

基于二次飞行和随机扰动的改进蝙蝠算法

赵乃刚, 李 勇

(山西大同大学 数学与计算机科学学院, 山西 大同 037009)

摘 要: 针对基本蝙蝠算法存在着不易跳出局部寻优、搜索精度低等问题, 提出了二次飞行和随机扰动的改进策略。每只蝙蝠对自身位置进行随机扰动后, 对上代速度采用自适应学习权重, 使蝙蝠能够趋好避坏地进行飞行搜索; 为了保证种群多样性, 对种群中最差的一部分蝙蝠进行惩罚, 舍弃对上代速度的学习, 进行二次飞行搜索, 提高算法的全局搜索能力。通过对七个典型不同类型的基准函数进行实验测试, 结果表明了改进策略对算法的优化精度和全局搜索能力有很大的提高, 使改进的算法拥有更好的收敛速度和寻优精度。

关键词: 蝙蝠算法; 二次飞行; 随机扰动; 自适应学习权重

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2017)05-0021-05

An Improved Bat Algorithm Based on Second Flight and Random Disturbance

ZHAO Nai-gang, LI yong

(School of Mathematics and Computer Science, Shanxi Datong University, Datong 037009, China)

Abstract: In order to overcome the shortcomings of conventional the basic bat algorithm (BA), such as easily trapping in local optima and lower search accuracy, this paper proposed the improvement strategy based on the second flight and the random disturbance. After each bat makes a random disturbance to its position. Using the adaptive weight learning the previous flight speed, so that bats can search in a good direction to avoid a bad direction. In order to ensure the diversity of the population, there is a punishment of the improved algorithm for the worst part to carry out a second flight search without learning last speed. This improves the global search ability of the improved algorithm. The algorithm tested on seven distinct types of benchmark functions. The results show that the improved strategy has a great improvement on the optimization accuracy and global search ability. The proposed algorithm has a better convergence rate and optimization accuracy.

Key words: bat algorithm(BA); second flight; random disturbance; adaptive learning weight

1 引言

蝙蝠算法(BA)是 Yang^[1]模拟蝙蝠觅食行为提出的一种新型算法。该算法设置参数少、实现方式简单,已成功应用于一些优化问题中^[2-5]。然而与诸多启发式算法相似,蝙蝠算法在求解高维复杂优化问题时不易跳出局部寻优等缺陷。近年来,国内外学者逐渐展开对蝙蝠算法的收敛性改进和应用仿真等问题研究。Baziar^[6]等人提出了一种能够自适应

调整回声响度和蝙蝠位置的改进方法,王文等人^[7]提出了采用机动飞行来改进蝙蝠算法,赖锦辉^[8]将多种学习机制引入蝙蝠算法中来提高算法收敛效率,李雅梅等人^[9]利用 Powell 机制改进蝙蝠算法,都在一定程度上提高了算法性能。

我们以标准蝙蝠算法的理论研究为框架,提出了个体随机扰动、速度自适应权重和二次飞行的改进策略,构造出带随机扰动的自适应蝙蝠算法(RWBA)和带二次飞行和随机扰动的自适应蝙蝠

收稿日期: 2016-10-09; **修回日期:** 2016-11-11

基金项目: 国家自然科学基金(61272095);国家自然科学基金青年基金(41401521);山西大同大学科学研究项目(2016K1)

算法(TFRWBA),实验仿真结果表明,与基本 BA、MFBA^[7]相比,个体随机扰动、速度自适应权重策略对算法的局部开发能力具有非常好的改进效果,二次飞行策略提高的种群多样性,使算法的全局搜索能力得到了提高。

2 基本蝙蝠算法

蝙蝠一般在夜间活动,在黑暗中发出声音脉冲,根据接收到的回声来探索食物的位置、避开障碍物及寻找栖息地。随着与搜索猎物的距离减小,蝙蝠发出声音的响度从最高响度减小到靠近猎物时的静音。蝙蝠通过发出和接收回声的时间差,来确定与目标的距离、方位及移动速度。将蝙蝠的寻找食物过程理想化如下:

(1)所有的蝙蝠都能够利用回声定位机制去感知目标距离,并且能以一种巧妙的方式区分食物与障碍物。

(2)蝙蝠在位置 X_i 以速度 V_i 随意飞行,以固定频率 f_i 、可变化波长 λ 和响度 A_0 搜索食物;同时根据目标与自身距离来调节发射出的脉冲波长(或频率),在靠近食物时调整发射脉冲的频度 $\gamma \in (0,1)$ 。

(3)响度的变化是从最大值 A_0 到最小值 $A_{\min}$ 。

由此概括蝙蝠算法的更新规则:设蝙蝠在 d 维空间中搜索食物,在 $t-1$ 时刻,蝙蝠 i 的位置和飞行速度分别为 X_i^{t-1} 和 V_i^{t-1} 全局最优位置为 X_* ,则蝙蝠 i 在 t 时刻的位置和飞行速度如下:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (1)$$

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X_*)f_i \quad (2)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (3)$$

对于局部搜索,在 $[0,1]$ 之间产生一个随机数 rand1,如果 $\text{rand1} > r_i^t$,则对当前最优解按式(4)进行随机扰动,产生一个新解 X_{new} ,并对新的解进行越界处理。再产生一个随机数 rand2,如果 $\text{rand2} > A_i^t$ 且 $f(X_{\text{new}}) < f(X_*)$,则蝙蝠 i 移向 X_{new} ,同时按式(5)和(6)进行调整脉冲响度 A_i^t 和速率 r_i^t ,使蝙蝠从发现猎物开始降低其脉冲发射的响度 A_i^t ,同时提高其脉冲发射的速率 r_i^t 。

$$X_{\text{new}} = X_* + \epsilon A^t \quad (4)$$

$$A_i^t = \alpha A_i^{t-1} \quad (5)$$

$$r_i^t = r_i^0 [1 - e^{-\gamma}] \quad (6)$$

式中: $f_{\min}$ 和 $f_{\max}$ 分别是蝙蝠发出声波的最小和最大频率; β 为 $[0,1]$ 之间的均匀随机数;每只蝙蝠发射声波的频率初始值在范围 $[f_{\min}, f_{\max}]$ 内均匀随机分布; ϵ 为 $[-1,1]$ 中的随机数; $A^t = \langle A_i^t \rangle$ 为全体蝙蝠在迭代次数为 t 时的平均响度; r_i^0 为初始速率; α

和 γ 均为常数, $0 < \alpha < 1, 0 < \gamma < 1$ 。

3 采用二次飞行和随机扰动的改进蝙蝠算法

3.1 个体随机扰动和运动速度的自适应学习权重

在 BA 算法中,蝙蝠个体前期随机分布在解空间的各个位置,食物的位置根据当前蝙蝠个体所在位置计算所得,但是随着迭代过程的进行,蝙蝠个体的位置和食物的位置会趋于同性,这样对于蝙蝠群个体之间的进行信息交流以及蝙蝠群个体和食物之间进行信息交流的意义变得越来越小;所以我们在新一代种群运动中对当前位置进行随机扰动(式(7)、(8));同时为了更好的学习前代速度,我们加入速度权重 w_i^t ,运动后的位置如果比之前位置适应值好,说明此次运动速度使得该蝙蝠距离食物更近,通过加大速度学习权重,来提高对当前速度的学习程度;反之则减小速度学习权重,降低对当前速度的学习程度(式(9))。以此来提高蝙蝠趋好避差的习性,其中 $\beta \in [0,1]$ 为随机数。

$$V_i^t = w_i^t * V_i^{t-1} + (\beta * X_i^{t-1} - X_*)f_i \quad (7)$$

$$X_i^t = \beta * X_i^{t-1} + V_i^t \quad (8)$$

$$w_i^t = \begin{cases} 1.25 * w_i^{t-1} & \text{if } f(X_i^t) \leq f(X_i^{t-1}) \\ 0.75 * w_i^{t-1} & \text{else} \end{cases} \quad (9)$$

3.2 二次飞行

随着迭代的进行,蝙蝠群可能聚集到局部极值附近,种群多样性降低,不利于种群跳出局部极值,出现过早收敛。种群多样性决定了全局搜索能力,为了保证种群多样性,提出了二次飞行策略,每次更新后,把最差的一部分蝙蝠(文中取 $40 * 15\% = 6$ 只蝙蝠)作为外出搜寻小队,让他们进行二次飞行,作为对它们的惩罚,而此次飞行不再学习上代的飞行速度,单一的背向目前最优食物位置飞行,即:

$$V_i^t = (X_i^{t-1} - X_*)f_i \quad (10)$$

3.3 算法流程

TFRWBA 算法基本流程如下:

Step1:初始化算法参数,在给定空间中随机生成初始种群蝙蝠的位置 X_i ,同时计算相应蝙蝠的适应度,随机产生脉冲发射的速率 R_0 和声音响度 A_0 ,并找出最优位置 X_* 。

Step2:蝙蝠 i 根据式(7)、(8)来更新速度和位置,计算适应值。判断 $f(X_i^t) < f(X_i^{t-1})$ 是否成立,按式(9)来更新速度学习权重。

Step3:如果随机生成的数 rand1 大于蝙蝠 i 的脉冲速率 r_i^t ,蝙蝠 i 根据式(4)在最优解附近产生一个局部

解 X_{new} , 如果 $f(X_{\text{new}}) < f(X_*)$, 则更新全局最优位置.

Step4: 若随机生成的数 rand2 小于蝙蝠 i 的脉冲响应度 A_i^t , 且 $f(X_{\text{new}}) < f(X_*)$, 则蝙蝠 i 移向 X_{new} , 同时按式((5) 减小蝙蝠 i 的脉冲响应度 A_i^t 和式(6) 增大蝙蝠 i 的脉冲速率 r_i^t .

Step5: 对种群按适应值的大小进行排序, 找出当前最优位置 X_* , 同时对最差的一部分蝙蝠(15%), 按式(10)、(3) 进行二次飞行更新位置, 更新全局最优位置.

Step6: 判断是否满足给出的终止条件, 若满足, 输出全局最优位置 X_* 及最优值 $f(X_*)$; 否则转到步骤 Step2.

4 仿真实验与分析

为了检验提出策略的合理性和有效性, 我们将

表 1 测试函数

测试函数	取值范围	最小值	函数名
$f_1 = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$	0	Sphere
$f_2(x) = \sum_{i=1}^n \{x_i^2 - 10\cos(2\pi x_i) + 10\}$	$[-5.12, 5.12]^n$	0	Rastrigin
$f_3(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$[-30, 30]^n$	0	Rosenbrock
$f_4(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	$[-32, 32]^n$	0	Ackley
$f_5(\vec{x}) = \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^n$	0	Griewank
$f_6(x) = -\sum_{i=1}^n (x_i \sin(\sqrt{ x_i }))$	$[-500, 500]^n$	$-418.9829 * n$	Schwefel
$f_7(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]^n$	0	Shifted Schwefel 1, 2

从表 2 和图 1 中看出, TFRWBA 算法从最优值、最差值、均值和方差四个方面来说, 它都远远优于 BA 和 MFBA 算法, 而且从方差和曲线进化图可以发现 TFRWBA 算法的具有非常好的稳定性; 由于没有扩大多样性, 不带二次飞行的 RWBA 算法在 $f_1 f_2 f_4 f_5 f_7$ 函数上比 TFRWBA 算法结果相差不大, 寻优结果和稳定性都是远远优于 BA 和 MFBA 算法; 对于极难极小化的病态非凸单模 Rosenbrock 函数 f_3 和非常多局部极小点的多峰函数 f_6 , TFRWBA 算法找到的最优值都有明显优势, 而 RWBA 算法除了在求解 f_3 函数的结果稍好于 MFBA 算法, 在求解 f_6 函数的结果却差于 MFBA, 究其原因, 还是因为随机扰动和自适应学习权重策略只是使算法的局部寻优能力得到了极大的开发, 所以引

构造两种算法, 一种是采用随机扰动和自适应权重的蝙蝠算法(RWBA), 另一种是采用二次飞行和随机扰动的自适应权重蝙蝠算法(TFRWBA), 同时与标准蝙蝠算法(BA) 和采用机动飞行的蝙蝠算法(MFBA)^[7] 分别在 7 个经典基准函数(表 1) 上进行测试. 在数值实验过程中, 四种算法的初始种群采用了一般的随机生成方式, 粒子群的粒子数目设置为 40, 高维函数的维数统一选择 100 维, 算法的最大迭代次数为 1 000, $A_i^0 = R_i^0 = \text{rand}$, $f_{\min} = 0$, $f_{\max} = 2$, $\alpha = \gamma = 0.9$, 评价算法的性能从最优值、最差值、平均值、方差四个方面进行了分析. 每种算法进行 30 次寻优运算, 每个函数的数值结果如表 2 所示, 三十次函数平均寻优曲线如图 1 所示.

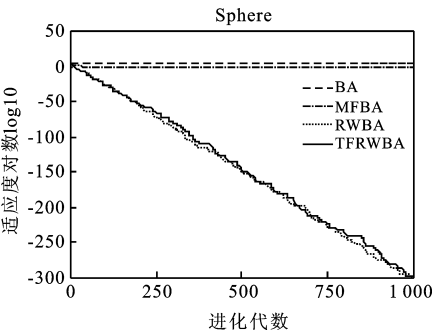
入的二次飞行正好开发了算法的全局搜索能力. 由于二次飞行增加了一部分时间, 但和非常好的寻优结果相比, 还是值得的.

5 结束语

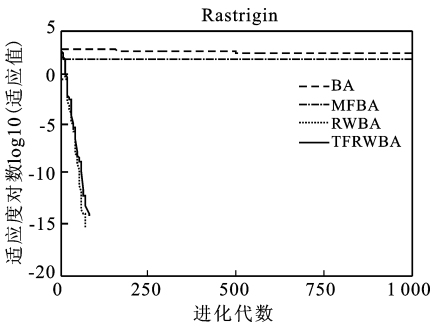
本文针对标准蝙蝠算法在求解全局优化问题的过程中不易跳出局部寻优和精度差的缺点, 提出了一种采用二次飞行和随机扰动的改进蝙蝠算法, 通过与已知的 BA 算法和 MFBA 算法在七个测试函数上的比较可以看出, 对于易于寻优的函数 TFRWBA 算法能够找到更高精度的解; 对于不易寻优的单峰和多峰函数, TFRWBA 算法同样能有效地避免陷入局部最优找到较好解, 在全局搜索和局部开发上表现出显著优势.

表 2 各种算法性能的测试结果

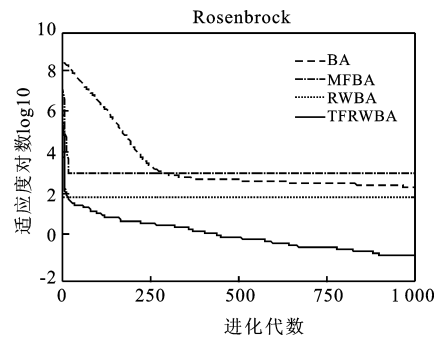
函数	算法	Min	Max	Mean	Std
f_1	BA	1.07E-01	1.86E+04	3.17E+03	4.88E+03
	MFBA	1.07E-03	4.74E+00	8.73E-01	1.33E+00
	RWBA	0	2.40E-298	8.02E-300	0
	TFRWBA	0	7.18E-297	2.39E-298	0
f_2	BA	3.79E+02	6.75E+02	5.24E+02	7.47E+01
	MFBA	8.21E-01	4.24E+02	1.29E+02	1.24E+02
	RWBA	0	0	0	0
	TFRWBA	0	0	0	0
f_3	BA	1.01E+02	1.88E+03	3.36E+02	4.41E+02
	MFBA	1.88E+00	1.16E+04	1.60E+03	3.20E+03
	RWBA	9.89E+01	9.90E+01	9.90E+01	1.27E-02
	TFRWBA	5.73E-04	1.68E+00	1.65E-01	3.11E-01
f_4	BA	2.00E+01	2.00E+01	2.00E+01	3.40E-04
	MFBA	1.07E-01	1.15E+01	3.55E+00	3.51E+00
	RWBA	8.88E-16	8.88E-16	8.88E-16	0
	TFRWBA	8.88E-16	8.88E-16	8.88E-16	0.00E+00
f_5	BA	5.29E+02	1.55E+03	1.00E+03	2.27E+02
	MFBA	3.78E-09	8.85E-03	7.55E-04	1.74E-03
	RWBA	0	0	0	0
	TFRWBA	0	0	0	0
f_6	BA	-21 110.243 82	-15 710.150 24	-18 707.868 69	1 378.433 471
	MFBA	-41 898.287 92	-30 054.431 74	-41 108.635 56	3 004.862 598
	RWBA	-27 793.115 03	-16 826.751 49	-21 269.133 38	2 671.538 514
	TFRWBA	-41 898.284 61	-41 898.253 17	-41 898.273 9	6.94E-03
f_7	BA	1.64E+04	1.11E+05	5.00E+04	2.40E+04
	MFBA	2.53E+04	1.23E+05	6.05E+04	2.01E+04
	RWBA	0	2.12E-296	7.07E-298	0
	TFRWBA	0	4.30E-298	1.67E-299	0



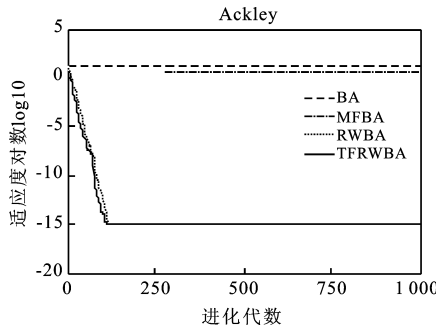
(a) f_1 函数



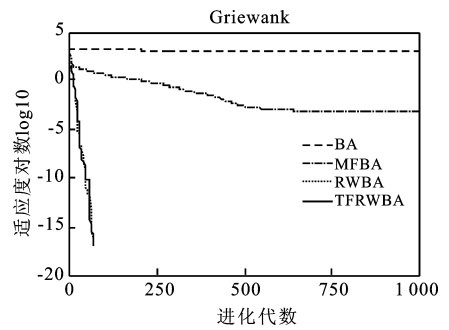
(b) f_2 函数



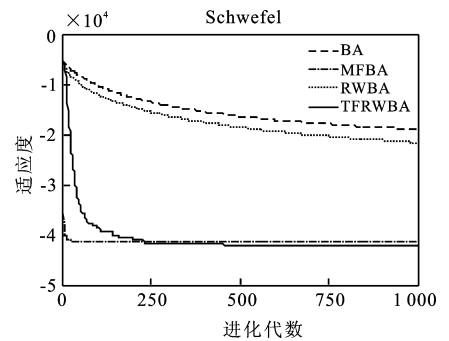
(c) f_3 函数



(d) f_4 函数



(e) f_5 函数



(f) f_6 函数

图 1 各算法在不同测试函数上的平均进化曲线

参考文献:

[1] Yang Xinshe. A new metaheuristic bat—inspired algorithm[C]//Nature Inspired Cooperative Strategies for Optimization. Spain, Granada, 2010:65-74.

[2] Yang Xinshe, Gandomi Amir Hossein. Bat algorithm: a novel approach for global engineering optimization [J]. Engineering Computation, 2012,29(5):464-483.

[3] Komarasamy G, Wahi A. An optimized K-means clustering technique using bat algorithm [J]. European Journal of Scientific Research, 2012,84(2):263-273.

[4] Gandomi A H, Yang Xin she, Alavi A H, et al. Bat algorithm for constrained optimization tasks [J]. Neural Computing and Applications, 2013, 22 (6): 1239-1255.

[5] Fister I, Rauter S, Yang Xinshe, et al. Planning the sports training sessions with the bat algorithm [J]. Neurocomputing, 2015,149(2):993-1002.

[6] Bazar A, Kavooosi fard A, Zare J. A novel self adap-

tive modification approach based on bat algorithm for optimal management of renewable MG[J]. Journal of Intelligent Learning Systems and Applications, 2013,5 (1):11-18.

[7] 王文, 王勇, 王晓伟. 采用机动飞行的蝙蝠算法[J]. 计算机应用研究, 2014,31(10):2962-2989.

[8] 赖锦辉. 基于部落结构的多种学习机制蝙蝠优化算法 [J]. 计算机应用研究, 2015,32(2):364-367.

[9] 李雅梅, 曹益华. 基于 Powell 机制的改进蝙蝠算法 [J]. 微电子学与计算机, 2015,32(3):73-76.

作者简介:

赵乃刚 男,(1975-),硕士研究生,讲师. 研究方向为数据挖掘. E-mail:2821499953@qq. com.

李 勇 男,(1974-),博士. 研究方向为智能算法及无线网络.