

文章编号:1007-6735(2014)01-0026-05

基于 Fuch 映射的混沌蝙蝠算法

孙文捷, 张惠珍, 张 健, 赵 坤

(上海理工大学 管理学院, 上海 200093)

摘要: 为了改善基本蝙蝠算法(bat-inspired algorithm, BA)的求解性能, 提高其搜索效率, 避免其搜索过程陷入局部最优, 利用 Fuch 映射对基本蝙蝠算法的局部最优解的邻域和蝙蝠的频率变化区间进行混沌遍历搜索, 提出了一种新型混合蝙蝠算法——Fuch 混沌蝙蝠算法(FCBA). 仿真计算结果表明: 与 BA 相比, FCBA 具有较好的收敛性能, 能够较快地收敛于测试算例的全局最优解.

关键词: 蝙蝠算法; Fuch 映射; 混沌优化
中图分类号: TP 301.6 **文献标志码:** A

Chaos Bat-inspired Algorithm Based on Fuch Mapping

SUN Wen-Jie, ZHANG Hui-zhen, ZHANG Jian, ZHAO Kun

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In order to improve the solving performance of the original bat-inspired algorithm (BA), increase its searching efficiency, and avoid falling into local optimal solution, the neighborhood of the local optimum and the frequency interval in the original BA were optimized by using the chaos optimization method based on Fuch mapping. Furthermore, a new hybrid bat-inspired algorithm named as Fuch chaos bat-inspired algorithm (FCBA) was proposed. The numerical results show that FCBA has better convergence performance, and can converge faster to the optimal solution of numerical examples.

Key words: bat-inspired algorithm; Fuch mapping; chaos optimization

近十几年内炙手可热的遗传算法、神经网络、模拟退火算法、蚁群算法、微粒群算法等^[1], 都是受自然规律和生物群体智能行为的启发而提出, 其在广泛的科学和工程技术领域内显示了其独特的能力和应用效果. 但是, 这类算法在求解复杂问题时, 也暴露出其

固有的一些缺陷, 如算法易陷入局部极值, 求解精度不高, 而且许多算法的理论基础较薄弱, 没有形成统一的算法框架, 仍有许多问题有待研究. 受蝙蝠回声定位行为的启发, Yang^[2]于 2010 年提出一种新型的元启发式算法——蝙蝠算法(bat-inspired algorithm

收稿日期: 2013-04-08
基金项目: 上海市大学生创新训练基金资助项目(201210252057); 上海市高校青年教师培养计划资助项目(slg12010); 高等学校博士学科点专项科研基金联合资助项目(20123120120005); 上海市教委科研创新资助项目(14YZ090); 上海理工大学国家级培育课题资助项目(13XGQ07)
第一作者: 孙文捷(1992-), 男, 本科生. 研究方向: 系统工程. E-mail: 2268605086@qq.com
通讯作者: 张惠珍(1979-), 女, 讲师. 研究方向: 运筹学, 智能优化. E-mail: huizhenzhang@usst.edu.cn

BA). 已有研究表明,BA 在某些方面将粒子群算法、遗传算法和和声算法的主要优点进行了良好的结合,并且粒子群算法和和声算法可以认为是蝙蝠算法在经过适当简化后的一种特殊情况. 因此,BA 较其它算法具有发挥更大作用的潜能^[2-3]. 混沌是一种普遍的非线性现象,具有遍历性、随机性与确定性相统一、对初始值变化敏感等特点^[4]. 由于遍历性可使搜索过程避免陷入局部极小,因此,混沌搜索已成为一种非常有效的优化算法. 针对传统混沌优化方法中优化结果对搜索初始值要求极高以及搜索效率较低的问题,傅文渊等^[5]提出一种自适应折叠混沌优化方法——Fuch 映射,与 Logistic 映射^[6]、Chebyshev 映射^[7]和 Tent 映射^[8]相比,Fuch 映射具有更强的混沌特性. 针对基本蝙蝠算法搜索效率低和较易陷入局部最优的缺点,本文在基本蝙蝠算法中引入了 Fuch 映射,设计了一种基于 Fuch 映射的蝙蝠算法——Fuch 混沌蝙蝠算法(FCBA). 算法循环过程中:一方面,通过混沌遍历频率变化区间,使得蝙蝠的速度能得到充分变化;另一方面,在蝙蝠所发射的脉冲速率还不太高时,利用 Fuch 映射对局部最优解的邻域进行混沌遍历搜索,使其跳出局部最优解. 通过求解基准测试函数对 FCBA 与 BA 的寻优能力和搜索效率进行比较. 结果表明,与基本蝙蝠算法相比,FCBA 具有全局搜索能力强和收敛速度快的优点.

1 蝙蝠算法

1.1 蝙蝠的回声定位能力

蝙蝠是一种神奇的动物,有高级的回声定位能力. 微型蝙蝠靠一种声纳,也称为回声定位器,来探测猎物,避免障碍物,在黑暗中找到它们的栖息地. 这些蝙蝠发出响亮的声音脉冲,然后聆听从周围的物体反弹回来的回声,利用双耳的时间差及回声的响度变化去建立周围环境的三维场景.

大多数蝙蝠用短波、调频信号对一个音阶横扫,而另一些蝙蝠则更经常使用固定频率的定位信号. 它们的信号带宽变化取决于物种,并经常通过使用更多谐波来提高,但是它们探测猎物和避免障碍的原理都是基于回声定位的声学原理. 研究显示:蝙蝠发出的脉冲频率通常为 25~150 kHz. 由声音在空气中的速度 $v = 340$ m/s 及超声波在频率 f 下的波长 $\lambda = v/f$ 可知:蝙蝠发出的脉冲波长 λ 在 2~14 mm 之间,这样的波长类同于它的猎物的大小. 此外,蝙蝠发出在超声波范围内的声波,其响度能达到 110 dB,且响度可以从搜索猎物时的最高变化到靠近猎物时的最小.

1.2 蝙蝠运动的数学描述

1.2.1 蝙蝠的速度更新和位置更新

假设搜索空间为 D 维,第 i 只蝙蝠在第 t 次进化时的位置和速度分别为 x_i^t 和 v_i^t ,则在第 $t+1$ 次进化时,其位置和速度可分别更新为 x_i^{t+1} 和 v_i^{t+1} ,即

$$F_i = F_{\min} + (F_{\max} - F_{\min})\beta \quad (1)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x^*)F_i \quad (2)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3)$$

式中, F_i , $F_{\max}$ 和 $F_{\min}$ 为第 i 只蝙蝠在当前时刻发出的声波的频率、声波频率的最大值和最小值; β 为随机数, $\beta \in [0, 1]$; x^* 为当前最优解.

对于大小为 n 的蝙蝠群体,可以从中选择一只蝙蝠(解),并更新该蝙蝠相应的位置,即在被选择解的附近产生一个新解

$$x_i^{\text{new}} = x_{\text{old}} + \epsilon A^t \quad (4)$$

该过程可被理解为局部搜索过程. 其中, x_{old} 为从当前最优解集中随机选择的一个解, A^t 为当前代前 i 只蝙蝠的平均响度; ϵ 为随机向量 $\epsilon \in [-1, 1]^D$.

1.2.2 响度和脉冲发射

蝙蝠实际捕猎过程中,其声波响度 $A(i)$ 随着与猎物距离的减小而不断减弱,但脉冲发射速率 $R(i)$ 随着与猎物距离的减小而逐渐提高. 蝙蝠 i 脉冲的响度 $A(i)$ 和发射速率 $R(i)$ 可更新为

$$A^{t+1}(i) = \alpha A^t(i)$$

$$R^{t+1}(i) = R^0(i)[1 - \exp(-\gamma t)] \quad (5)$$

其中, $0 < \alpha < 1$ 和 $\lambda > 0$ 均为常量. $A(i) = 0$ 时意味着蝙蝠 i 刚刚发现一只猎物,暂时停止发出任何声音. 不难发现 $t \rightarrow \infty, A^t(i) \rightarrow 0, R^t(i) = R^0(i)$.

上述现象反映在实际算法中,即随着响度的不断减弱和脉冲速率的不断提高,蝙蝠以一定的概率进行局部搜索和接受新解;但是,进行局部搜索的概率随着脉冲速率的不断提高而降低,接受新解的概率随着响度的不断减弱而降低. 这既反映了蝙蝠的实际捕猎情况,也有助于加快算法收敛,并在算法运行初期能够跳离局部最优.

基本蝙蝠算法中,蝙蝠的速度和位置的更新策略和标准粒子群算法更新速度和位置的策略有些相似之处. 虽然从某种程度上来讲,蝙蝠算法可视为标准粒子群优化和由响度和脉冲速率控制的集中局部搜索的一种均衡组合. 但是,与其它仿生类智能优化相似,基本蝙蝠算法仍有容易陷入局部最优的不足之处.

2 Fuch 映射

Fuch 映射是一种新型的一维离散映射,其表达式为

$$x_{n+1} = \cos(1/x_n^2) \quad (6)$$

其中, $x_n \neq 0, n \in \mathbb{Z}^+$. 文献[5]对 Fuch 映射的不动点特性和 Lyapunov 指数^[9]进行了深入研究,并计算得到 Fuch 映射的 Lyapunov 指数 λ 为 2.165. 与传统的混沌映射(如 Logistic 映射、Chebyshev 映射和 Tent 映射)相比, Fuch 映射有如下优势^[5]:

a. Fuch 映射具有更强的混沌特性,给定微小的初始值变化,混沌映射产生的输出是完全不同的,并且系统输出毫无规律;

b. 映射相比于 Logistic 映射和 Tent 映射在解空间内具有更加均衡的遍历性;

c. 考察遍历整个系统解空间的混沌迭代次数时, Fuch 映射与传统有限折叠混沌映射相比具有更小的迭代次数,能更好地实现混沌寻优;

d. Fuch 映射的不动点为超越数,若初始值不为 0,则该映射产生的混沌不收敛于有理数不动点,表明在初始值不为 0 的情形下均能产生混沌.

Fuch 映射有不因初值设置不当而陷入不动点(局部最优)的优点. 因此,将 Fuch 映射引入基本蝙蝠算法,其可以不依赖蝙蝠初始搜索的精度,对局部最优解的邻域进行混沌遍历搜索.

3 基于 Fuch 映射的蝙蝠算法

3.1 频率变化区间的混沌搜索

在基本的蝙蝠算法中,每只蝙蝠利用特有的回声定位能力,测算出自己与当前所求得的最优解间的距离,并根据自动发出的脉冲频率来调节自己的速度. 但是脉冲频率往往是从 $[F_{\min}, F_{\max}]$ 中随机取得,这很有可能使速度的变化落在某个局部的区间中,根据式(3)可知:特别是在蝙蝠很靠近所搜索到的当前最优解时,速度会被迫“停滞”,这就影响了蝙蝠的搜索效率. 另一方面,在蝙蝠的脉冲发射速率不断升高后,进行邻域搜索的机会将越来越小,这使得蝙蝠对所搜索到的初始解的依赖性加大,这很有可能使算法陷入局部最优.

为了改进基本蝙蝠算法的上述不足之处,本文定义第 i 只蝙蝠在第 t 次进化时产生混沌变量 μ_i^t ,并利用其对频率变化区间进行混沌搜索

$$F_i = F_{\min} + (F_{\max} - F_{\min}) \mid \mu_i^t \mid \quad (7)$$

与式(1)相比,式(7)使得蝙蝠在靠近当前最优解的同时,其速度依然能得到充分变化.

3.2 局部最优解邻域的混沌搜索

基本蝙蝠算法中的局部搜索为:从现有的最佳解决方案中选择一只蝙蝠后,每只蝙蝠的新位置在随机游走中就近产生. 为了进一步改善基本蝙蝠算

法的局部搜索功能,改善其容易陷入局部最优的不足之处,使其能够较快地求出所求问题的最优解或满意解,本文利用 Fuch 映射对基本蝙蝠算法的局部最优解邻域进行混沌搜索,具体搜索步骤如下:

Step1 产生随机数 $rand$, 若 $rand > R_i$, 则进行局部搜索, 否则不进行局部搜索; f_{new} 为蝙蝠初始位置所相应的目标函数值;

Step 2 确定以当前最优解 x^* 为中心的邻域搜索半径

$$\rho = \min(x^* - low, upper - x^*)$$

式中, low 和 $upper$ 分别为变量的上下界.

Step 3 令第 t 次进化中第 i 只蝙蝠所产生的混沌变量 μ_i^t 为本次邻域搜索的初始混沌变量 k_0 ; 记总迭代次数为 m , 临时变量为 $temp$, x_i^t 为第 i 只蝙蝠第 t 次进化时所搜索到的新解. 以最小化问题为例,混沌遍历搜索过程为

```

fnew = f(x_i^t);
if rand > R_i
    k_0 = μ_i^t;
    for n = 1:m
        k_n = cos(1/k_{n-1}^2);
        temp = x^* + ρ * k_n;
        if f(temp) < fnew
            x_i^t = temp; fnew = f(x_i^t);
        end
    end
end
end

```

3.3 Fuch 混沌蝙蝠算法

为了使基本蝙蝠算法能够跳出局部最优,通过较小地迭代次数搜索到全局最优解,本文不仅利用 Fuch 映射对频率变化区间和局部最优解邻域进行混沌搜索,而且在每次进化迭代开始前对 Fuch 映射的初值进行赋值扰动,得到一种新型蝙蝠算法——Fuch 混沌蝙蝠算法,简记为 FCBA.

假设求函数 $f(x)$ 的最小值,算法最大循环次数为 gen ,蝙蝠种群大小为 n ,第 i 只蝙蝠的位置为 x_i ,速度为 v_i ,脉冲发射速率为 R_i ,脉冲响度为 A_i ,个体适应值 $fitness(i) = f(x_i)$. 用于求解 $f(x)$ 的 FCBA 算法的基本步骤可以概括如下:

```

for t = 1:gen
    对混沌变量赋初值: μ_i^t = rand;
    for i = 1:n
        对蝙蝠的频率变化区间进行混沌搜索,更新其速度和位置,并对蝙蝠的速度与位置进行越界处理;
        fnew = f(x_i^t);
        if rand > R_i

```

```
对局部最优解的邻域进行混沌遍历搜索
end if
if rand>A_i&&fnew<fitness(i)
接受由局部搜索产生的新解为  $x_i^t$ ;
fitness(i) = fnew;
end if
更新当前最佳解  $x^*$  及其对应的速率、脉冲发
射速率、脉冲响度和脉冲频率;
end
增大  $R_i$ ,减小  $A_i$ ;
end
邻域搜索半径  $\rho$  的确定方法已经可以避免位
置越界的发生,因此上述 FCBA 算法中只对蝙蝠的
初始搜索进行位置纠偏.
```

4 仿真实验

为测试算法的求解性能,本文利用基本蝙蝠算

法和 FCBA 算法对 4 个经典函数优化问题进行求
解.为了确保两种算法具有可比性,使它们在相同的
起点上进行比较,计算过程中,不仅对两种算法共有
的绝大部分参数设定了相同的值(如表 1 所示),而
且对两种算法设置了相同的迭代次数,但对其求解
精度没有设置.

表 1 参数设置

Tab.1 Setting up of parameters value

参数	取值	参数	取值
种群大小	20	α	0.95
最大进化代数	50	β	0.9
速度区间	[4,2]	频率范围	[0,2]
初始响度	$A_0=0.25$	初始脉冲发射速率	$\gamma_0=0.5$

表 2 给出了 4 个测试函数及其变量取值区间和
全局最优解.除函数 $f_4(x)$ 为单模函数,无局部最优
解,只有全局最优解外,其它 3 个函数都是典型的多
峰函数,有多个极小值,均较难优化,常用来测试算
法的全局寻优能力.

表 2 4 个标准测试函数

Tab.2 4 Benchmark functions

测试函数	搜索空间	最优目标函数值	理论全局最优解
$\min f_1(x) = -0.5 + \frac{(\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5)}{(1 + 0.001 \times (x_1^2 + x_2^2))^2}$	$x_1 \in [-4,4] \quad x_2 \in [-4,4]$	-1	[0,0]
$\min f_2(x) = -20 \times e^{-0.2 \times \sqrt{\frac{x_1^2 + x_2^2}{2}}} - e^{\frac{\cos(2\pi x_1) + \cos(2\pi x_2)}{2}} + 20 + e$	$x_1 \in [-32,32] \quad x_2 \in [-32,32]$	0	[0,0]
$\min f_3(x) = 20 + x_1^2 + x_2^2 - 10 \times (\cos(2\pi x_1) + \cos(2\pi x_2))$	$x_1 \in [-5,5] \quad x_2 \in [-5,5]$	0	[0,0]
$\min f_4(x) = x_1^2 + x_2^2 + x_3^2$	$x_1 \in [-32,32] \quad x_2 \in [-32,32] \quad x_3 \in [-32,32]$	0	[0,0]

算法运行环境为 Windows 764 位操作系统,
Intel 处理器,2.40 GHZ,4 GB 内存;仿真软件为
MatlabR2012a.

表 3 和表 4(见下页)分别给出了 4 个测试函数
利用基本蝙蝠算法和 FCBA 算法的求解结果.

对比表 3 和表 4 中的数据,不难发现:FCBA 与
BA 求解 4 个测试函数时,所耗费的计算时间相差并
不很大,但 FCBA 在寻优精度上却有着较大优势,尤
其求解 $f_2(x)$, $f_3(x)$ 和 $f_4(x)$ 时,FCBA 在寻优精
度上的优势特别明显,其求解结果在最好值、最劣值

和平均值 3 方面均明显优于基本蝙蝠算法的计算
结果.

给定相同的迭代次数,两种算法所耗费的计算
时间相差并不大.为了进一步分析算法的收敛性能,
图 1(见下页)给出了两种算法求解 4 个测试函数的
迭代收敛曲线,图 1 中(a),(b),(c)和(d)的横坐标
表示进化迭代次数 gen ,纵坐标表示函数值 $f(x)$.

从图 1 可知,与 BA 相比,FCBA 能够较快地收
敛于全局最优解,其在较小的进化迭代次数内便可
接近所测函数的理论全局最优解.

表 3 FCBA 的算法寻优结果

Tab.3 Results obtained by using FCBA

测试函数	最好值	最差值	平均值	最短时间/s	最长时间/s	平均时间/s
$f_1(x)$	-0.999 999 90	-0.999 000 00	-0.999 980 00	0.009 240 00	0.019 157 69	0.010 100 00
$f_2(x)$	0.002 904 56	0.010 282 28	0.005 401 00	0.006 444 34	0.023 846 63	0.007 650 00
$f_3(x)$	0.000 008 39	0.168 654 36	0.006 430 00	0.002 781 29	0.008 026 99	0.003 701 08
$f_4(x)$	0.000 000 36	0.008 834 38	0.000 239 00	0.040 426 16	0.082 553 84	0.057 705 11

表 4 BA 的算法寻优结果
Tab.4 Results obtained by using BA

测试函数	最好值	最差值	平均值	最短时间/s	最长时间/s	平均时间/s
$f_1(x)$	-0.990 284 09	-0.983 370 80	-0.990 150 00	0.000 346 00	0.002 440 04	0.000 915 00
$f_2(x)$	13.969 339 98	14.707 438 00	13.991 320 00	0.000 356 21	0.001 130 22	0.000 578 00
$f_3(x)$	7.959 697 15	8.251 772 37	7.965 758 00	0.000 254 87	0.001 992 74	0.000 737 16
$f_4(x)$	2.170 721 94	27.828 346 60	12.715 550 00	0.000 791 54	0.003 273 49	0.001 372 85

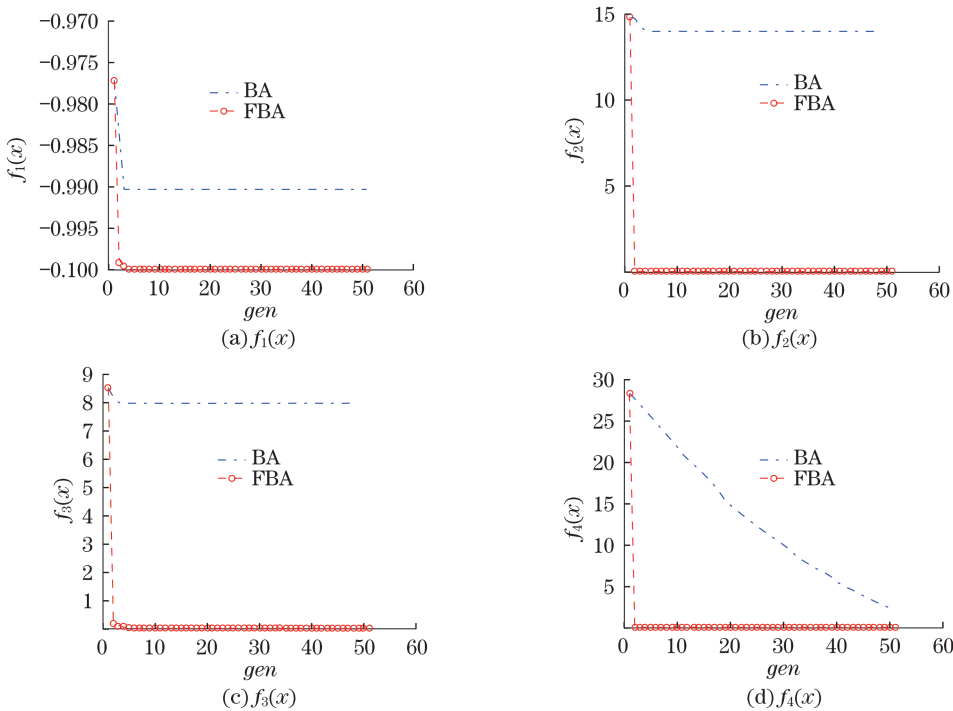


图 1 目标函数随循环次数的变化曲线
Fig.1 Changing curve of the objective function value

5 结 论

将 Fuch 映射引入到基本蝙蝠算法中,设计了一种混沌蝙蝠算法.算法中,一方面,在蝙蝠的脉冲发射速率还不是很高,即蝙蝠还没有很靠近猎物时,对局部最优解的邻域进行混沌遍历搜索,大大优化了初始值的质量与逼近全局最优解的速度;另一方面,通过对局部最优解的邻域进行混沌遍历搜索,不仅有效地改善了蝙蝠初始搜索所得到的解,而且对基本蝙蝠算法的局部搜索功能进行了很好的改进,使其在较小的进化迭代次数内较快地收敛于全局最优解或求得问题的满意解.仿真结果表明:提出的混沌蝙蝠算法求解函数优化问题时,在收敛速度和精度上均优于基本蝙蝠算法,且具有一定的可行性和较好的寻优能力.

参考文献:

[1] 王凌.智能优化算法及其应用[M].北京:清华大学出

版社,2001.
[2] Yang X S. A new metaheuristic bat-inspired algorithm [J]. Nature Inspired Cooperative Strategies for Optimization,2010,284:65-74.
[3] Yang X S, Gandomi A H. Bat algorithm: a novel approach for global engineering optimization [J]. Engineering Computation,2012,29(5):267-289.
[4] 李兵,蒋慰孙.混沌优化方法及其应用[J].控制理论与应用,1997,14(4):613-615.
[5] 傅文渊,凌朝东.自适应折叠混沌优化方法[J].西安交通大学学报,2013,47(2):33-38.
[6] 范九伦.分段 Logistic 混沌映射及其性能分析[J].电子学报,2009,37(4):720-725.
[7] 石军.基于 Chebyshev 映射的混沌特性及其性能分析[J].现代电子技术,2008(23):93-96.
[8] 单梁,强浩,李军,等.基于 Tent 映射的混沌优化算法[J].控制与决策,2005,20(2):179-182.
[9] 李小亭,蔡诗东.混沌和李亚普诺夫特征指数[J].物理,1996,25(5):282-285.

(编辑:金虹)