_i }$	$[-500,500]$	-418.983
$F_9 = \frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	$[-5,5]$	-78.332
$F_{10} = -0.0001 [\sin(x_1)\sin(x_2) 100 - [(x_1^2 + x_2^2)]^{\frac{0.5}{\pi}} +1 ^{0.1}]$	$[-10,10]$	-2.062 612 18
$F_{11} = -\cos(x_1)\cos(x_2) \exp[-(x_1 - \pi)^2 - (x_2 - \pi)^2]$	$[-100,100]$	-1
$F_{12} = \sqrt{ \cos \sqrt{ x_1^2 + x_2^2 } } + 0.01(x_1 + x_2)$		-0.184 67

4.2 算法参数设计

在本次实验中,群体规模均为 30,维度大小为 20,每个函数独立运行 100 次,最大迭代次数设为 1 000. 涉及的算法均由 MATLAB R2014a 完成,运行在一台配置 Intel(R)Celeron(R)CPU G3900@2.80 GHz2. 80 GHz 的电脑上,操作系统是 Windows7.

本文从取得最好和最差的全局最优值(评价算法在多次运行中能达到的最优和最差值)、平均值(评价

算法所能达到的精度)、方差(评价算法的稳定性和鲁棒性)等指标来测试算法的有效性,测试的相关参数及结果见表 2、3.

表 2 算法相关参数

算法	相关参数
PSO	$c_1=c_2=1.4944\ 5, w=0.729$
BSA	$C=S=1.5, a_1=a_2=1, P\in[0.8, 1], FL\in[0.5, 0.9], FQ=3$
IBSA	$C=S=1.5, a_1=a_2=1, P\in[0.8, 1], FL\in[0.5, 0.9], FQ=10$

表 3 函数测试结果

函数	算法	最好	最差	平均	方差
f_1	PSO	0.008 527 449	0.169 372 533	0.053 117 927	0.034 849 329
	BSA	0	0	0	0
	IBSA	0	0	0	0
f_2	PSO	1.097 0e-54	1.744 5e+02	4.676 3e+00	2.435 7e+01
	BSA	3.489 6e-12	3.746 3e+04	1.513 0e+03	5.443 3e+03
	IBSA	1.051 6e-24	1.006 4e+02	1.634 4e+00	1.052 7e+01
f_3	PSO	1.689 7e-11	3.456 1e-01	6.840 5e-01	2.864 1e-01
	BSA	1.857 4e-09	9.493 1e-01	4.840 5e-02	1.774 1e-01
	IBSA	1.509 7e-18	1.463 1e-01	5.072 8e-03	2.187 9e-02
f_4	PSO	-2.600 6e+00	-6.139 4e-01	-2.580 8e+00	1.986 7e-01
	BSA	-2.600 6e+00	-1.744 9e+00	-2.572 6e+00	1.462 3e-01
	IBSA	-2.600 6e+00	-2.600 6e+00	-2.600 6e+00	1.331 3e-13
f_5	PSO	-2.506 5e+02	-6.684 4e+01	-1.610 2e+02	5.801 8e+01
	BSA	-2.271 2e+02	-1.239 0e+02	-2.197 6e+02	2.607 2e+01
	IBSA	-2.271 2e+02	-2.271 2e+02	-2.271 2e+02	5.182 6e-13
f_6	PSO	-1.000 0e+00	-7.796 2e-01	-9.978 0e-01	2.203 8e-02
	BSA	-1.000 0e+00	-1.000 0e+00	-1.000 0e+00	6.027 9e-08
	IBSA	-1.000 0e+00	-1.000 0e+00	-1.000 0e+00	0
f_7	PSO	1.298 8e-25	3.089 6e-01	9.971 6e-03	4.836 9e-02
	BSA	0	4.962 2e-02	9.924 4e-04	7.017 6e-03
	IBSA	0	1.400 4e-08	3.545 1e-10	2.030 6e-09
f_8	PSO	-3.181 7e+00	-3.181 7e+00	-3.181 7e+00	1.344 2e-15
	BSA	-3.181 7e+00	-3.181 7e+00	-3.181 7e+00	4.394 7e-07
	IBSA	-3.181 7e+00	-3.181 7e+00	-3.181 7e+00	1.339 0e-15
f_9	PSO	-3.916 6e+01	-3.916 6e+01	-3.916 6e+01	9.283 6e-14
	BSA	-3.916 6e+01	-3.916 6e+01	-3.916 6e+01	3.497 6e-06
	IBSA	-3.916 6e+01	-3.916 6e+01	-3.916 6e+01	9.283 6e-14
f_{10}	PSO	-2.070 5e+00	-2.070 5e+00	-2.070 5e+00	2.231 6e-03
	BSA	-2.070 5e+00	-2.070 0e+00	-2.070 5e+00	7.828 8e-05
	IBSA	-2.070 5e+00	-2.070 5e+00	-2.070 5e+00	2.323 0e-15
f_{11}	PSO	-1.000 0e+00	-3.408 8e-31	-8.400 1e-01	3.684 4e-01
	BSA	-1.000 0e+00	-9.455 8e-01	-9.958 3e-01	9.003 0e-03
	IBSA	-1.000 0e+00	-1.000 0e+00	-1.000 0e+00	7.526 1e-11
f_{12}	PSO	-1.932 5e+00	-1.006 7e+00	-1.526 0e+00	2.361 4e-01
	BSA	-1.973 2e+00	-1.915 0e+00	-1.948 9e+00	1.517 2e-02
	IBSA	-1.977 1e+00	-1.931 4e+00	-1.974 6e+00	9.913 2e-03

为更加直观的展现本算法的性能,我们绘制了以上三种群体智能算法比较有代表性的收敛曲线。

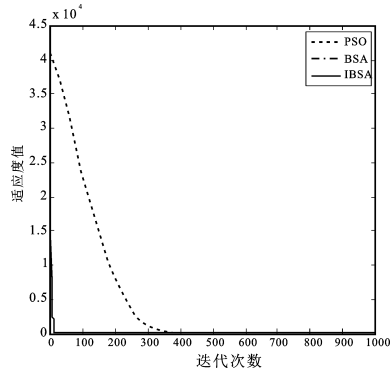


图 1 对 f_1 的寻优效果图

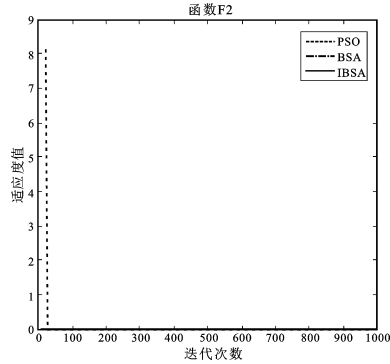


图 2 对 f_2 的寻优效果图

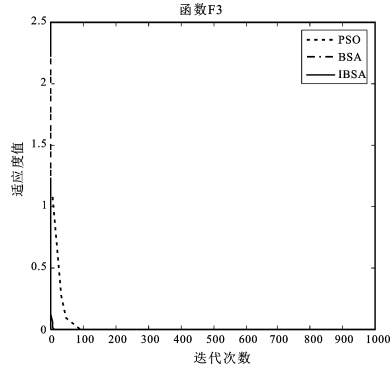


图 3 对 f_3 的寻优效果图

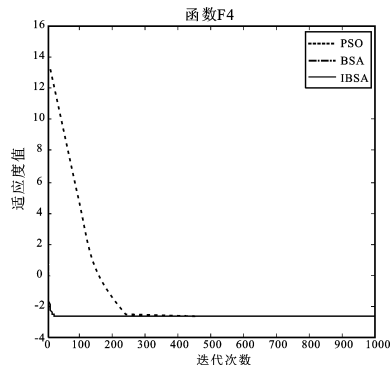


图 4 对 f_4 的寻优效果图

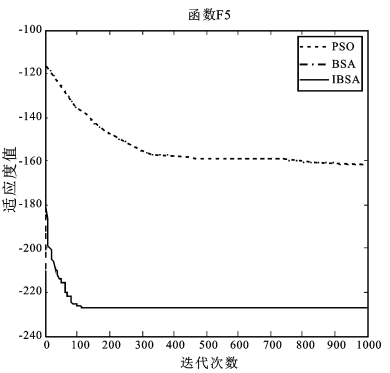


图 5 对 f_5 的寻优效果图

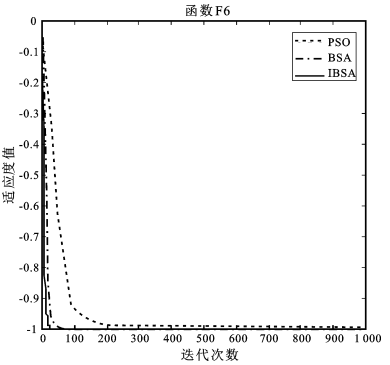


图 6 对 f_6 的寻优效果图

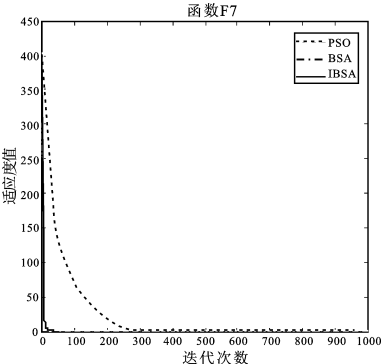


图 7 对 f_7 的寻优效果图

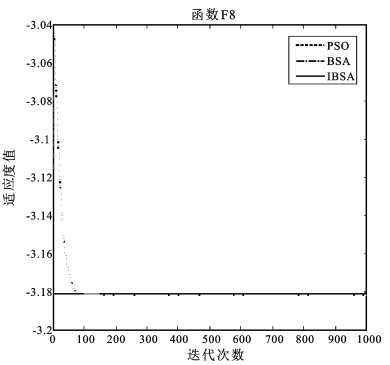
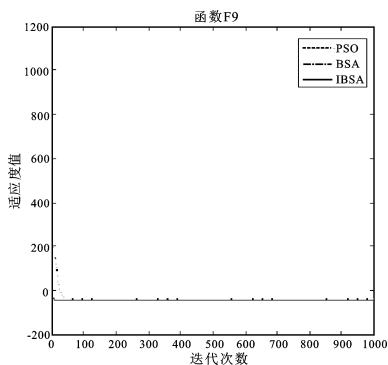
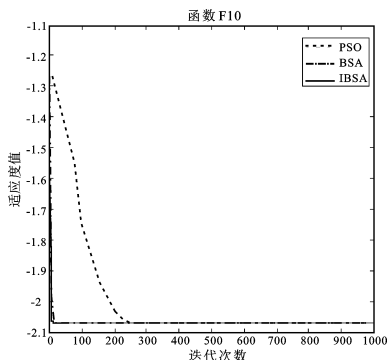
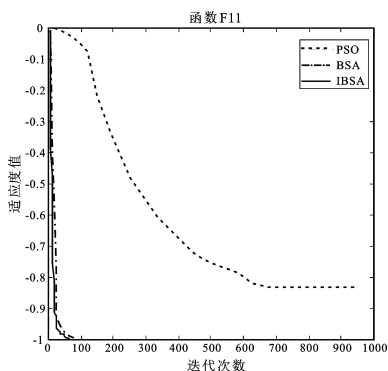
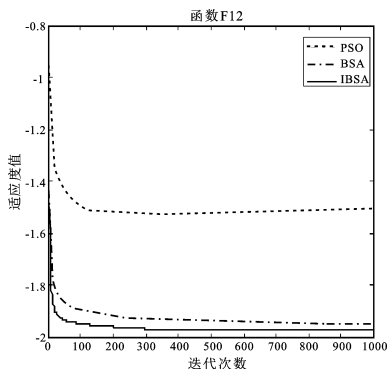


图 8 对 f_8 的寻优效果图

4.3 实验结果分析

比较实验结果中的最优值、最差值、均值和方差,我们可以看出,对于具有不同特点的单模和多模

图 9 对 f_9 的寻优效果图图 10 对 f_{10} 的寻优效果图图 11 对 f_{11} 的寻优效果图图 12 对 f_{12} 的寻优效果图

函数,本文提出的改进算法(IBSA)在寻优能力、收敛速度和解的稳定性方面较优于其他两种算法。算法具有较强的鲁棒性,精度较高。

从测试图中我们可以进一步看出,总体上,IBSA和BSA的寻优效果明显优于PSO,但IBSA前期的收敛速度又明显快于BSA。在某些情况下,如测试函数 f_5 和 f_{12} ,IBSA可以取得比BSA更好的寻优效果。

实验表明,相比于BSA算法和PSO算法,IBSA具有极强的局部勘探能力,能有效避免陷入局部最优,同时,不管是对单峰函数还是多峰函数,IBSA算法都能较快的找到函数的最优近似解且达到稳定,在最终求解精度即寻优能力方面也优于上述两个算法,表明IBSA算法具有较强的全局和局部搜索能力。

5 结束语

为克服鸟群算法易陷入局部最优、早熟收敛的问题,本文对鸟群的警戒行为和飞行行为做了调整改进。通过数值实验对比表明,本文提出的算法在收敛能力和稳定性均优于原算法。

参考文献:

- [1] Kuo H C, Lin C H. A directed genetic algorithm for global optimization[J]. Applied Mathematics & Computation, 2013, 219(14):7348-7364.
- [2] Rezaee J A R, Jasni J. Parameter selection in particle swarm optimization: a survey[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2013, 25(4):527-542.
- [3] Yang X S, Deb S. Cuckoo search: recent advances and applications[J]. Neural Computing & Applications, 2014, 24(1):169-174.
- [4] Gandomi A H, Alavi A H. Krill herd: a new bio-inspired optimization algorithm [J]. Communications in Nonlinear Science and Numerical Simulation, 2012, 17(12):4831-4845.
- [5] Meng X, Liu Y, Gao X, et al. A new bio-inspired algorithm: chicken swarm optimization[J]. Springer International Publishing, 2014, 8794:86-94.
- [6] Meng X B, Gao X Z, Lu L, et al. A new bio-inspired optimisation algorithm: Bird Swarm Algorithm[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2016, 28(4):673-687.

作者简介:

李延延 男,(1990-),硕士研究生.研究方向为数据挖掘与模式识别。

万仁霞(通讯作者) 男,(1975-),博士,副教授,硕士生导师.研究方向为智能计算、数据挖掘、模式识别等。

E-mail:wanrx1022@126.com.