

DOI:10.19431/j.cnki.1673-0062.2019.01.012

一种改进的混合蝙蝠算法

郜振华, 吴昊

(安徽工业大学 管理科学与工程学院, 安徽 马鞍山 243000)

摘要:为解决基本蝙蝠算法中存在的易陷入局部最优且求解精度不足的问题,提出一种改进的混合蝙蝠算法,引入了分组迭代模式和多种速度迭代公式加强了全局搜索能力,更新了种群领域搜索公式的基础上引用了t分布作为种群最优解的领域搜索方式,补足了蝙蝠算法的局部搜索能力,避免了算法陷入局部最优解。通过多个标准测试函数的实验验证改进的混合蝙蝠算法能有效解决基本蝙蝠算法中出现的问题。

关键词:蝙蝠算法;混合算法;分组迭代

中图分类号:TP18 **文献标志码:**A **文章编号:**1673-0062(2019)01-0062-05

An Improved Hybrid Bat Algorithm

GAO Zhenhua, WU Hao

(Institute of Management Science and Engineering, Anhui University of Technology,
Maanshan, Anhui 243000, China)

Abstract: In order to solve the problem that the basic bat algorithm is easy to fall into the local optimal and the precision of solving is insufficient, an improved hybrid bat algorithm is proposed, the packet iteration mode and various velocity iteration formulas are introduced to strengthen the global search ability, and the domain search method of t distribution as the optimal solution of the population is referenced on the basis, the local search ability of BAT algorithm is supplemented to avoid the algorithm falling into the local optimal solution. Through the experiment of several standard test functions, it is proved that the improved hybrid bat algorithm can effectively solve the problems in the basic bat algorithm.

key words: bat algorithm; hybrid algorithm; group iteration

收稿日期:2018-11-21

作者简介:郜振华(1972-),男,教授,主要从事工业工程方面的研究. E-mail:1013823212@qq.com

0 引言

蝙蝠算法(bat algorithm, BA)是 X. S. Yang 教授于 2010 年提出的^[1],通过模拟蝙蝠在自然界觅食的过程来实现算法的迭代与寻优过程。和粒子群优化算法(particle swarm optimization algorithm, PSO)一样,蝙蝠算法也是基于群体的随即搜索机制,而区别在于蝙蝠算法的随机性更强,因此蝙蝠算法具有收敛速度快、鲁棒性好的优点^[2]。从蝙蝠算法提出至今,主要应用于求解函数优化问题上,少数学者用其解决离散问题。盛孟龙^[3]等人针对蝙蝠算法处理高维复杂问题易陷入局部最优解的情况,引入了交叉变化来更新蝙蝠群体的位置,提出了一种改进的自适应变异蝙蝠算法;裴宇航^[4]等人在蝙蝠算法的速度更新公式中加入了惯性权重,并采用一种均匀分布和贝塔分布的随即调整策略;薛菲^[5]在其博士论文中阐述了大量蝙蝠算法的改进方案,具有非常高的参考价值;魏三强、张超^[6]等人为了进行局部搜索时蝙蝠算法易陷入局部极值的问题,使用 t-分布对局部搜索时的最优解进行变异操作。为最优解各维度增加 t 分布型随机扰动项;吕石磊、黄永霖^[7]等人在基本蝙蝠算法中加入了自适应步长控制机制和变异机制,有效解决了基本蝙蝠算法易陷入局部最优解的问题;吕赵明、张颖江^[8]等人将细菌觅食算法与蝙蝠算法相结合,以弥补蝙蝠算法局部搜索能力不足的问题,增加试探扰动因子和随机惯性权重来提高算法的求解精度与充分性;杜艳艳、刘升^[9]简化了基本蝙蝠算法的流程,并加入了位置收缩因子以平衡局部和全局搜索;Z. Chen 等人^[10]将基本蝙蝠算法的速度因子去除,同时利用正态分布来控制位置的变化,有效避免了基本蝙蝠算法易陷入局部最优解的情况。因此在以上研究的基础上,提出了一种新型的混合并行蝙蝠算法(hybrid parallel bat algorithm, HPBA)。

1 蝙蝠算法

蝙蝠算法将可行域内分布的各个可行解视为蝙蝠个体,每个蝙蝠都具有各自独立的速度、位置、频率、音量等,利用目标函数值来判断各个蝙蝠所处位置的优劣与否,通过速度、频率的变化,来改变蝙蝠所处位置,利用响度、脉冲频率的变化来逐步逼近最优解。在模拟蝙蝠搜索过程中, c 时刻蝙蝠 i 的速度更新公式如下所示:

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

$$v_i^c = v_i^{c-1} + (x_i^c - x_*)f_i \quad (2)$$

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

其中 β 是在 $[0, 1]$ 区间随机产生的向量;为了方便算法实现,常取频率 $f_{\min} = 0, f_{\max} = 100$,初始种群中蝙蝠的频率从 $[f_{\min}, f_{\max}]$ 中随机产生。 x_* 代表着当前最优解。和其他的仿生算法相似,在搜索过程中,也可以设定其局部搜索模式,如果随机数 $\text{rand} > r_i^0$ 其更新公式如下:

$$X_{\text{new}} = X_{\text{old}} + \delta A^c \quad (4)$$

在公式(4)中 $\delta \in [-1, 1]$ 是一个随机数, A^c 是 c 时刻所有蝙蝠的音量均值。

如果随机数 $\text{rand} > A^c$ & $f(X_{\text{new}}) < f(X_{\text{old}})$,则接受这个新解,同时替换相应的位置与速度。蝙蝠的脉冲响度 A 与脉冲频率 r 伴随着迭代而进行着更新,当蝙蝠逐步接近猎物的时候,响度 A 会逐步降低,同时脉冲频率 r 会逐步提高,具体公式如下:

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

$$r_i^c = r_i^0 [1 - \exp(-\gamma c)] \quad (6)$$

α, γ 分别是响度衰减系数、脉冲频率增加系数,二者均为常数,正由于二者的作用,使响度 A 逐步趋近于 0,脉冲频率 r 趋近与 r_0 。在实际应用过程中,常使 $\alpha = \gamma = 0.9$ 。在进行上述运行过程之后,对蝙蝠种群适应度进行排序,更新出当前最优个体及其位置;重复进行公式(1)~公式(6),直至满足算法结束条件,输出最优解。

2 改进的蝙蝠算法

基本蝙蝠算法在进行迭代时,常常会因为陷入局部最优解而出现种群更新停滞的情况,无法找寻到全局最优点,导致整个搜索结果失准,精度不高。尤其在使用蝙蝠算法处理高维度函数求极值问题时,往往会结果偏差巨大,导致这些现象出现的原因主要是算法在迭代后期的种群多样性丢失比较严重。在蝙蝠算法的迭代过程中,往往只考虑当前最优解的影响,而且对位置和速度的更新都是随机的,在这种情况下会丢失大量次优解和跳出局部最优解的可能性。

因此本文采用分组的形式保存多个次优解,加以结合粒子群算法的迭代公式在内的多种速度迭代公式来保证后期种群多样性,对各个次优解和最优解进行不同方式的迭代,以避免种群陷入局部最优解。

2.1 速度更新改进

为了避免在迭代过程中由于仅考虑当前最优解而带来的种群多样性丢失的问题,在随机产生初始种群的情况下,随机将其分为 N 组,分别记录为 $\text{Group}_{j_best} (j=1, 2, \dots, N)$, 设定两个阈值 k_1 、 $k_2 (k_1, k_2 \in (0, 1))$, 并使 $k_1 < k_2$, 产生随机数 rand , 并根据产生的随机数与 k_1 、 k_2 之间的大小关系, 选择不同的速度更新方式。在此, 本文引入了组间信息交流公式与粒子群算法中的速度更新公式作为可选择项, 改进后的速度更新伪代码如下公式(7)~公式(9)所示。

If $\text{rand} \leq k_1$;

$$v_i^c = v_i^{c-1} + f_i(x_i^{c-1} - \text{Group}_{j_best}) \quad (7)$$

Elseif $\text{rand} > k_1$ & $\text{rand} < k_2$;

$$v_i^c = w * v_i^{c-1} + c1 * f_i(x_i^{c-1} - \text{Group}_{j_best}) + c2 * f_i(x_i^{c-1} - x_*) \quad (8)$$

Elseif $\text{rand} \geq k_2$;

$$v_i^c = v_i^{c-1} + f_i(x_i^{c-1} - x_*) \quad (9)$$

End

根据上述的伪代码可以看出, 在进行蝙蝠个体的速度更新时, 不再局限于当前种群最优解的影响, 考虑到了次优解(即组间最优解)对速度更新的影响以及引入了粒子群算法中的速度更新公式, 扩大了全局搜索范围, 一定程度上避免了陷入局部最优解的情况, 增强了处理高维度多峰问题的能力。

2.2 领域搜索公式更新

在基本蝙蝠算法当中, 有效解的领域搜索过程是基于公式(4)的随机步长的更新, 这种更新迭代方式一定程度上保证了搜索范围的随机性, 但在面对高维度多峰问题、种群陷入局部最优解的情况时, 难以跳出局部最优解而导致算法的搜索能力降低、种群多样性丢失。为此设计了多层动态领域搜索模式, 即迭代过程中, 当 $\text{rand} > r_i^0$, 种群内各个蝙蝠在其所在组的组间最优蝙蝠周围飞行, 产生新的个体, 具体如公式(10)所示:

$$X_{\text{new}} = \delta A^c + \text{Group}_{j_best} \quad (10)$$

同时对于各个组间最优蝙蝠, 将组间最优蝙蝠中的最优异蝙蝠设为全局最优蝙蝠, 其解作为全局最优解, 记作 x_* , 各组间最优蝙蝠的领域搜索则在全局最优解周围进行, 为了提高局部搜索能力, 结合文献[6]引入了 t -分布作为权重替代 δ , g 为现迭代代数, 如公式(11)所示:

$$X_{\text{new}} = t(g) * X_* + X_* \quad (11)$$

2.3 HPBA 算法实施步骤

本文提出的改进算法运行步骤如下所示:

步骤1: 初始化参数: 种群规模 P 、维度 d 、目标函数 $f(x)$ 、初始脉冲 r_0 、响度 A_0 、频率 f 、取值范围 $X_{\max}$ 、 $X_{\min}$ 等;

步骤2: 随机产生初始蝙蝠种群, 计算个体适应度和种群平均适应度, 记录最优个体及其位置 x_* , 随机将种群内个体分为 N 组, 分别记录各组组内最优解;

步骤3: 使用公式(1)、(7)、(8)、(9)、(3)求解更新蝙蝠新的位置 x_i^c , 并根据 $X_{\max}$ 、 $X_{\min}$ 来约束蝙蝠的位置。

3 实验与结果分析

3.1 试验方案与设计

本文使用4个基准测试函数来测试算法的性能, 其中 $f_1(x)$ 到 $f_4(x)$ 都是高维多峰函数, 目的是用来检验待测试算法的局部搜索能力和避免早熟快速收敛能力, 4个函数的相关参数如表1所示。为了比较出 HPBA 的实际性能, 增加了基本蝙蝠算法与文献[11]中的动态调整惯性权重的自适应蝙蝠算法(DAWBA)进行对比, 各算法在 Intel(R) Core(TM) i5-3230 M CPU @ 2.68 GHz、RAM 8.00 GB、Windows 7 环境下, 在 Matlab 2014b 软件上编程并各自运行30次。

3.2 试验参数设定

为了避免由于参数设定不同而带来的实验结果误差, 对各算法的基本参数选择相同参数, 均以迭代次数达到最大迭代次数作为算法终止准则, 具体参数如下, 种群大小 $P=40$, 最大迭代次数 $G=500$, 响度 $A_0=0.5$, 初始脉冲 $r_0=0.5$, 频率范围 $f \in [0, 2]$, 脉冲音量衰减系数 $\alpha=0.95$, 脉冲衰减系数 $\gamma=0.95$, HPBA 算法中引入的粒子群算法速度更新公式中的惯性权重 $\omega=0.72984$, 学习因子 $c_1=c_2=1.49618$, DAWBA 算法中其余参数设定与文献[11]一致。

3.3 实验结果分析

将3个待验证算法 BA、HPBA、DAWBA 分别用表1中的4个测试函数进行实验, 经过各自30次的独立运行, 运行数据结果如表2所示, 图像对比如图1所示。

由表2可以看到3种算法在不同测试函数中的最优解、最差解和30次试验下的解集平均值, 对于存在多局部最优点的 f_1 、 f_2 函数实验时, BA 的求解结果都距离理论最优解差距较大, 而且易

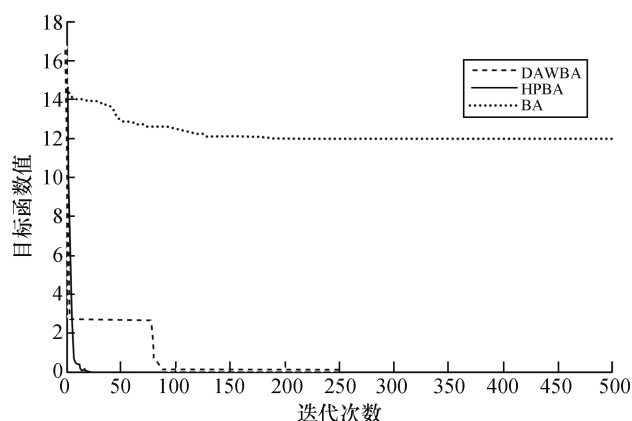
陷入局部最优点且难以跳出;HPBA 和 DAWBA 在运行 f_1 、 f_2 函数时,运行精度比 BA 有着明显的提升,通过图 1(a)、图 1(b)和表 1 的数据可以看出 HPBA 比 DAWBA 的运行结果精度更高和收敛速度更快,在 f_2 的试验中,HPBA 每一次都搜索到了理论最优值。对于 f_3 、 f_4 这类多峰函数,HPBA

在前期的快速收敛能力得到了体现,尤其是在图 1(d)的图像中,HPBA 在 20 代左右就完成了前期的全局搜索过程比 DAWBA 所需要的迭代次数要少,而且得到的计算精度要高出 8 个数量级,同样的 BA 在进行 f_3 、 f_4 函数检验时,依然暴露了其易陷入局部最优解的问题。

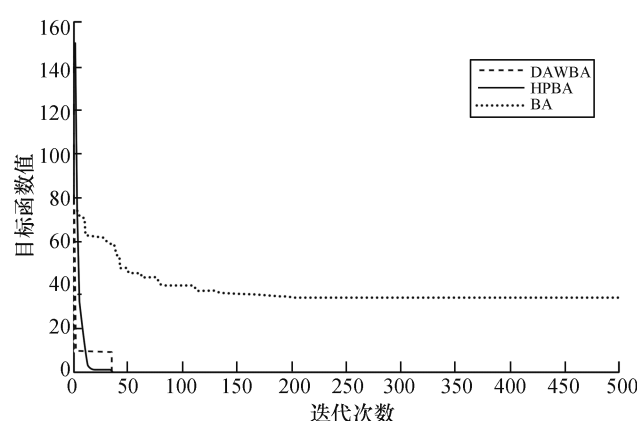
表 1 标准测试函数实验参数

Table 1 Standard test function experimental parameters

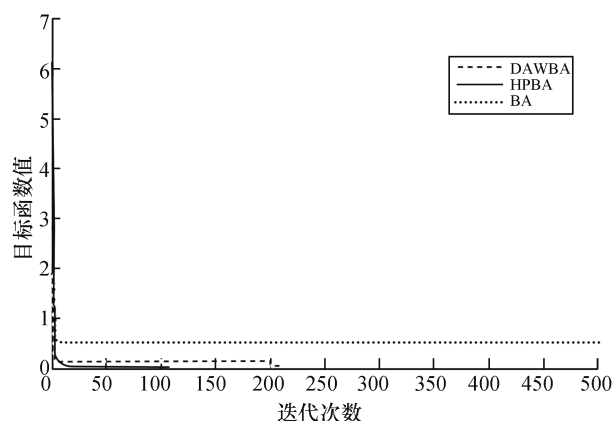
函数名	表达式	变量范围	理论最优值
Ackley	$f_1 = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$	0
Rastrigin	$f_2 = 10d + \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i)]$	$[-5.12, 5.12]$	0
Griewank	$f_3 = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-5.12, 5.12]$	0
Schaffer F7	$f_4 = \frac{1}{d-1} \sum_{i=1}^{d-1} [(x_i^2 + x_{i+1}^2)^{0.25} + (x_i^2 + x_{i+1}^2)^{0.25} \sin^2(50(x_i^2 + x_{i+1}^2)^{0.1})]$	$[-100, 100]$	0



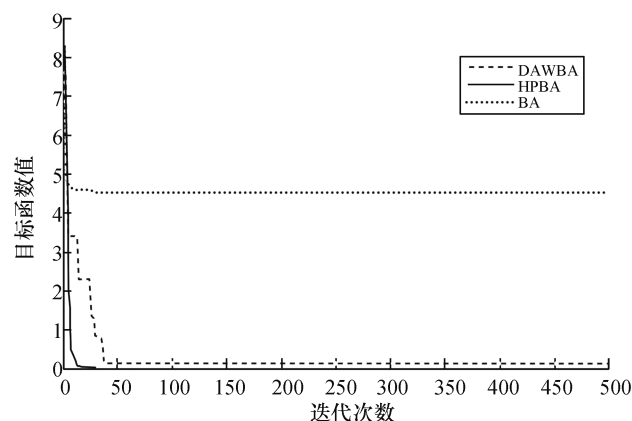
(a)Ackley函数收敛曲线



(b)Rastrigin函数收敛曲线



(c)Griewank函数收敛曲线



(d)Schaffer F7函数收敛曲线

图 1 3 种算法在标准测试函数上的收敛曲线

Fig.1 Convergence curve of 3 algorithms on standard test function

表 2 三种算法实验对比

Table 2 3 Experiment comparison of algorithms

函数	算法	最优解	最差解	平均值
f_1	BA	9.11	16.299 8	13.494 65
	DAWBA	1.544×10^{-5}	3.33×10^{-5}	2.803×10^{-5}
	HPBA	8.88×10^{-16}	1.69×10^{-14}	1.54×10^{-15}
f_2	BA	9.951 3	55.718 8	27.586 08
	DAWBA	7.11×10^{-15}	1.30×10^{-13}	1.95×10^{-14}
	HPBA	0	0	0
f_3	BA	0.091 092	1.171 2	0.568 644
	DAWBA	1.307×10^{-11}	2.564×10^{-11}	2.185×10^{-11}
	HPBA	0	0	0
f_4	BA	3.878 8	6.7 767	5.507 64
	DAWBA	0.034 846	1.129	0.353 538
	HPBA	8.76×10^{-11}	2.32×10^{-8}	4.37×10^{-9}

4 结 论

本文在蝙蝠算法的基础上,引入了分组迭代模式和多种速度迭代公式加强了全局搜索能力,更新了种群领域搜索公式的基础上引用了 t 分布作为种群最优解的领域搜索方式,补足了蝙蝠算法的局部搜索能力,避免了算法陷入局部最优解。通过测试函数的实验,验证了 HPBA 算法具有良好的寻优性能。在未来的研究工作中,作者将尝试把 HPBA 算法与其他工程应用问题相结合,以扩展其应用领域。

参考文献:

[1] YANG X S. A newmetaheuristic bat-inspired algorithm [M]. Berlin:Springer,2010.

[2] YANG X S,GANDOMI A H. Bat algorithm;a novel approach for global engineering optimization[J]. Engineering computations,2012,29(5):464-483.

[3] 盛孟龙,贺兴时,王慧敏. 一种改进的自适应变异蝙蝠算法 [J]. 计算机技术与发展, 2014, 24 (10): 131-134.

[4] 裴宇航,刘景森,李煜. 一种动态调整惯性权重的自适应蝙蝠算法[J]. 计算机科学,2017,44(6):240-244.

[5] 薛菲. 基于蝙蝠算法的启发式智能优化研究与应用 [D]. 北京:北京工业大学,2016.

[6] 魏三强,张超. 蝙蝠算法的一种改进方法[J]. 东北师大学报(自然科学版),2017,49(4):76-81.

[7] 吕石磊,黄永霖,陈海强,等. 基于自适应步长的改进蝙蝠算法[J]. 控制与决策,2018,33(3):557-564.

[8] 吕赵明,张颖江. 改进的新型蝙蝠算法[J]. 计算机技术与发展,2018,28(5):63-67,72.

[9] 杜艳艳,刘升. 一种改进的自适应混合型蝙蝠算法[J]. 微电子学与计算机,2018,(6):135-140.

[10] CHEN Z,ZHOU Y,LU M. A simplified adaptive bat algorithm based on frequency[J]. Journal of computational information systems,2013,9(16):6451-6458.

[11] 裴宇航,刘景森,李煜. 一种动态调整惯性权重的自适应蝙蝠算法 [J]. 计算机科学, 2017, 44 (6): 240-244.

(责任编辑:周泉)