

基于越界重置和高斯变异的蝙蝠优化算法^{*}

李永恒, 赵志刚

(广西大学计算机与电子信息学院, 广西 南宁 530004)

摘 要:针对蝙蝠算法个体越界、易早熟收敛的问题,提出一种基于越界重置和高斯变异的蝙蝠优化算法。新算法将飞越解空间边界的个体拉回解空间内,利用越界重置策略重新分配位置。通过高斯变异策略控制个体的搜索范围,使种群以最优解为中心向四周呈放射状搜索,增强了算法的局部搜索和全局寻优能力。蝙蝠算法在靠近目标解时响度和脉冲发射频率更新不协调,影响了算法的持续进化能力,通过线性渐变策略保证响度和脉冲发射频率的变化与算法持续进化相适应。研究了在解空间不同位置关系的情况下新算法和对比算法的优化能力,并结合实验数据对算法收敛稳定性进行分析。实验结果表明,提出的新算法具有较好的收敛速度和精度,其全局寻优能力和高维问题优化能力体现了很好的鲁棒性。

关键词:蝙蝠算法;越界重置;高斯变异;搜索范围

中图分类号:TP301.6

文献标志码:A

doi:10.3969/j.issn.1007-130X.2019.01.019

An improved bat algorithm based on cross-border relocation and Gaussian mutation

LI Yong-heng, ZHAO Zhi-gang

(College of Computer and Electronics Information, Guangxi University, Nanning 530004, China)

Abstract: Aiming at the problem that individuals cross border and suffer from premature convergence in the bat algorithm, we propose an improved bat algorithm based on cross-border relocation and Gaussian mutation. The algorithm pulls the individuals which cross the solution boundary back into the solution space, and uses the cross-border relocation strategy to relocate. We then use the Gaussian mutation strategy to control the search range of individuals, and the population is radically searched around the optimal solution as the center, which enhances the local search and global optimization ability of the bat algorithm. Since the loudness and pulse frequency of the bat algorithm are inconsistent when bats approaching the target solution, which affects the continuous evolution ability of the algorithm, we introduce the linear gradient strategy to ensure that the updates of loudness and pulse frequency are compatible with the continuous evolution of the algorithm. We compare the optimization ability of the new algorithm with other algorithms under different position relationships in the solution space, and analyze the convergence stability of the new algorithm with the experimental data. Experimental results show that the proposed algorithm has better convergence speed and accuracy. In addition, the global optimization ability and high dimensional problem optimization ability of the algorithm demonstrate good robustness.

Key words: bat algorithm; cross-border relocation; Gaussian mutation; search range

^{*} 收稿日期:2017-09-17;修回日期:2018-03-10

基金项目:广西自然科学基金(2015GXNSFAA139296)

通信地址:530004 广西南宁市广西大学计算机与电子信息学院

Address: College of Computer and Electronics Information, Guangxi University, Nanning 530004, Guangxi, P. R. China

1 引言

群智能是对自然界社会性动物或昆虫集体行为的模拟,蝙蝠算法 BA (Bat-inspired Algorithm)^[1]的提出是受到蝙蝠回声定位的启发。群智能算法需要解决粒子越界问题。以粒子群优化算法^[2]为例,最大速度和位置限制策略不能保证将粒子控制在解空间内,粒子在算法进化前期大量聚集在边界,导致算法收敛速度变慢。文献[3]提出了吸收墙、反射墙和隐匿墙。文献[4]综合了吸收墙和反射墙的特点,提出了衰减墙。文献[5]比较了4种边界变异策略,并对4种策略在优化问题中存在的优势和不足进行了对比。迟玉红等人^[6]研究了不同边界策略下粒子的运动轨迹,提出对搜索空间的边界进行缩放的方法,有效控制了粒子的搜索范围。蝙蝠算法与其他群智能算法本质相同,因此解决蝙蝠算法的种群个体越界问题并提出一种能减少个体在边界聚集、提高优化效率的边界控制策略是本文的研究重点。

本文在蝙蝠算法的基础上,分析其早熟停滞的原因,从蝙蝠算法的越界控制策略、种群多样性和蝙蝠个体接近目标时脉冲发射频率和响度的变化三个角度来改进蝙蝠算法。

2 蝙蝠算法

2.1 蝙蝠算法的结构

蝙蝠个体在进化过程中通过计算适应度值来调整他们的移动速度、频率和位置。蝙蝠更新位置和速度的公式如式(1)和式(2)所示:

$$V_{t+1} = V_t + (X_t - best) f \quad (1)$$

$$X_{t+1} = X_t + V_{t+1} \quad (2)$$

其中, t 表示当前进化次数, X_t 、 V_t 表示蝙蝠在第 t 次进化的位置和速度, $f \in [0, 2]$ 表示蝙蝠发出的声音频率, $best$ 表示当前全局最优解。

局部搜索时,每只蝙蝠的位置更新公式如式(3)所示:

$$X_{new} = X_{old} + \epsilon A', \epsilon \in [-1, 1] \quad (3)$$

其中, $\epsilon \in [-1, 1]$ 是随机数, A' 是所有蝙蝠在第 t 代的平均响度。

一旦蝙蝠发现了猎物,响度会逐渐降低,脉冲速率会逐渐提高。响度 A 和脉冲发射频率 r 的更新公式如式(4)所示:

$$A' = \alpha A'^{-1}, r' = r_0 [1 - e^{-\gamma}] \quad (4)$$

其中, α 和 γ 是恒量, α 类似于模拟退火算法中冷却进程表中的冷却因素。

2.2 蝙蝠算法存在的问题

(1) 个体越界问题。

蝙蝠算法对越界个体的处理方式是将越界个体置于边界上,其速度的大小和方向不变。置于边界上的越界粒子往往在算法的下次进化中仍然停留在边界,这导致了算法前期的收敛速度慢。常见的边界变异策略有3种:①吸收墙(Absorbing)。如果粒子的任一维度的位置产生了越界,则将越界粒子置于边界上,并将粒子的速度置零。②反射墙(Reflecting)。粒子越界的位置处理方式和吸收墙相同,都是将越界粒子置于边界上。越界粒子的速度方向置为相反,速度大小不发生变化。③减震墙(Damping)。一旦粒子位置产生了越界,将越界粒子置于边界上,粒子速度改变为相反的方向,并用一个0~1的随机因子来控制速度大小。

(2) 缺乏种群多样性。

蝙蝠算法以一定的概率获得当前最优解附近的临时替代解,并以一定的概率接受。蝙蝠算法接受新解的思想使算法的种群多样性在进化后期不够丰富。文献[7]提出自适应变异蝙蝠算法,根据迭代次数判断是否进行高斯变异,为蝙蝠个体开拓搜索空间。文献[8]采用小生境技术来提高蝙蝠种群的多样性。文献[9]采用差分进化的搜索思想来增强种群的多样性。以上学者们提出的策略都增强了蝙蝠算法的持续进化能力,一定程度上解决了算法早熟收敛问题。

3 改进的蝙蝠算法

对蝙蝠算法的改进分为3个方面:改进算法的越界变异策略;控制局部搜索步长,增加高斯扰动;改进响度和脉冲速率的公式。

3.1 越界重置策略

吸收墙策略的缺点在于将大量粒子置于边界上,若全局最优位置不位于边界附近,且待优化问题又含有多个局部最优值,采用此策略的算法极易陷入局部最优。反射墙、减震墙与吸收墙相比,由于速度方向的变化,可以避免大量粒子停滞在边界附近的局部最优解的情况,但是也具有一定的局限性。在这两种边界变异策略的影响下,粒子在触碰边界后虽然在下次进化过程中位置会发生变化,

但是会出现粒子分布在以边界为中心的超圆附近的情况。以二维空间的优化问题为例,反射墙与减震墙影响的粒子在越界后的下一次进化过程中的位置通常会呈现环形分布,若此时全局最优点位于边界中心,则算法在后期的迭代中会出现两种情况:(1)陷入非中心点附近的局部最优解;(2)在群体间协同和当前最优解的吸引下继续进化,但是增加了时间开销与计算量。

本文提出的边界变异策略充分利用蝙蝠算法随机搜索的特性。在算法进化过程中,如果出现了蝙蝠个体的位置处于解空间之外,将越界个体在解空间内进行随机重置,并保持个体的速度大小和方向。越界变异策略的位置更新如式(5)所示:

$$X_{\text{new}} = Lb + \lambda(Ub - Lb) \quad (5)$$

其中, Lb 是解空间的下界, Ub 是解空间的上界, $\lambda \in [0, 1]$ 是一个随机变量。

3.2 变异策略

算法寻优初期,为避免陷入停滞,在最优解附近生成局部解的步长应足够大;而算法后期为了提高解的精度,最优解附近生成新解的步长要足够小。因此为蝙蝠设置线性递减的步长变化,算法在最优解附近生成局部解的公式修改为:

$$step = \frac{step(T-t)}{T} + step_{\min} \quad (6)$$

$$X_{\text{new}} = best + step * \epsilon \quad (7)$$

其中, T 表示最大迭代次数, t 表示当前迭代次数, $step$ 是局部搜索时的步长,可以控制局部搜索的范围, $step_{\min}$ 是最小步长, ϵ 表示与 $best$ 同阶的随机矩阵。

为了提高蝙蝠种群的多样性,对进化过程中产生的不满足条件的新解进行高斯变异。个体进行高斯变异的更新公式为:

$$X = X + Gauss \quad (8)$$

其中, $Gauss$ 表示数学期望为 μ 、方差为 σ 的高斯分布。参数 σ 影响高斯分布的范围和密集程度,因此可以通过调整 σ 的值来控制搜索空间的种群密度。 σ 的更新公式如式(9)所示:

$$\sigma = \eta(b_{\max} - b_{\min})r \quad (9)$$

其中, η 是一个经验参数,用来控制高斯变异的范围, b_{\max} 表示解空间上限, b_{\min} 表示解空间下限, r 表示脉冲发射频率。

3.3 响度和脉冲发射频率

如果按照基本蝙蝠算法中所提出的脉冲发射频率和响度的更新公式,算法在初期就已经失去生

成局部解的能力,导致算法的收敛速度变低。而且算法在后期基本失去接受新解的能力,如果算法在进化后期跳出了局部最优,获得了更好的解,却由于不满足接受条件而不能更新最优解的位置,将增加算法停滞早熟的可能性。

对基本蝙蝠算法的响度 A 和脉冲发射频率 r ,采用线性变化的方式进行改进,如式(10)和式(11)所示:

$$A = A_{\max} - \frac{(A_{\max} - A_{\min})t}{T} \quad (10)$$

$$r = r_{\min} + \frac{(r_{\max} - r_{\min})t}{T} \quad (11)$$

其中, A_{\max} 表示蝙蝠响度的最大值, A_{\min} 表示蝙蝠响度的最小值; r_{\max} 表示蝙蝠脉冲发射频率的最大值, r_{\min} 表示蝙蝠脉冲发射频率最小值; t 表示算法当前迭代次数, T 表示算法总迭代次数。

3.4 算法流程

本文提出的基于越界重置和高斯变异的蝙蝠优化算法 CRBA(an improved Bat Algorithm with Cross-border Relocation and Gaussian mutation) 流程如下:

步骤 1 参数初始化,初始化蝙蝠的位置 X 和速度 V , 为不同蝙蝠个体初始化频率 f 、脉冲发射频率 r , 响度 A ;

步骤 2 计算蝙蝠种群的适应度值,选择最优个体为 $best$;

步骤 3 按照式(1)、式(2)更新蝙蝠的速度和位置,按照式(5)进行越界处理;

步骤 4 生成一个 $0 \sim 1$ 的随机数 n , 若 $n > r$, 按照公式(3)在全局最优解 $best$ 附近生成一个新的局部解 X_{new} ;

步骤 5 若蝙蝠当前响度满足 $A < n$, 且当前解的适应度值更优,接受这个新解,按照公式(10)、公式(1)更新响度和脉冲发射频率;若不满足条件,则按照公式(8)对当前解进行高斯变异;

步骤 6 排列蝙蝠并找到当前最优解 $best$;

步骤 7 如未满足结束条件,则返回步骤 3;

步骤 8 输出全局最优位置。

4 实验仿真

4.1 实验设计

为了评价改进蝙蝠算法的鲁棒性,选用了 6 个经典的标准测试函数: Sphere 函数、Griewank 函数、Rastrigin 函数、Rosenbrock 函数、Ackley 函数、Quartic Noise 函数。由于实际优化问题中,很难判定最优解与搜索空间的位置关系,因此实验设计了最优解和搜索空间的 3 组位置关系:最优解位于中心(Center)、最优解靠近边界(Close)、最优解恰在边界上(Exactly)。测试函数及位置关系如表 1 所示。

Table 1 Test functions and position relationship
表 1 测试函数及位置关系

| 名称 | 函数 | 搜索空间 (Center) | 搜索空间 (Close) | 搜索空间 (Exactly) | 理论 最优值 | 寻优 精度 |
|------------------|---|------------------|-----------------|-------------------|-----------|----------|
| Sphere | $f_1(x)=\sum_{i=1}^D x_i^2$ | $[-100,100]^D$ | $[-10,190]^D$ | $[0,200]^D$ | 0 | 0.01 |
| Griewank | $f_2(x)=\frac{1}{4000}\sum_{i=1}^D x_i^2-\prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}})+1$ | $[-600,600]^D$ | $[-10,1190]^D$ | $[0,1200]^D$ | 0 | 0.01 |
| Rastrigin | $f_3(x)=\sum_{i=1}^D (x_i^2-10\cos(2\pi x_i)+10)$ | $[-5.12,5.12]^D$ | $[-1,9.24]^D$ | $[0,10.24]^D$ | 0 | 50 |
| Rosenbrock | $f_4(x)=\sum_{i=1}^{D-1} (100(x_{i+1}-x_i^2)^2+(x_i-1)^2)$ | $[-30,30]^D$ | $[-3,57]^D$ | $[0,60]^D$ | 0 | 100 |
| Ackley | $f_5(x)=-20\exp\left[-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}\right]-\exp\left[\frac{1}{D}\sum_{i=1}^D \cos 2\pi x_i\right]+20+e$ | $[-32,32]^D$ | $[-3,61]^D$ | $[0,64]^D$ | 0 | 0.01 |
| Quartic Noise | $f_6(x)=\sum_{i=1}^D ix_i^4+random(0,1)$ | $[-1.28,1.28]^D$ | $[-0.2,2.54]^D$ | $[0,2.56]^D$ | 0 | 0.01 |

参数定义如下: N 表示种群规模, D 表示问题维度, T 表示最大迭代次数,分3组进行实验:(1) A 组: $N=20,D=10,T=1000$;(2) B 组: $N=20,D=20,T=1000$;(3) C 组: $N=50,D=100,T=1000$;每组实验分别运行50次,统计算法优化的最优解和寻优成功率。将CRBA算法与以下四种算法进行对比:变步长的自适应蝙蝠算法VSABA(Variation Step Adaptive Bat Algorithm)^[10]、采用吸收墙变异策略(Absorbing)、采用减震墙变异策略(Damping)和采用反射墙变异策略(Reflecting)的蝙蝠算法。测试这5种算法在3组实验的不同位

置关系下的优化能力。

4.2 实验结果

实验统计了5种算法在最优解和边界不同位置下对每个函数的优化结果,其中, Min 表示寻优最小值, Max 表示寻优最大值, sr 表示算法在指定目标精度下的寻优成功率,实验数据如表2~表7所示。

4.3 算法收敛稳定性分析

算法收敛的判别标准是算法能否在相应函数最优值的附近达到指定的寻优精度。与CRBA对比的算法的收敛准则有以下3个:(1)相邻两次迭

Table 2 Optimization results of Sphere function
表 2 Sphere 函数优化结果统计

| f_1 | | A 组 | | | B 组 | | | C 组 | | |
|------------|---------|----------|----------|------|----------|----------|------|----------|----------|------|
| | | Min | Max | sr | Min | Max | sr | Min | Max | sr |
| CRBA | Center | 0 | 3.36e-08 | 1 | 0 | 1.78e-07 | 1 | 0 | 5.93e-06 | 1 |
| | Close | 0 | 3.45e-08 | 1 | 0 | 1.95e-07 | 1 | 0 | 6.29e-06 | 1 |
| | Exactly | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| VSABA | Center | 7.60e-09 | 2.50e-08 | 1 | 4.96e-08 | 1.08e-07 | 1 | 2.30e-06 | 3.16e-06 | 1 |
| | Close | 1.57e-09 | 2.21e-08 | 1 | 5.24e-08 | 1.12e-07 | 1 | 2.29e-06 | 3.15e-06 | 1 |
| | Exactly | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| Absorbing | Center | 5.73e-09 | 2.30e-08 | 1 | 5.05e-08 | 1.17e-07 | 1 | 2.31e-06 | 3.13e-06 | 1 |
| | Close | 7.03e-09 | 2.13e-08 | 1 | 5.19e-08 | 1.13e-07 | 1 | 2.38e-06 | 3.47e-06 | 1 |
| | Exactly | 0 | 0 | 1 | 0 | 0 | 1 | 1.20e-06 | 2.15e-06 | 1 |
| Damping | Center | 5.25e-09 | 2.25e-08 | 1 | 4.87e-08 | 1.14e-07 | 1 | 2.17e-06 | 3.24e-06 | 1 |
| | Close | 6.74e-09 | 2.26e-08 | 1 | 4.88e-08 | 1.13e-07 | 1 | 2.14e-06 | 3.35e-06 | 1 |
| | Exactly | 0 | 0 | 1 | 0 | 6.92e-08 | 1 | 9.40e-07 | 2.08e-06 | 1 |
| Reflecting | Center | 6.45e-09 | 2.31e-08 | 1 | 5.75e-08 | 1.16e-07 | 1 | 2.34e-06 | 3.17e-06 | 1 |
| | Close | 6.72e-09 | 2.26e-08 | 1 | 4.66e-08 | 9.72e+03 | 0.48 | 1.91e-06 | 3.10e-06 | 1 |
| | Exactly | 0 | 1.90e-08 | 1 | 0 | 1.44e+04 | 0.48 | 2.03e-06 | 3.14e-06 | 1 |

Table 3 Optimization results of Griewank function

表 3 Griewank 函数优化结果统计

| f_2 | | A 组 | | | B 组 | | | C 组 | | |
|------------|---------|-------|----------|------|----------|----------|------|----------|----------|------|
| | | Min | Max | sr | Min | Max | sr | Min | Max | sr |
| CRBA | Center | 0 | 4.00e-09 | 1 | 0 | 1.44e-08 | 1 | 0 | 2.70e-06 | 1 |
| | Close | 0 | 4.70e-09 | 1 | 0 | 1.57e-08 | 1 | 0 | 1.83e-07 | 1 |
| | Exactly | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| VSABA | Center | 10.03 | 90.65 | 0 | 5.92 | 151.41 | 0 | 5.68e-08 | 41.31 | 0.48 |
| | Close | 0.174 | 0.272 | 0 | 3.95e-09 | 612.40 | 0.50 | 6.20e-08 | 4.33e+03 | 0.14 |
| | Exactly | 0 | 348.79 | 0.98 | 0 | 1.15e+03 | 0.80 | 0 | 3.54e+03 | 0.54 |
| Absorbing | Center | 0.061 | 10.60 | 0 | 4.63e-09 | 0.90 | 0.38 | 6.00e-08 | 0.011 | 0.90 |
| | Close | 0.022 | 19.80 | 0 | 4.31e-09 | 0.12 | 0.44 | 5.87e-08 | 0.012 | 0.84 |
| | Exactly | 0 | 11.04 | 0.82 | 0 | 22.91 | 0.08 | 4.30e-08 | 0.025 | 0.76 |
| Damping | Center | 1.37 | 31.73 | 0 | 3.87e-09 | 8.08 | 0.22 | 5.83e-08 | 0.016 | 0.86 |
| | Close | 0.036 | 5.03 | 0 | 2.88e-09 | 0.25 | 0.42 | 6.08e-08 | 0.034 | 0.92 |
| | Exactly | 0 | 15.22 | 0.50 | 0 | 5.06 | 0.06 | 4.11e-08 | 0.047 | 0.70 |
| Reflecting | Center | 8.70 | 80.98 | 0 | 3.49 | 118.99 | 0 | 6.28e-08 | 16.55 | 0.60 |
| | Close | 0.191 | 365.70 | 0 | 0.226 | 1.21e+03 | 0 | 1.39e+03 | 4.56e+03 | 0 |
| | Exactly | 0 | 310.64 | 0.52 | 115.25 | 1.22e+03 | 0 | 1.99e+03 | 4.80e+03 | 0 |

Table 4 Optimization results of Rastrigin function

表 4 Rastrigin 函数优化结果统计

| f_3 | | A 组 | | | B 组 | | | C 组 | | |
|------------|---------|-------|----------|------|--------|----------|------|----------|----------|----|
| | | Min | Max | sr | Min | Max | sr | Min | Max | sr |
| CRBA | Center | 0 | 7.90e-06 | 1 | 0 | 4.02e-05 | 1 | 0 | 0.001 | 1 |
| | Close | 0 | 7.21e-06 | 1 | 0 | 3.51e-05 | 1 | 0 | 0.001 | 1 |
| | Exactly | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 5.57e-05 | 1 |
| VSABA | Center | 5.96 | 94.52 | 0.66 | 24.87 | 157.20 | 0.2 | 126.36 | 372.11 | 0 |
| | Close | 7.95 | 89.54 | 0.9 | 17.90 | 389.02 | 0.38 | 1.50e+03 | 2.04e+03 | 0 |
| | Exactly | 0 | 243.76 | 0.9 | 0 | 510.40 | 0.58 | 1.09e+03 | 2.77e+03 | 0 |
| Absorbing | Center | 10.94 | 82.58 | 0.76 | 21.88 | 141.28 | 0.32 | 121.38 | 384.05 | 0 |
| | Close | 7.95 | 183.06 | 0.32 | 117.40 | 372.10 | 0 | 1.42e+03 | 1.92e+03 | 0 |
| | Exactly | 0 | 201.97 | 0.52 | 198.98 | 536.26 | | 2.17e+03 | 2.98e+03 | 0 |
| Damping | Center | 14.92 | 65.66 | 0.88 | 22.88 | 86.56 | 0.46 | 112.43 | 347.24 | 0 |
| | Close | 7.95 | 155.21 | 0.34 | 74.62 | 417.87 | 0 | 1.35e+03 | 1.97e+03 | 0 |
| | Exactly | 0 | 201.97 | 0.32 | 56.71 | 518.35 | | 1.96e+03 | 2.74e+03 | 0 |
| Reflecting | Center | 10.94 | 78.60 | 0.84 | 26.86 | 140.28 | 0.50 | 138.29 | 367.13 | 0 |
| | Close | 7.95 | 164.16 | 0.44 | 98.49 | 344.25 | 0 | 1.36e+03 | 1.88e+03 | 0 |
| | Exactly | 0 | 198.98 | 0.40 | 0 | 578.05 | 0.04 | 2.08e+03 | 2.87e+03 | 0 |

代点的位置向量距离相差充分小;(2)相邻两次迭代点处函数值的下降量充分小;(3)无约束优化问题中,目标函数具有一阶连续可微性,且梯度充分接近于零。以上实验结果表明,在含有多个极值的多峰函数优化过程中,对比算法容易停滞在目标函数的鞍点附近。对比算法本身无法区分鞍点和极小点,只能以一定的概率寻找最优解,所以才会出

现表 2~表 7 所示的实验结果:以有限的概率达到收敛标准,收敛性极其不稳定。而 CRBA 算法虽然不能判断当前驻点是鞍点还是极小点,但是算法自适应缩放的寻优过程提高了算法的全局寻优能力。且 CRBA 算法进行种群多样性变异操作是按照最大化利用每一次迭代的的思想,进行高斯变异的方差与迭代的次数呈线性相关。按照这种思想和

Table 5 Optimization results of Rosenbrock function
表 5 Rosenbrock 函数优化结果统计

| f_4 | | A 组 | | | B 组 | | | C 组 | | |
|------------|---------|----------|----------|------|----------|----------|------|----------|----------|------|
| | | Min | Max | sr | Min | Max | sr | Min | Max | sr |
| CRBA | Center | 5.90e-06 | 0.020 | 1 | 4.76e-05 | 0.091 | 1 | 2.46e-05 | 0.151 | 1 |
| | Close | 6.03e-07 | 0.218 | 1 | 1.52e-06 | 0.072 | 1 | 6.38e-05 | 0.012 | 1 |
| | Exactly | 9.27e-07 | 0.015 | 1 | 4.33e-06 | 0.136 | 1 | 1.91e-04 | 0.043 | 1 |
| VSABA | Center | 0.413 | 1.16e+03 | 0.82 | 13.83 | 1.52e+03 | 0.64 | 92.26 | 677.94 | 0.28 |
| | Close | 4.26 | 4.97e+03 | 0.64 | 15.35 | 5.55e+03 | 0.54 | 93.19 | 6.31e+03 | 0.20 |
| | Exactly | 5.91 | 812.87 | 0.8 | 17.10 | 5.74e+03 | 0.62 | 96.76 | 5.96e+03 | 0.62 |
| Absorbing | Center | 1.69 | 530.00 | 0.80 | 11.66 | 1.53e+03 | 0.80 | 90.02 | 1.45e+03 | 0.32 |
| | Close | 2.99 | 4.93e+03 | 0.56 | 7.03 | 5.34e+03 | 0.32 | 94.58 | 5.11e+03 | 0.06 |
| | Exactly | 0.125 | 443.49 | 0.94 | 0.034 | 117.61 | 0.90 | 1.00 | 112.21 | 0.86 |
| Damping | Center | 1.41 | 386.81 | 0.92 | 10.12 | 1.12e+03 | 0.74 | 94.33 | 676.58 | 0.32 |
| | Close | 0.347 | 4.49e+03 | 0.66 | 13.08 | 5.46e+03 | 0.24 | 91.37 | 5.19e+03 | 0.08 |
| | Exactly | 0.065 | 435.91 | 0.98 | 0.044 | 116.10 | 0.92 | 0.02 | 156.04 | 0.80 |
| Reflecting | Center | 2.85 | 354.27 | 0.84 | 9.91 | 1.70e+03 | 0.76 | 83.29 | 1.373+03 | 0.28 |
| | Close | 4.39 | 5.08e+03 | 0.64 | 8.12 | 1.11e+03 | 0.66 | 94.88 | 3.638+03 | 0.06 |
| | Exactly | 0.015 | 5.17e+03 | 0.76 | 2.50 | 5.86e+03 | 0.62 | 97.81 | 4.857+03 | 0.04 |

Table 6 Optimization results of Ackley function
表 6 Ackley 函数优化结果统计

| f_5 | | A 组 | | | B 组 | | | C 组 | | |
|------------|---------|----------|----------|------|----------|----------|------|----------|----------|-----|
| | | Min | Max | sr | Min | Max | sr | Min | Max | sr |
| CRBA | Center | 8.88e-16 | 2.41e-04 | 1 | 8.88e-16 | 4.01e-04 | 1 | 3.73e-04 | 0.001 | 1 |
| | Close | 8.88e-16 | 2.49e-04 | 1 | 2.02e-06 | 4.08e-04 | 1 | 1.34e-05 | 8.70e-04 | 1 |
| | Exactly | 2.26e-05 | 2.46e-04 | 1 | 8.88e-16 | 3.82e-04 | 1 | 8.88e-16 | 9.72e-04 | 1 |
| VSABA | Center | 12.57 | 19.95 | 0 | 13.02 | 19.96 | 0 | 12.96 | 19.96 | 0 |
| | Close | 1.07e-04 | 4.77 | 0.64 | 2.36e-04 | 4.42 | 0.12 | 3.59 | 4.27 | 0 |
| | Exactly | 8.88e-16 | 8.88e-16 | 1 | 8.88e-16 | 4.93 | 0.82 | 8.88e-16 | 4.56 | 0.1 |
| Absorbing | Center | 1.59e-04 | 19.96 | 0.02 | 12.75 | 18.40 | 0 | 12.38 | 19.96 | 0 |
| | Close | 9.37e-05 | 4.54 | 0.26 | 2.31 | 4.36 | 0 | 3.55 | 4.46 | 0 |
| | Exactly | 8.88e-16 | 4.42 | 0.64 | 8.88e-16 | 4.66 | 0.12 | 3.63 | 4.69 | 0 |
| Damping | Center | 1.22e-04 | 16.04 | 0.02 | 6.20 | 16.93 | 0 | 3.50 | 19.96 | 0 |
| | Close | 1.20e-04 | 4.66 | 0.26 | 2.31 | 4.29 | 0 | 3.50 | 4.27 | 0 |
| | Exactly | 8.88e-16 | 4.77 | 0.50 | 8.88e-16 | 4.54 | 0.04 | 3.62 | 4.52 | 0 |
| Reflecting | Center | 9.45 | 19.93 | 0 | 13.38 | 19.96 | 0 | 13.60 | 19.96 | 0 |
| | Close | 9.68e-05 | 4.66 | 0.32 | 2.30e-04 | 4.83 | 0.02 | 3.50 | 4.23 | 0 |
| | Exactly | 8.88e-16 | 4.77 | 0.46 | 8.88e-16 | 4.66 | 0.02 | 3.92 | 4.65 | 0 |

策略,CRBA 算法区别于对比算法的优势就是永远不会停滞,种群在最大迭代次数限制内的每一次迭代都在进化,为当前种群搜集更多的信息,在更大概率上利于发现全局最优解。

从表 2~表 7 的实验结果来看,无论是单峰还是含有多个鞍点的多峰函数,CRBA 的寻优成功率

更高,收敛性更加稳定。

图 1~图 6 是在问题维度 $D=10$ 和 $D=100$ 时,对目标函数优化的收敛曲线。其中,横坐标表示进化次数,纵坐标是适应度值,为了直观表示,取 $\lg(f_{\min})$ 作为评判指标, f_{\min} 表示当前进化过程中所搜索到的全局最小值。

Table 7 Optimization results of Quartic Noise function

表 7 Quartic Noise 函数优化结果统计

| f_6 | | A 组 | | | B 组 | | | C 组 | | |
|------------|---------|----------|----------|------|----------|-------|------|----------|----------|------|
| | | Min | Max | sr | Min | Max | sr | Min | Max | sr |
| CRBA | Center | 5.71e-06 | 0.001 | 1 | 1.88e-05 | 0.003 | 1 | 7.37e-08 | 0.001 | 1 |
| | Close | 2.09e-05 | 7.11e-04 | 1 | 1.48e-05 | 0.001 | 1 | 3.83e-06 | 2.78e-04 | 1 |
| | Exactly | 4.19e-06 | 0.006 | 1 | 4.82e-05 | 0.005 | 1 | 3.16e-06 | 0.004 | 1 |
| VSABA | Center | 0.001 | 0.050 | 0.26 | 0.023 | 0.166 | 0 | 0.828 | 1.89 | 0 |
| | Close | 0.001 | 0.045 | 0.24 | 0.025 | 0.308 | 0 | 0.813 | 2.78 | 0 |
| | Exactly | 2.71e-06 | 0.017 | 0.98 | 7.72e-06 | 0.015 | 0.98 | 1.10e-05 | 3.24 | 0.30 |
| Absorbing | Center | 0.001 | 0.033 | 0.58 | 0.011 | 0.110 | 0 | 0.482 | 1.84 | 0 |
| | Close | 7.83e-04 | 0.017 | 0.86 | 0.008 | 0.095 | 0.02 | 0.495 | 1.70 | 0 |
| | Exactly | 4.13e-05 | 0.024 | 0.90 | 6.28e-04 | 0.089 | 0.02 | 0.686 | 2.33 | 0 |
| Damping | Center | 0.001 | 0.031 | 0.68 | 0.015 | 0.092 | 0 | 0.578 | 1.490 | 0 |
| | Close | 4.23e-04 | 0.016 | 0.90 | 0.004 | 0.101 | 0.02 | 0.493 | 1.51 | 0 |
| | Exactly | 6.72e-05 | 0.022 | 0.94 | 0.001 | 0.090 | 0.10 | 0.616 | 2.08 | 0 |
| Reflecting | Center | 0.002 | 0.047 | 0.4 | 0.027 | 0.236 | 0 | 0.779 | 1.99 | 0 |
| | Close | 0.003 | 0.058 | 0.26 | 0.025 | 0.251 | 0 | 1.000 | 3.32 | 0 |
| | Exactly | 1.15e-07 | 0.034 | 0.72 | 6.60e-05 | 0.269 | 0.12 | 1.412 | 3.11 | 0 |

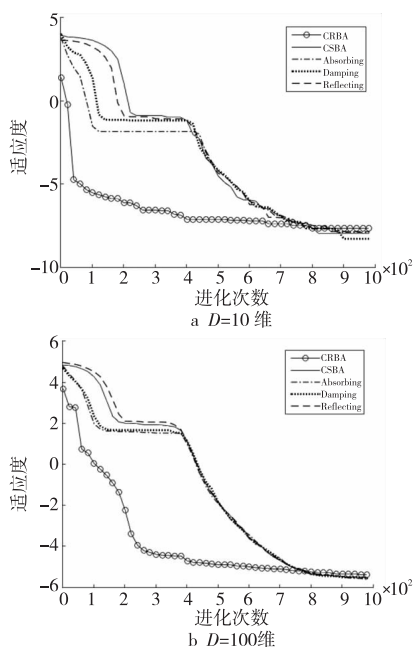


Figure 1 Optimization curve of Sphere function

图 1 Sphere 函数优化曲线

对 Sphere 函数来说,大部分算法都能轻松达到优化效果。从图 2 可以看出,CRBA 算法的收敛速度明显快于对比算法。Griewank 函数的特性随着问题维数增加优化难度降低,VSABA、Absorbing、Damping 和 Reflecting 非常依赖问题的维数。CRBA 算法在不同的维度下均能达到理想的优化效果,可见 CRBA 算法的优化能力与问题维数关系不大,具有更强的适应性。从 Rosenbrock 函数

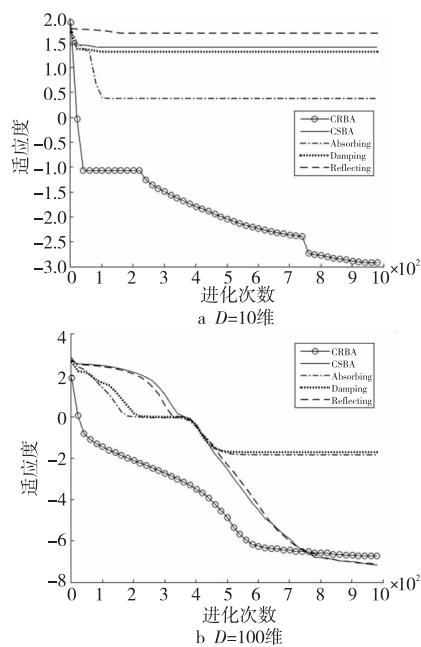


Figure 2 Optimization curve of Griewank function

图 2 Griewank 函数优化曲线

的优化结果来看,对比算法不能通过搜索信息来辨别搜索方向,从而难以极小化,而 CRBA 算法能达到相当高的收敛精度。从 Ackley 函数算法的优化结果来看,只有最优解靠近边界时,对比算法才有一定的概率寻得全局最优。CRBA 的优化不依赖于最优解与边界的位置关系,无论最优解位于解空间中心还是边界,CRBA 算法均能轻松达到优化效果。此外,CRBA 算法的持续进化和全局寻优能力

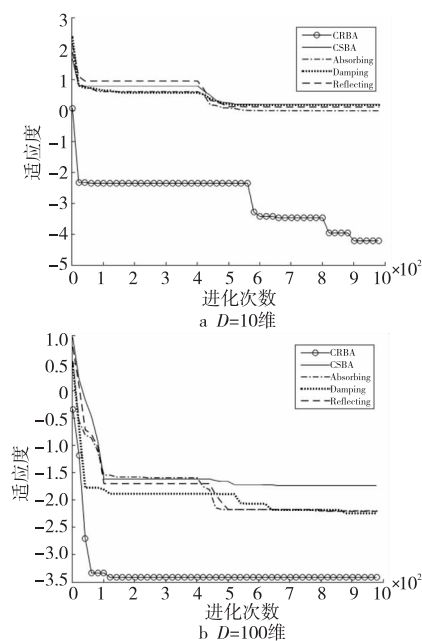


Figure 3 Optimization curve of Quartic Noise function

图 3 Quartic Noise 函数优化曲线

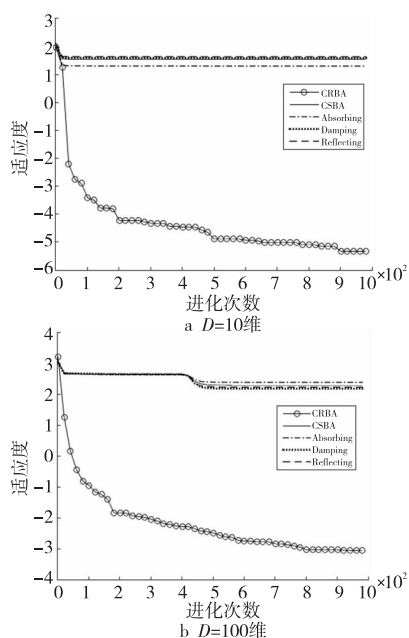


Figure 4 Optimization curve of Rastrigin function

图 4 Rastrigin 函数优化曲线

并没有随着问题维度的增加而停滞,因此更适应于高维问题的优化。Rastrigin 函数是一个含有大量局部最优点的复杂多峰函数,随着维度的增高,VSABA、Absorbing、Damping 和 Reflecting 逐渐失去寻优能力。CRBA 算法在保证收敛精度的同时也保证了寻优成功率。Quartic Noise 函数的优化结果显示了 CRBA 算法优化含大量噪声问题能力的优越性。

分析 CRBA 算法优化能力出众的原因:CRBA 算法的越界随机重置策略减少了前期个体撞界的

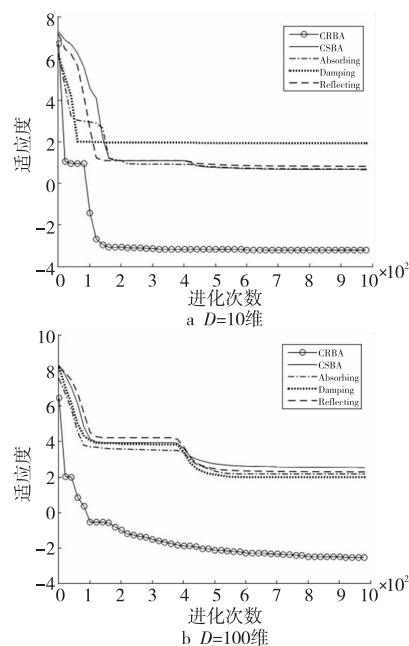


Figure 5 Optimization curve of Rosenbrock function

图 5 Rosenbrock 函数优化曲线

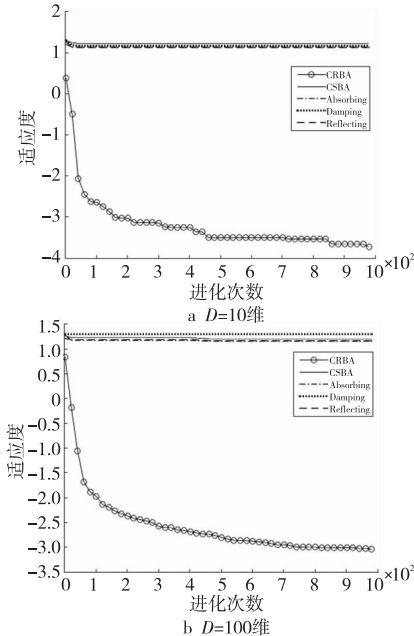


Figure 6 Optimization curve of Ackley function

图 6 Ackley 函数优化曲线

数量和概率,保证尽可能多的个体对整个种群提供有效的搜索信息。将越界粒子随机重置,使粒子很大概率能够搜索到未到达的区域,有利于粒子跳出局部最小值点。CRBA 算法控制最优解附近的局部搜索的范围随着进化次数增加而缩小。在搜索前期,算法局部搜索的范围大,很容易定位到所感兴趣的搜索区域,保证了种群前期进化、快速收敛。种群进化趋于停滞时,高斯变异策略使种群的个体以最优解为中心向四周呈放射状搜索,且越界重置策略降低了个体撞上边界的几率,提高了搜索效

率。蝙蝠个体逐渐远离当前最优解,去开发未曾搜索过的区域,一旦发现新的最优解,种群又快速聚集进行局部搜索。正是 CRBA 算法的随机重置、快速聚集、快速发散的特点,才使种群能够快速定位到可能存在最优解的区域,并进行充分的局部开发。

5 结束语

本文分析和对比了常见的几种边界变异策略下蝙蝠算法的优化效果,提出了一种基于越界随机重置和高斯变异的蝙蝠算法。该算法既保证了收敛精度,又具有较强的持续进化能力和全局寻优能力。实验表明,本文所提出的算法不依赖于问题的维度和最优解与解空间的位置关系,在解决单峰、多峰和含噪声变量的函数优化问题中都表现了较强的鲁棒性。实验中发现,当收敛精度设定在一定范围内时,CRBA 算法的全局寻优能力较强。若设置过高的收敛精度,CRBA 算法的寻优成功率会大大降低。所以,未来的研究方向是在保证全局收敛的前提下,进一步提高收敛精度。

参考文献:

- [1] Yang X, Gandomi A H. Bat algorithm: A novel approach for global engineering optimization [J]. Engineering Computations, 2012, 29(5): 464-483.
- [2] Kennedy J, Eberhart R. Particle swarm optimization [C] // Proc of IEEE International Conference on Neural Networks, 1995: 1942-1948.
- [3] Robinson J, Rahmat-Samii Y. Particle swarm optimization in electromagnetics [J]. IEEE Transactions on Antennas & Propagation, 2004, 52(2): 397-407.
- [4] Huang T, Mohan A S. A hybrid boundary condition for robust particle swarm optimization [J]. IEEE Antennas & Wireless Propagation Letters, 2005, 4(1): 112-117.
- [5] Xu S, Rahmat-Samii Y. Boundary conditions in particle swarm optimization revisited [J]. IEEE Transactions on Antennas & Propagation, 2007, 55(3): 760-765.
- [6] Chi Yu-hong, Sun Fu-chun, Wang Wei-jun, et al. An improved particle swarm optimization algorithm with search space zoomed factor and attractor [J]. Chinese Journal of Computers, 2011, 34(1): 115-130. (in Chinese)

- [7] Yue Xiao-xue, Zheng Yun-shui, Lin Jun-ting. Bat algorithm with adaptive mutation [J]. Computer Measurement & Control, 2015, 23(2): 516-519. (in Chinese)
- [8] Gao Shan, Ma Liang, Zhang Hui-zhen. Niche bat algorithm for function optimization [J]. Mathematics in Practice and Theory, 2014, 44(15): 253-260. (in Chinese)
- [9] He Zi-kuang, Wu Wei-min, Lin Zhi-yi. Hybrid bat algorithm and its application [J]. Computer Engineering and Design, 2016, 37(11): 3112-3117. (in Chinese)
- [10] Zhang Yu-nan, Liu Fu-yong. Improved variation step adaptive bat algorithm and application [J]. Journal of Guangxi University for Nationalities (Natural Science Edition), 2013, 19(2): 51-54. (in Chinese)

附中文参考文献:

- [6] 迟玉红, 孙富春, 王维军, 等. 基于空间缩放和吸引子的粒子群优化算法 [J]. 计算机学报, 2011, 34(1): 115-130.
- [7] 岳小雪, 郑云水, 林俊亭. 自适应变异的蝙蝠算法 [J]. 计算机测量与控制, 2015, 23(2): 516-519.
- [8] 高珊, 马良, 张惠珍. 函数优化的多小生境蝙蝠算法 [J]. 数学的实践与认识, 2014, 44(15): 253-260.
- [9] 何子旷, 吴伟民, 林志毅. 混合蝙蝠算法及其应用 [J]. 计算机工程与设计, 2016, 37(11): 3112-3117.
- [10] 张宇楠, 刘付永. 一种改进的变步长自适应蝙蝠算法及其应用 [J]. 广西民族大学学报(自然科学版), 2013, 19(2): 51-54.

作者简介:



李永恒(1992-), 男, 山东烟台人, 硕士生, 研究方向为智能优化计算。E-mail: ahengjiqing@163.com

LI Yong-heng, born in 1992, MS candidate, his research interest includes intelligent optimization computing.



赵志刚(1973-), 男, 广西桂林人, 博士, 教授, CCF 会员(E200018026M), 研究方向为人工智能和优化计算。E-mail: zzg-mail2002@163.com

ZHAO Zhi-gang, born in 1973, PhD, professor, CCF member(E200018026M), his research interests include artificial intelligence, and optimization computing.