

一种改进的自适应混合型蝙蝠算法

杜艳艳, 刘 升

(上海工程技术大学 管理学院 上海 201620)

摘 要: 针对基本蝙蝠算法(BA)存在的寻优精度不高,易出现早熟收敛等缺陷,本文提出了一种改进的自适应混合型蝙蝠算法(YSBA)。首先,该算法舍弃了速度这一参数,简化了计算;其次,加入位置收缩因子 β ,用来控制与约束蝙蝠的位置,平衡蝙蝠算法中局部和全局搜索,提高算法的寻优精度,最后,重新设置了响度 A 和脉冲频率 r 的计算方法,此方法可以也可以避免陷入局部最优。最后通过 11 个典型的基准函数优化试验,与基本蝙蝠算法(BA)以及采用机动飞行的蝙蝠算法(MFBA)相比,发现改进的自适应混合型蝙蝠算法能够解决局部过分搜索的问题,避免陷入局部最优值,具有较高的计算精度。

关键词: 蝙蝠算法;收缩因子;优化函数;全局优化

中图分类号: TP183

文献标识码: A

文章编号: 1000-7180(2018)06-0135-06

An Efficient Adaptive Improved-bat Algorithm

DU Yan-yan, LIU Sheng

(School of Management, ,Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: Aiming at the existence of basic bat algorithm (BA) optimization accuracy is not high, traps into local optima easily. This paper presents a new improved bat algorithm, which is named YSBA. In this algorithm, firstly, to simplify the calculation and improve the convergence speed, a new search equation is proposed in generate new solutions. Secondly, location constrict factor is added, which can be used to control with the location of the bats, balance the global and local search of bats and improve the optimization precision of the algorithm. Finally, reset the method of the calculation of loudness and rate, which can also be used to avoid trapping into local search. To verify the performance of our algorithm, 11 typical experiments are employed. The experimental results show that the new algorithm (YSBA) is significantly improved, which includes optimization accuracy, convergence speed, and they can also avoid falling into a local optimum.

Key words: Bat-inspired Algorithm, Gauss mutation, Lévy flights, global optimization

1 引言

20 世纪 50 年代中期,人们从生物进化的一些行为特征中受到启发,创立了优化中的仿生学,进而演化出群体智能优化算法。

蝙蝠算法^[1] (Bat Algorithm, BA)采用频率调谐的技术来增强种群的多样性,利用迭代过程中脉冲响度和脉冲发射频率的适时改变来实现算法的全

局搜索和局部搜索的自动切换,从而平衡了全局搜索和局部搜索对算法寻优性能的影响。

BA 算法属于新兴的元启发式算法,因其特有的优点,近几年备受研究者的关注,逐渐成为智能计算领域的热点问题,相关的科研成果也日益增加,在 FJSP 调度问题^[2]、函数优化^[3,4]、无线传感器、云制造供应链^[5]等诸多方面的研究成果日益浮现。

本文针对基本 BA 的不足提出了一种新的改进

算法 YSBA (An Efficient Adaptive improved-bat Algorithm), 最后通过实验仿真结果表明, 该算法各种性能得到明显提高。

2 基本蝙蝠算法

BA 流程如下:

(1) 参数初始化: 群体规模 N 、目标函数 $f(X)$, 响度衰减度 α 、脉冲发射率增加系数 g 、脉冲发射率 R_0 、响度 A_0 、声波频率 f_i ;

(2) 蝙蝠初始位置为 X_i , 初始速度 V_i , 寻找当前最佳蝙蝠的所处位置 X_0 , 并计算其适应度 $f(X_0)$;

(3) 按照(1)-(3) 更新蝙蝠的速度和当前位置;

$$V_i^t = V_i^{t-1} + [X_i^t - X_0] f_i \quad (1)$$

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (2)$$

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

(4) 如果随机数 $rand > R_0$, 通过(4) 式进行随机游走, 产生新解 X_{new} ;

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

(5) 如果均匀分布随机数 $rand < A_0 \& f(X_{new}) < f(X_i)$, 则接受这个新解, 让蝙蝠飞至更新之后的位置, 并利用公式(5), (6) 更新响度 A_i 和脉冲发射率 R_i

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

$$R_i^t = R_i^0 [1 - \exp(-\gamma t)] \quad (6)$$

(6) 排列所有蝙蝠, 并进行评估, 找出当前最佳蝙蝠所处位置及其适应度值;

(7) 满足结束条件, 则进入下一步. 如未满足终止条件, 则返回 3);

(8) 输出最优的蝙蝠个体和全局最优解。

3 一种有效的自适应的改进蝙蝠算法

3.1 算法的简化部分

基本蝙蝠算法中的当前速度和位置主要依据其自身上一次的速度、位置和蝙蝠群的全局最优值这三个变量, 当全局最优值在后期收敛于某一局部时, 蝙蝠算法将只在这一区域进行局部遍历, 在后期搜索时, 蝙蝠在局部搜索区域过分迭代。

因此, 本文去除基本蝙蝠算法公式中的速度项, 而利用随机位置和最优位置更新整个搜索过程, 从而减轻了局部过分搜索的情况. 位置的计算方法是依靠基本蝙蝠算法的每次算法迭代的全局最优位置 $best$, 并随机产生一个新的位置 $prand$, 这两个参数根据公式(7)~(13)得到, 其中 $prand$ 在一定程度上增加了蝙蝠的多样性, $best$ 保证了在最优位置附近

搜索, 其更新公式如下:

$$X_i^t = \epsilon * (X_i^{t-1} + f_{2t} * rand * (best - X_i^{t-1}) + f_{1t} * rand * (prand - X_i^{t-1})) \quad (7)$$

$$prand = X_i^{t-1} * rand() * 3 \quad (8)$$

式中, t 为当前迭代次数; ϵ 为收缩因子; f_{1t} ; f_{2t} 为频率, $best$ 为全局最优位置; $prand$ 为个体的随机位置; X_i^t 为 t 次迭代第 i 只蝙蝠的位置; 公式(7) 中第一项 X_i^{t-1} 是上一次迭代中第 i 只蝙蝠的位置; 第二项 f_{2t} 是个体学习部分, 即根据以前自己的尝试和经历, 积累的经验; f_{1t} 是追新的部分, 表示蝙蝠对随机的新的蝙蝠的信任度大小。

可以看出, 改进后的蝙蝠算法的公式中没有速度这一部分, 从而降低了算法的复杂度, 而引入随机位置可以避免局部过分搜索。

3.2 收缩因子的确认

为了有效地控制微粒的飞行速度, 使算法达到全局搜索和局部搜索两者间的平衡, Clerc 和 Kennedy 于 2002 年提出了收缩因子粒子群算法^[6], 利用收缩因子代替原来基本的粒子群算法中的惯性权重, 使得算法的整体精度提高. 在蝙蝠算法中针对算法寻优精度不高的缺陷, 本文加入了收缩因子。

蝙蝠算法在每次迭代中, 应当是全局搜索和局部搜索同时进行, 取得均衡的结果, 而在基本 BA 中, 由于速度系数的设置为 1, 比较单一, 在迭代初期对于全局搜索并未产生明显的效果, 搜索效率偏低, 在后期随着迭代次数的增加, 局部挖掘能力逐渐变弱, 所有蝙蝠都会集中在某一区域, 从而搜索精度不是很高, 因此本节采用了收缩因子, 以提高算法的搜索精度, 本节提出收缩因子 ϵ 的计算公式如下:

$$\epsilon = \frac{2}{|2 - phi - \sqrt{phi^2 - 4phi}|} \quad (9)$$

$$phi = 4 * (f_{1t} + f_{2t}) \quad (10)$$

3.3 频率的确定

在蝙蝠算法中, 蝙蝠个体发出的脉冲的大小与频率的大小有关, 而由式(7)可知, 频率的大小又间接影响位置的变化情况, 因此本文使频率随着迭代次数进行自适应变化. 在这个算法中, f 分成两个部分, 个体学习部分 f_{1t} 和追新部分 f_{2t} , 这样的设置使算法在迭代前期 f_{1t} 比 f_{2t} 大时, 根据式(7), 倾向于追新部分的学习, 即选择根据自身认知, 随机选择位置这样保证了蝙蝠的全方位搜索的能力; 在蝙蝠算法后期进行局部搜索 f_{2t} 比 f_{1t} 大时, 就会倾向选择已有的最优位置, 即式(7)中第二项, 这样保证了

蝙蝠每个个体都在局部最优位置附近搜索,具有局部搜索能力.结合以上分析,本文提出了 f 的变化是随着迭代中适应度值的变化而变化的自适应方法,随着迭代次数的增加, f 的变化保证了算法在前期具有较好的全局搜索能力,在后期具有较好的局部搜索能力,其中频率 $f_i(i=1,2)$ 的变化公式如下所示:

$$f1t = 1 - \exp(-|Fbest(t) - Fave(t)| * rand)$$

(11)

$$f2t = 1.1 - f1t$$

(12)

$$Fave(t) = \frac{1}{N} \sum_{i=1}^N f(X_i(t))$$

(13)

3.4 响度、脉冲速率的确定

在蝙蝠搜索过程中,是通过改变脉冲 R 和响度 A 来控制的,在靠近猎物的过程中,响度 A 逐渐减小,而脉冲频率 R 逐渐增加,通过脉冲发射率增加系数 a 来控制脉冲增加率,通过衰减系数 γ 来减小响度,这样使得蝙蝠在靠近猎物时不易被猎物察觉,而且可以准确掌握猎物的动向,从而方便抓住猎物,由此可以看出,响度 A 、频率 R 在一定程度上存在某种关系,因此本文通过如下的公式表示 A 和 R :

$$A = A_0 * (\exp(-rand * t))$$

(14)

$$R = R_0 * (1 - A/A_0)$$

(15)

4 YSBA 算法实施步骤

本文提出的 YSBA 的具体实施步骤如下:

Step1:参数初始化:群体规模 N 、维数 D ,目标函数 $f(x)$,初始脉冲发射率 R_0 、初始响度 A_0 、声波频率 $f1t$ 、 $f2t$;

Step2:蝙蝠初始位置为 X_i ,寻找当前最佳蝙蝠所处位置 $best$,计算其适应度 F_{best} ,并且计算蝙蝠的平均适应度值 F_{ave} ;

Step3:利用公式(7)–(13)求解出新的蝙蝠位置 Xt_i ,同时限制蝙蝠的边界位置.

Step4:如果均匀分布随机数 $rand > R_0$,蝙蝠个体此最佳解附近区域形成一个局部解,并进行随机游走,产生新解 X^* 代替当前解,否则跳过此步;

Step5:如果均匀分布随机数 $rand < A$ & $f(X^*) < f(X_i)$,则接受这个新解,并让蝙蝠飞至更新之后的位置,否则不更新;

Step6:与最优蝙蝠比较,如果 $f(X^*) < F_{best}$,替换之前最优蝙蝠,并按照公式(14)、(15)更新响度 A 和脉冲发射率 R ;

Step7:排列所有蝙蝠,并进行评估,找出当前最

佳蝙蝠所处位置即最优解;

Step8:满足结束条件,则进入下一步.如未满足终止条件,则返回 Step2,

Step9:输出最优的蝙蝠个体和全局最优解.

5 实验性能测试

5.1 实验测试函数

为了证明增加收缩因子的自适应蝙蝠算法的有效性,本文在借鉴文献^[6]的基础上,将 YSBA 与 MFBA^[6](bat algorithm by using maneuver flight, MFBA)、基本 BA 进行了性能对比分析,选择 11 个典型的基准函数优化问题进行验证,由于每个测试函数均有其自身特点,每个算法不能适用于所有的测试函数,不同性质的函数得到的实验结果可以较全面的展示算法的性能.

文中选取的函数基本分为下面三类: $f_1(x) \sim f_2(x)$ 为高维单峰函数, $f_3(x) \sim f_7(x)$ 为高维多峰函数, $f_8(x) \sim f_{11}(x)$ 为低维函数.表 1 列出了所有函数的名称和对应表达式,维度,搜索空间,理论最优值 $fmin$.

为了增强可比性,在本次算法性能测试中,MFBA,BA 以及 YSBA 算法的参数设置应尽可能保持一致.参数取值设置如表 2 所示:

5.2 GMBA 收敛性测试结果与分析

5.2.1 固定迭代次数的寻优精度及收敛速度

为了证明新提出的 YSBA 算法的有效性,选用基于上述 11 个函数的最小化问题在 Matlab R2014a 中对每个测试函数进行 50 次独立实验,从而防止算法的偶然性带来的误差.实验结果如表 3 所示,同时还显示最优值、最差值以及平均值.其中,最优值反映了算法运行至最大迭代次数时的求解精度,平均值反应了解的收敛稳定程度.从表 3 可以看出,在三种算法中,YSBA 的适应度平均值、最优值、最差值都相对较低,寻优效果最佳, $f_3(x)$, $f_5(x)$, $f_8(x)$, $f_{10}(x)$ 都得到了理论最优值,对于函数 $f_1(x)$, $f_2(x)$, $f_6(x)$, $f_7(x)$, $f_{11}(x)$,其寻优最优值都提高了至少 65,97,39,116 个数量级.可以看出 YSBA 在收敛精度、收敛稳定性等方面均优于 BA 和 MFBA.

从收敛曲线对比图 1-11 可以看出(由于 $f_1(x)$, $f_2(x)$, $f_3(x)$, $f_7(x)$, $f_{11}(x)$ 理论提高较大,本文对于函数适应度值取以 10 为底的对数,以便达到让迭代曲线的显示和观察更为便捷的目的),YSBA 比 BA, MFBA 收敛速度快,优化精度高,迭代

表 1 测试函数

编号	名称	函数表达式	维度	搜索范围	fmin
$f_1(x)$	Shpere	$\sum_{i=1}^n x_i^2$	30	$[-5.12, 5.12]$	0
$f_2(x)$	Quartic	$\sum_{i=1}^d x_i^4$	30	$[-1.28, 1.28]$	0
$f_3(x)$	Rastrigin	$\sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
$f_4(x)$	Ackley	$20 + e - 20\exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2})\exp(\frac{1}{d}\sum_{i=1}^d \cos(2\pi x_i))$	30	$[32.768, 32.768]$	0
$f_5(x)$	Griewank	$\frac{1}{4000}\sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\frac{x_i}{\sqrt{i}} + 1$	30	$[-600, 600]$	0
$f_6(x)$	Alpine	$\sum_{i=1}^n x_i \sin(x_i + 0.1x_i) $	30	$[-10, 10]$	0
$f_7(x)$	Schwefel's	$\sum_{i=1}^d x_i + \prod_{i=1}^d x_i $	30	$[-500, 500]$	0
$f_8(x)$	Shaffersf7	$(x_1^2 + x_2^2) \cdot 1/4 [1.0 + \sin 2(50(x_1^2 + x_2^2) \cdot 1/10)]$	2	$[-5.12, 5.12]$	0
$f_9(x)$	drop wave	$-(1 + \cos(12 \cdot \sqrt{\sum_{i=1}^n x_i^2}) / \frac{1}{2}(\sum_{i=1}^n x_i^2) + 2)$	2	$[-5.12, 5.12]$	-1
$f_{10}(x)$	Exponential	$-\exp(0.5 \sum_{i=1}^n x_i^2)$	2	$[-5.12, 5.12]$	-1
$f_{11}(x)$	Axis	$\sum_{i=1}^d ix_i^2$	10	$[-5.12, 5.12]$	0

表 2 参数设置表

	N	A	R	aa	gg	Iter
BA	40	0.5	0.5	Null	Null	500
MFBA	40	Null	Null	0.9	0.9	500
GMBA	40	0.5	0.5	Null	Null	500

次数少(除图 4 优化效果不是特别大外). YSBA 不管对于多峰函数还是单峰函数,在迭代初期就表现出良好的性能,测试函数的迭代曲线近似直线,下降速度很快,表明能很快找到全局最优解,并不断向最优解靠近,达到很高的精度.

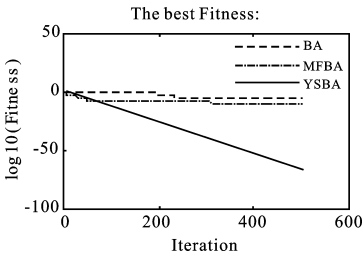


图 1 F1 函数收敛曲线对比

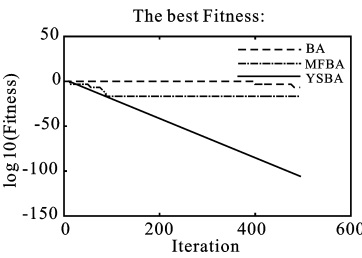


图 2 F2 函数收敛曲线对比

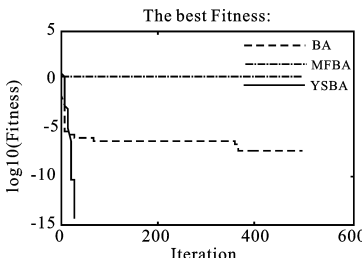


图 3 F3 函数收敛曲线对比

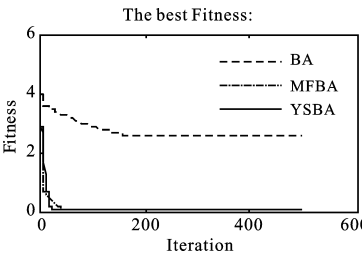


图 4 F4 函数收敛曲线对比

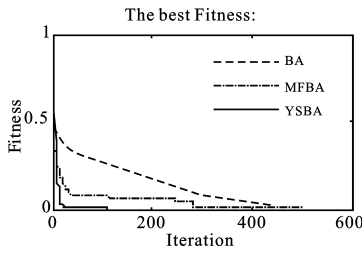


图 5 F5 函数收敛曲线对比

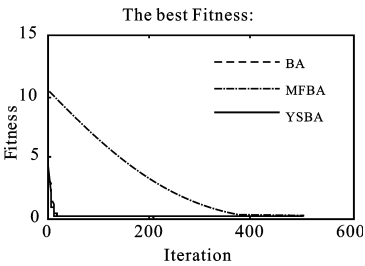


图 6 F6 函数收敛曲线对比

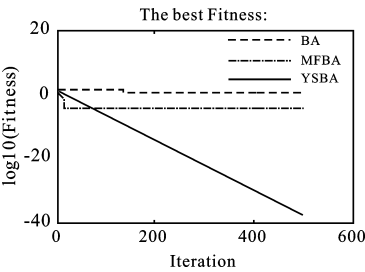


图 7 F7 函数收敛曲线对比

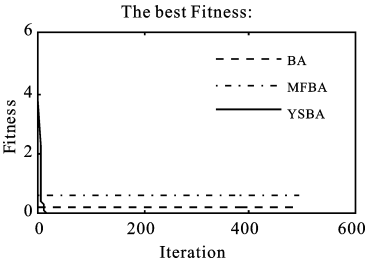


图 8 F8 函数收敛曲线对比

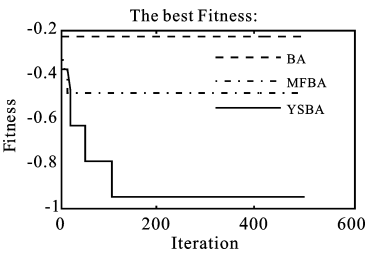


图 9 F9 函数收敛曲线对比

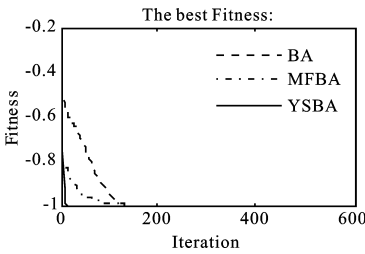


图 10 F9 函数收敛曲线对比

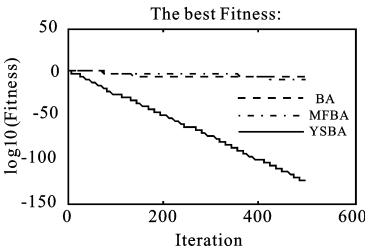


图 11 F11 函数收敛曲线对比

5.2.2 YSBA 在高维函数上的优化性能测试实验

仿生群智能优化算法普遍存在易陷入局部最优,以至于后期多样性差,收敛速度慢,寻优精度低,尤其是对高维、多峰函数的复杂优化问题,以平均最优值以及相对变化率作为算法性能的评价指标,将 f_1, f_5, f_9 的维度 50 维为间隔从 50 到 300 依次改变,分别采用 BA, MFBA, YSBA 三种算法对不同维度的 5 个函数进行最小值寻优,以验证 YSBA 在高维函数上的优化性能随函数维度的增加而变化的情况,此实验依然独立进行 30 次,将其平均最优值及相对变化率列入表 4,其中相对变化率 $R = |50 \text{ 维平均最优值} - 300 \text{ 维平均最优值}| / 50 \text{ 维平均最优值}$,由表 4 数据可知,随着函数维度的增加,各种算法的优化性能均有一定程度的降低,即平均最优值增大;但是无论维度如何变化,YSBA 寻得的最优值依然是三种算法中最小的,即优化性能最好。

表 4 算法在高维函数上的性能比较

函数	算法	D=50	D=100	D=150	D=200	D=250	D=300	RC
$f_1(x)$	BA	9.428e-05	9.1463	46.6424	62.0084	84.6030	1.169e+02	1.240e+06
	MFBA	2.166e-04	5.049e-04	1.136e-03	1.370e-03	5.147e-02	25.3481	1.169e+05
	YSBA	8.188e-76	6.005e-55	6.341e-47	3.713e-40	5.732e-35	5.315e-30	6.491e+45
$f_5(x)$	BA	0.063 0	0.198 3	0.394 1	40.645 4	74.448 4	1.559e+03	2.474e+04
	MFBA	0.071 3	0.268 5	0.002 2	4.351 8	25.051 5	76.064 3	1.065e+03
	YSBA	0	0	0	0	0	0	0
$f_9(x)$	BA	-0.985 8	-4.460e-4	-5.220e-8	-7.63e-16	-1.07e-17	-5.84e-24	1
	MFBA	-0.999 99	-0.999 99	-0.899 99	-0.999 99	-0.999 99	-0.999 99	0
	YSBA	-1	-1	-1	-1	-1	-1	0

综上所述,本文算法比 BA 与 MFBA 算法具有更高的收敛可靠性、更高的收敛精度及更快的收敛速度,相比于参考文献中的算法也具有更高的寻优精度.

6 结束语

本文在 BA 算法的基础上,利用收缩因子平衡蝙蝠算法中局部和全局搜索,提出了一种改进的自适应混合型蝙蝠算法,同时简化了位置计算公式.实验对比分析说明,改进后的 YSBA 算法在求解精度、全局寻优上都要优于 BA 算法和 MFBA 算法,这在很大程度上避免了陷入局部最优,具有较强的全局搜索能力.但是由于随机位置的不确定性,实验的最差值不是特别理想,因此如何选择合适参数,优化最差位置的值,将是下一步研究的内容.

参考文献:

[1] Yang X-S. A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization [M]. Berlin:Springer, 2010: 65-74.

[2] 徐华,张庭,包哲人,等. 求解柔性作业车间调度问题的改进蝙蝠算法[J]. 信息与控制,2016,45(6): 722-728.

[6] 周峰,刘辉,李超峰. SIFT 算法在图像配准中的应用[J]. 办公自动化,2009,168(9):40-41.

[7] 李永明. 基于 Hessian 矩阵的边缘检测方法[J]. 激光,2014,35(5):27-33.

[8] 冉秋,袁超. 引入 Hessian 矩阵尺度极值检测的图像角度特征匹配[J]. 科技通报,2014,30(8):86-88.

[9] 刘影,陆安江,张正平. 基于曲线拟合的 SIFT 特征精确定位算法[J]. 通信技术,2013,46(1):92-94.

[10] 王琰,丁明明. 基于熵的柱面全景图拼接算法及其鲁棒性分析[J]. 小型微型计算机系统,2007,28(1): 148-150.

[11] 丁湘楠. 基于 LDA 的 SIFT 算法在遥感图像配准中的研究与应用[D]. 江苏,江苏科技大学,2014.

[12] 庄志国,孙惠军,董继扬等. 基于角点检测的图像匹配算法及其在图像拼接中的应用[J]. 厦门大学学报,2007,46(4):501-505.

[3] GolmaryamiM, SalehS, ArdekaniA, et al. A new modified bat algorithm to solve optimal management of multi-objective reconfiguration problem[J]. Journal Of Intelligent & Fuzzy Systems, 2014, 27 (3): 1567-1573.

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

[5] 张勇凯,李芳,包晓晓. 改进蝙蝠算法在云制造供应链中的应用[J]. 数学理论与应用,2015,5(2):83-94.

[6] Clerc M, Kennedy J. The Particle swarm-explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Transactions on Evolutionary Comput7tion, 2002, 6(1):58-73.

[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):2962-2964.

作者简介:

杜艳艳 女,(1993-),硕士研究生. 研究方向为群智能算法、计算机应用.

刘升(通信作者) 男,(1966-),博士,教授. 研究方向为计算机应用、智能计算、人工智能.

E-mail:845714323@qq. com.

(上接第 134 页)

[13] Richard Szeliski. Video mosaics for virtual environments[J]. IEEE Computer Graphics and Applications, 1996, 16(2): 22-30.

[14] Richard Hartley, Andrew Zisserman. Multiple View Geometry in Computer Vision [M]. Cambridge : Cambridge University Press , 2003,23-150.

[15] Brandt S. Maximum Likelihood Robust Regression with Known and Unknown Residual Models[C]// Proceedings of the ECCV Copenhagen, [S. L.]:2002;97-102.

作者简介:

尚明珠 女,(1977-),硕士,实验师. 研究方向为图像处理、计算机视觉. E-mail:shj417@163. com.

王克朝 男,(1980-),博士,副教授. 研究方向为图像处理、软件工程.