

改进的新型蝙蝠算法

吕赵明, 张颖江

(湖北工业大学 计算机学院 湖北 武汉 430068)

摘 要: 对基本的蝙蝠算法及其他群智能优化算法进行试验研究后,发现基本蝙蝠算法存在易陷入局部最优、求解精度不高等缺陷,而细菌觅食算法具有群体智能算法并行搜索、易跳出局部极小值等优点。由于基本蝙蝠算法对求解的空间搜索不充分,通过实验分析提出了试探扰动因子;另外针对蝙蝠算法缺乏对父代的继承性,分别使用线性递减权重法、随机权重法和自适应权重法对蝙蝠算法求解性能进行了对比实验,发现随机惯性权重求解精度较高。基于以上分析,提出了一种改进的新型蝙蝠算法。该算法融合细菌觅食算法的趋化算子来改进蝙蝠算法的局部搜索能力,增加试探扰动因子来提高算法的求解精度和充分性,采用随机惯性权重来均衡算法的探索能力和挖掘能力。为了验证该算法的性能,选择几个高维的经典函数进行实验,结果表明,改进的新型蝙蝠算法同基本的蝙蝠算法和粒子群算法相比提高了寻优性能。

关键词: 蝙蝠算法; 细菌觅食算法; 翻滚; 惯性权重; 扰动因子

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2018)05-0063-05

doi: 10.3969/j.issn.1673-629X.2018.05.015

An Improved New Bat Algorithm

LYU Zhao-ming ZHANG Ying-jiang

(School of Computing, Hubei University of Technology, Wuhan 430068, China)

Abstract: Based on the experimental study of the basic bat algorithm and other groups of intelligent optimization algorithms, it is found that the basic bat algorithm is easy to fall into the local optimum and the accuracy of the solution is not high. The bacterial foraging algorithm has the advantages of parallel search and jumping out of local minimum easily from group intelligence algorithm. Because the basic bat algorithm is not sufficient to search the solution space, the probe disturbance factor is put forward. In addition, the bat algorithm lacks the inheritance of the parent. For this, we use respectively the linear descent inertial weight, the random inertial weight and the adaptive inertial weight to compare the performance of the bat algorithm, which shows that the accuracy of the random inertia weight is highest. Based on the above analysis, we propose an improved new bat algorithm which combines the chemotaxis operator of the bacterial foraging algorithm to improve the local search ability of the bat algorithm, increases the probabilistic perturbation factor to improve the accuracy and sufficiency of the algorithm and adopts the inertia weight to equalize ability of exploer and exploit in the algorithm. To verify the performance of the improved algorithm, several high quality classical functions are selected to test. The experiments show that the proposed bat algorithm improves the performance compared with the basic bat algorithm and particle swarm algorithm.

Key words: bat algorithm; bacterial foraging algorithm; roll; inertia weight; disturbance factor

0 引言

在许多科学计算中都会遇到最优化问题,最优化问题是通过无限可行的方案进行寻优从而得到最优的方案。传统的优化算法,如线搜索、非线性搜索、线性规划和整数规划等,对函数有一定要求,对高维、非线性的问题计算复杂度大,在实际工程中不适用。20世纪50年代中期一些研究者摆脱了经典数学优化方法,受生物群体行为特点、觅食方法等机理的启发,提

出了仿生智能优化算法,如进化算法、粒子群算法、人工蜂群算法等,对传统算法无法处理的问题可以求解到较好的结果。

Yang Xinshe 教授受蝙蝠利用回声定位的能力在黑暗中捕食猎物和躲避障碍物行为的启发,在2010年提出了蝙蝠算法(BA)^[1-2]。在对基本的蝙蝠算法进行试验研究后,发现基本蝙蝠算法存在易陷入局部最优、求解精度不高等缺陷。文献[3]通过自适应改变

收稿日期: 2017-06-15

修回日期: 2017-10-18

网络出版时间: 2018-02-08

基金项目: 教育部基金项目(NGII20150404)

作者简介: 吕赵明(1990-),男,硕士研究生,研究方向为智能计算、网络安全;张颖江,教授,研究方向为网络安全。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20180207.1917.084.html>

蝙蝠的步长,提高了算法的求解精度。文献[4]把差分进化算法应用到蝙蝠算法中,提高了算法的性能。文献[5]提出每只蝙蝠动态自适应地调整自己的飞行速度和方向,不仅提高了全局搜索性能,而且有效避免了过早收敛。

在上述研究的基础上,文中提出了基于细菌觅食算法算子改进的新型蝙蝠算法。为了避免陷入局部最优,融合细菌觅食算法的趋化算子来提高算法的局部搜索能力;为了提高算法的求解精度,增加了试探扰动因子来充分搜索解空间,采用随机惯性权重^[6]来平衡算法的探索能力和挖掘能力。

1 蝙蝠算法

基本蝙蝠算法的仿生原理是模仿蝙蝠的回声定位的机制设计的仿生智能优化算法。该算法将空间中捕食的蝙蝠模拟为问题的解,将蝙蝠不断变化的位置模拟为问题函数的适应度值,将蝙蝠捕食过程中不断变化的频率、脉冲速率、音强模拟为解的优化搜索过程,整个群体的运动在问题求解空间中产生从无序到有序的演化过程^[1-2]。

具体的仿真数学模型为蝙蝠在 d 维空间中位置和速度的更新,通过以下公式实现:

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

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

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

式1中通过最大频率和最小频率来调整当前蝙蝠 i 的频率,其中 β 为 $[0, 1]$ 之间的随机数;通过式2调整当前蝙蝠 i 的速度,其中 v_i^{t-1} 和 x_i^t 分别是蝙蝠 i 在 $t-1$ 和 t 代的速度和位置, x^* 表示当前搜索过程中蝙蝠所处的最佳位置;通过式3更新所处位置, x_i^t 和 x_i^{t-1} 分别表示蝙蝠 i 在 t 和 $t-1$ 代的位置。当蝙蝠进入局部搜索时,一旦从当前最优解中选择一个解,每只蝙蝠通过随机游走的方式产生一个新解,其表示公式为:

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

其中, $\varepsilon \in [-1, 1]$; $A^t = \langle A_i^t \rangle$ 为每代中所有蝙蝠响度的平均值。

在迭代过程中,一旦蝙蝠发现了猎物,其脉冲发射的响度 A_i 通常会减小,而脉冲速率 r_i 会相应增加。其调整方式表示为:

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

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

其中, α 为音强的衰减系数; γ 为脉冲速率增加系数。对于任何 $0 < \alpha < 1$ 和 $\gamma > 0$ 的量都有:

$$A_i^t \rightarrow 0, r_i^t \rightarrow r_i^0 \text{ as } t \rightarrow \infty \quad (7)$$

通过多次试验,发现 $\alpha = \gamma = 0.9$ 效果较好。初始化时每只蝙蝠都应该有不同的音强和脉冲速率,如果

新解被更新,它们的音强和脉冲速率将相应更新,也就意味着蝙蝠正在向最优解移动。

基本蝙蝠算法的计算步骤如下:

步骤1: 设置算法的基本参数,包括种群大小、空间维数、音强衰减系数 α 、脉冲频率 f_i 、脉冲速率增加系数 γ 、最大音强 A 、迭代次数;

步骤2: 初始化蝙蝠种群的位置、速度、脉冲速率 r_i 和音强 A_i ,计算每只蝙蝠的适应度值,找出当前蝙蝠的最佳位置和适应值;

步骤3: 通过式1~3调整频率来更新蝙蝠的位置和速度;

步骤4: 根据产生的随机数,如果 ($\text{rand} > r_i$) 对当前最佳解的位置进行随机扰动产生新位置,并计算新解;

步骤5: 根据产生的随机数和产生的新解,如果 ($\text{rand} < A_i \& f(x_i) < f(x^*)$) 成立,则接受这个新解并根据式5和式6增大 r_i ,减小 A_i ;

步骤6: 判断新解是否优于全局最优解,若优于全局最优解,则更新最优解和最优蝙蝠位置;

步骤7: 判断是否满足最大迭代次数,不满足转步骤3;

步骤8: 对结果进行分析输出,算法结束。

2 细菌觅食算法

细菌觅食算法 (bacterial foraging optimization, BFO)^[7-9] 是 K.M.Passino 在 2002 年模仿细菌在人体肠道内寻找食物的行为提出的群体智能算法,该算法有三个算子,分别是趋化算子、繁殖算子、迁徙算子。算法在进行搜索时按照三个算子的执行顺序依次执行,从而使得待求的解不断更新,不断逼近最优解。

趋化算子是算法的核心部分,决定着细菌能否找到食物源,对算法的性能和收敛性有着极其重要的影响。其中包括翻转和前进两种算子,翻转算子表示细菌向任意方向移动的模式;前进算子表示在细菌翻转后,若适应值没有得到改善而沿着同一方向移动若干步,直到达到最大移动次数或其适应度值稳定的模式。相对应的公式如下:

$$X_i(j+1, k, l) = X_i(j, k, l) + \text{rand}() \times \text{step} \times \Phi(i) \quad (8)$$

$$\Phi(i) = \frac{X_i(j, k, l) - X_{\text{rand}}(j, k, l)}{\|X_i(j, k, l) - X_{\text{rand}}(j, k, l)\|} \quad (9)$$

其中, $X_i(j, k, l)$ 表示细菌个体 i 当前的位置, j 表示细菌的第 j 代趋化算子, k 表示细菌的第 k 代繁殖算子, l 表示细菌的第 l 代迁徙算子; step 表示前进步长; $\Phi(i)$ 表示细菌的随机翻滚方向; $X_{\text{rand}}(j, k, l)$ 表示当前个体领域内的一个随机位置。

3 新型蝙蝠算法

基本蝙蝠算法在寻优过程中,当搜索到近似解后会出现停滞现象,表明此时种群没有进化能力,导致种群多样性受到限制而最终陷入局部最优解。对高维复杂函数问题就会明显地出现寻优精度不高的现象,可见种群的多样性及种群的质量对蝙蝠算法的性能起着决定性作用。基本蝙蝠算法往往是随机初始化一定数量的种群,这样会出现种群的不均匀现象,导致算法陷入局部最优。基本蝙蝠算法在调整自身速度时是通过随机频率、位置差和上一代的速度来更新当前速度的,缺乏对父代速度继承的特性,这样算法缺乏均衡搜索的能力。从算法的步骤4和步骤5可以看出,蝙蝠发现猎物后,若适应值和音强没有同时减小,则当前蝙蝠的适应度值不改变,另外若适应值没有当前最小适应值优,则丢弃该位置,这样可能跳过邻近最优解^[10-15]。

基于以上分析,文中采用细菌觅食算法的趋化算子来改善种群的均匀度,提高种群的质量,从而改进蝙蝠算法的局部搜索能力;增加随机惯性权重来提高蝙蝠对父代的继承性,从而均衡搜索能力;增加试探扰动因子来提高算法搜索的充分性和求解精度。

为了改善种群的均匀度,提高种群的质量,融合细菌觅食算法的趋化算子。细菌觅食算法的趋化算子是细菌躲避有害区域向食物源区域聚集的行为。趋化算子又包括翻转算子和前进算子,翻转算子是通过先将每只细菌进行翻滚,使其向一个随机方向前进一个步长后判断适应值是否得到改善,若适应值得到改善,则细菌继续在这个方向上前进指定的步数,若此时细菌的适应度值不再改善或达到了指定的步数,则算法结束^[8]。由此可见,前进算子使细菌个体在自己的领域范围进行了深度的寻优搜索,这样在领域内能找到更加优秀的解,从而增强了算法的局部搜索能力。在经过趋化算子后,每个个体都靠近了食物丰富的区域,改善了最初的位置,就整个群体来看,整个种群得到了均匀化,提高了整个群体的质量。

趋化算子的搜索步骤如下:

步骤1:根据式8和式9计算翻转算子;

步骤2:判断群体的适应度值是否得到改善;

步骤3:若得到改善,群体执行前进算子;

步骤4:判断群体的适应度值是否得到改善;

步骤5:若得到改善,判断是否达到指定的最大前进次数,若没有达到跳转到步骤3;

步骤6:判断趋化算子的执行次数,若达到,算法结束并输出优化后的群体,否则跳转到步骤1继续执行。

为了提高蝙蝠对父代的继承性,从而均衡搜索能力,使用下面的速度调整公式:

$$v_i^t = w^* v_i^{t-1} + (x_i^t - x^*) f_i \quad (10)$$

$$w = \mu + \sigma^* N(0, 1) \quad (11)$$

$$\mu = \mu_{\min} + (\mu_{\max} - \mu_{\min}) * \text{rand}(0, 1) \quad (12)$$

其中, μ_{\max} 、 μ_{\min} 为随机权重平均值的最大值和最小值; σ 为随机权重的方差; $N(0, 1)$ 为标准正态分布的随机数。

为了提高算法搜索的充分性和求解精度,在蝙蝠更新位置后若不满足局部搜索或飞向猎物的条件,对当前的蝙蝠进行试探扰动,保持蝙蝠在邻域内的游动状态,而不是根据速度公式飞到下一个位置,根据试探扰动的新位置的适应度值试图更新当前最佳位置及最优值。若适应值未得到改善,说明原蝙蝠(位置更新前的蝙蝠)寻优能力差,因此再通过试探扰动因子更新原蝙蝠位置,在下次迭代过程中由更新后较优的位置为起点继续搜索。这样不仅确保了蝙蝠的灵活性而且增加了搜索的全面性和求解的精度。其试探扰动因子公式如下:

$$x_{\text{new}}^* = x_i^* + x_i^* (u \times \text{rand}(1, d)) \quad (13)$$

$$x_{\text{old}}^* = x_i^* + x_i^* (u \times \text{rand}(1, d)) \quad (14)$$

其中, $u \times \text{rand}(1, d)$ 为试探扰动因子, u 为 $[0, 1]$ 之间的随机数, d 为维数。

综上所述,新型蝙蝠算法的计算步骤如下:

步骤1:设置算法的基本参数,包括种群大小、空间维数、音强衰减系数 α 、脉冲速率增加系数 γ 、最大音强 A 、最大迭代次数等;

步骤2:初始化蝙蝠种群的位置和速度、脉冲速率;

步骤3:根据式8和式9对蝙蝠群体执行趋化算子;

步骤4:计算每只蝙蝠的适应度值,找出当前蝙蝠的最佳位置和适应值;

步骤5:通过式1、式3和式10调整频率来更新蝙蝠的位置和速度;

步骤6:根据产生的随机数,如果 $(\text{rand} > r_i)$ 对当前最佳解的位置进行随机扰动,产生新位置,并计算新解;

步骤7:根据产生的随机数和新解,如果 $(\text{rand} < A_i \& f(x_i) < f(x_*))$ 成立,接受这个新解并增大 r_i 、减小 A_i ;

步骤8:若步骤7不成立,则根据式13对当前蝙蝠进行试探扰动并计算新解,若新解优于当前最优解,则更新蝙蝠当前位置和最优解,否则根据式14进行试探扰动,更新原蝙蝠位置;

步骤9:判断是否满足最大迭代次数,不满足转步骤5;

步骤10:对结果进行分析输出。

4 实验

4.1 实验环境和测试函数

实验采用 Windows7 64 位操作系统,处理器为 Intel(R) Core(TM) i3-2120 3.30 GHz,内存 4 G,编程实验环境为 MatlabR2013a。

为了验证改进的新型蝙蝠算法(WRBFBA)的寻优性能,与基本的 BA 算法和基本的粒子群算法进行对比,从经典的测试函数中选取 5 个不同维数的函数进行实验,如表 1 所示。

表 1 标准测试函数实验参数

函数名	函数	维数	搜索范围	理论最优值
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	10	[-100,100]	0
Rastrigrin	$f_2(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	10	[-5.12,5.12]	0
Rosenbrock	$f_3(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	10	[-30,30]	0
Ackley	$f_4(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	10	[-30,30]	0
Schaffer F6	$f_5(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} - 0.5$	2	[-100,100]	-1

4.2 实验参数设置

群智能优化算法的初始参数设置对算法的优化性能起着关键作用,因此为了验证算法的可比性,分别对 WRBFBA 算法、基本 BA 算法和基本 PSO 算法的初始参数进行一致性设置:

每种算法的种群数都设置为 $s = 40$,迭代次数为 300,趋化算子次数 $N_c = 40$,最大前进次数 $N_s = 5$,独立运行 20 次。对基本的 BA 算法,蝙蝠的频率范围 $f \in [0, 2]$,初始脉冲速率 $r^0 = 0.75$,脉冲速率增加系数 $\gamma = 0.9$,最大脉冲音强 $A = 0.7$,脉冲音强衰减系数 $\alpha = 0.9$ 。WRBFBA 算法基本参数同 BA 算法一致,其中趋化算子中步长 $c = 0.0005$,随机权重平均值的最大值 $\text{mean_max} = 0.9$,随机权重平均值的最小值 mean_

$\text{min} = 0.4$ 。在 PSO 算法中学习因子 $c_1 = c_2 = 2$,惯性权重 $\omega = 0.5$, $\sigma = 0.2$ 。

4.3 实验结果及分析

表 2 是分别使用算法 WRBFBA、PSO 和 BA 对 5 个测试函数进行实验得到的结果。可以看出,WRBFBA 算法在函数的最优值、最差值、平均值和标准差这四个指标中对四个高维复杂函数的寻优精度比 BA 算法和 PSO 算法高;对于 f_2, f_4 和 f_5 这样具有较多的局部最小值的函数,WRBFBA 算法搜索到的最优值比 BA 算法和 PSO 算法精度高并且 f_5 搜索到最优值 -1。对于 f_3 这样很难辨别搜索最优值方向和搜索到全局最优值的函数,算法 WRBFBA 的寻优结果比 BA 算法和 PSO 算法得到的寻优结果误差较小。

表 2 三种算法实验结果对比

函数	算法	最优值	最差值	平均值	标准差
f_1	PSO	0.192 5	1.001 6e+04	2.019 8e+03	4.095 9e+03
	BA	9.602 4e+03	1.768 7e+04	1.404 2e+04	2.561 0e+03
	WRBFBA	4.065 4e-07	1.857 5e-06	1.174 0e-06	4.057 5e-07
f_2	PSO	16.914 4	91.545 6	54.832 6	20.869 3
	BA	13.929 6	83.576 4	49.151 0	19.307 4
	WRBFBA	4.108 0e-05	4.975 4	0.249 0	1.112 5
f_3	PSO	71.170 1	7.994 7e+07	4.030 4e+06	1.786 9e+07
	BA	9.092 7e+06	4.592 7e+07	2.574 5e+07	1.073 1e+07
	WRBFBA	4.917 6	6.148 8	5.594 0	0.326 8
f_4	PSO	6.647 5	19.950 4	18.496 0	3.324 2
	BA	16.750 7	19.145 7	18.266 0	0.567 2
	WRBFBA	6.086 6e-04	0.001 7	0.001 4	3.017 8e-04
f_5	PSO	-1	-0.504 1	-0.969 4	0.109 6
	BA	-0.990 3	-0.570 2	-0.794 5	0.130 1
	WRBFBA	-1.000 0	-1.000 0	-1.000 0	8.860 7e-10

从图1~4每种算法的收敛情况可以看出, WRBFBA 算法快速向下收敛, 明显比 BA 算法和 PSO 算法收敛速度快; BA 和 PSO 算法收敛到一定值后基本保持不变的趋势; 由此可见改进的新型蝙蝠算法不仅避免了陷入局部最优解, 而且提高了算法的收敛速度和寻优精度。

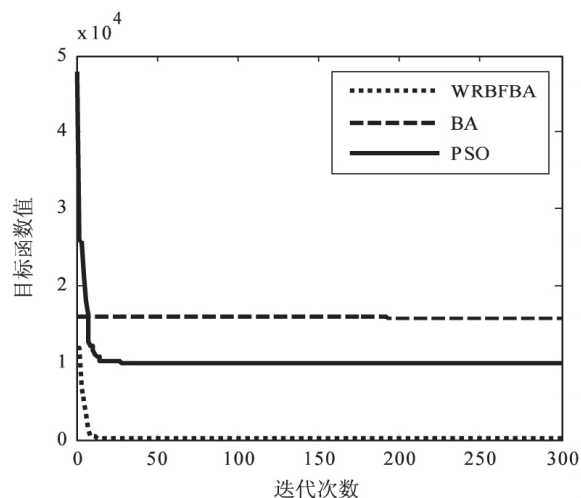


图1 三种算法对函数 f_1 的收敛曲线

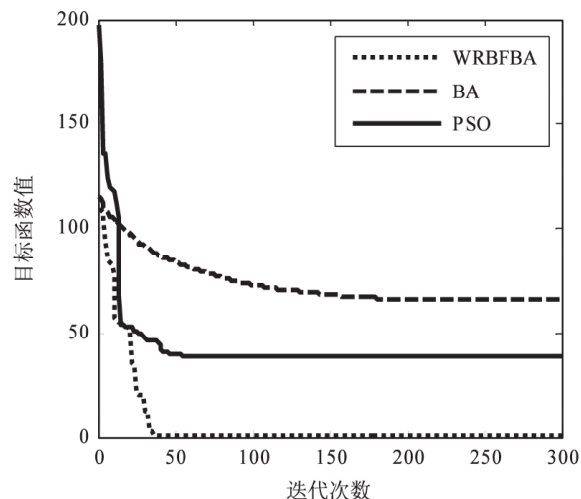


图2 三种算法对函数 f_2 的收敛曲线

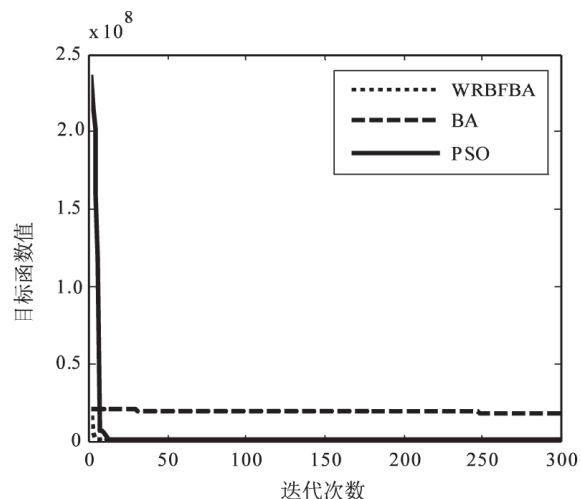


图3 三种算法对函数 f_3 的收敛曲线

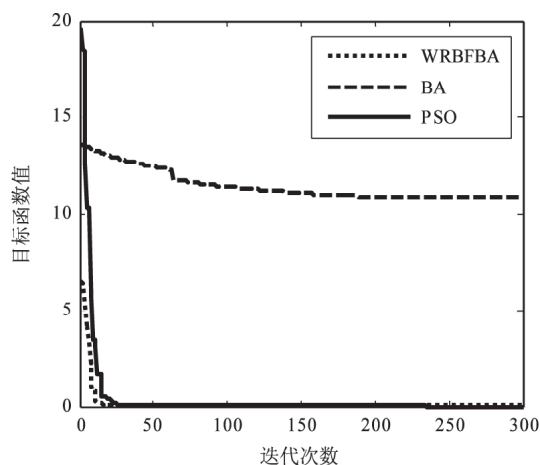


图4 三种算法对函数 f_4 的收敛曲线

5 结束语

针对基本蝙蝠算法存在的问题, 提出采用细菌觅食算法趋化算子改进蝙蝠种群的多样性和质量, 从而改进蝙蝠算法的局部搜索能力, 利用试探扰动因子来提高算法的求解精度和充分性, 采用随机惯性权重来均衡算法的探索能力和挖掘能力。实验结果表明, 提出的新型蝙蝠算法比基本的 BA 算法和 PSO 算法的寻优性能有明显的提高。下一步将对蝙蝠算法参数设置理论及实际的应用进行研究。

参考文献:

- [1] YANG Xinshe. A new metaheuristic bat-inspired algorithm [C]//Studies in computational intelligence. Berlin: Springer 2010: 65-74.
- [2] GANDOMI A H, YANG Xinshe, ALAVI A H, et al. Bat algorithm for constrained optimization tasks [J]. Neural Computing & Applications 2013 22(6): 1239-1255.
- [3] 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.
- [4] 肖辉辉, 段艳明. 基于 DE 算法改进的蝙蝠算法的研究及应用 [J]. 计算机仿真 2014 31(1): 272-277.
- [5] WANG Xiaowei, WANG Wen, WANG Yong. An adaptive bat algorithm [C]//9th international conference on intelligent computing theories and technology. Nanning, China [s. n.] 2013: 216-223.
- [6] 龚纯, 王正林. 精通 MATLAB 最优化计算 [M]. 北京: 电子工业出版社 2012: 163-179.
- [7] PASSINO K M. Biomimicry of bacterial foraging for distributed optimization and control [J]. IEEE Control Systems, 2002 22: 52-67.
- [8] 杨淑莹, 张桦. 群体智能与仿生计算: Matlab 技术实现 [M]. 北京: 电子工业出版社 2012: 196-200.
- [9] DAS S, BISWAS A, DASGUPTA S, et al. Bacterial foraging

(下转第 72 页)

资源时可以直接返回结果。

4 结束语

针对非结构化 P2P 网络中节点连接的随机性,提出了一种基于兴趣二叉搜索树的非结构化 P2P 覆盖网络拓扑结构,将内容相似度较大的节点构建成一棵二叉搜索树,利用二叉搜索树的特性,提高了搜索效率。通过 PeerSim 仿真,与传统的洪泛算法相比,该算法可以大大提高资源的搜索成功率,减少搜索的平均路径。

参考文献:

- [1] TEWARI S ,KLEINROCK L.Proportional replication in peer-to-peer networks [C]//International conference on computer communications. [s.l.]: IEEE, 2006: 1-12.
- [2] 王学龙,张璟.P2P 关键技术研究综述[J].计算机应用研究, 2010, 27(3): 801-805.
- [3] GAETA R ,GRANGETTO M.Identification of malicious nodes in peer-to-peer streaming: a belief propagation-based technique [J].IEEE Transactions on Parallel & Distributed Systems, 2013, 24(10): 1994-2003.
- [4] ZHAO B Y ,KUBIATOWICZ J D ,JOSEPH A D.Tapestry: an infrastructure for fault-tolerant wide-area location and routing[R].California: University of California at Berkeley, 2001.
- [5] 王挺,吴晓军,张玉梅.基于遗传算法的双向搜索 Chord 算法[J].计算机应用研究, 2016, 33(1): 46-49.
- [6] FURNESS J ,KOLBERG M. Considering complex search techniques in DHTs under churn [C]//Proceedings of the 2011 IEEE consumer communications and networking conference. Washington DC, USA: IEEE Computer Society, 2011: 559-564.
- [7] ADYA A ,BOLOSKEY W J ,CASTRO M ,et al.Farsite: federated, available, and reliable storage for an incompletely trusted environment [J].ACM SIGOPS Operating Systems Review, 2002, 36(SI): 1-14.
- [8] FERREIRA R A ,RAMANATHAN M K ,AWAN A ,et al. Search with probabilistic guarantees in unstructured peer-to-peer networks [C]//Proceedings of the fifth IEEE international conference on peer-to-peer computing. Washington DC, USA: IEEE Computer Society, 2005: 165-172.
- [9] ADAMIC L A ,LUKOSE R M ,PUNIYANI A R ,et al. Search in power-law networks [J].Physical Review E, 2001, 64(4): 046135.
- [10] RAMANATHAN M K ,KALOGERAKI V ,PRUYNE J. Finding good peers in peer-to-peer networks [C]//Proceedings of the 16th international symposium on parallel and distributed processing. Washington DC, USA: IEEE Computer Society, 2011: 24.
- [11] TAKEDA D ,SUGAWARA S.A content searching scheme using popularity and link degree of nodes in unstructured P2P networks with cache routers [C]//International conference on complex, intelligent, and software intensive systems. Washington DC, USA: IEEE Computer Society, 2016: 590-594.
- [12] 何可,吴晓军,张玉梅.基于节点兴趣的非结构化 P2P 网络拓扑结构研究[J].计算机工程与应用, 2016, 52(9): 102-107.
- [13] 谭义红,陈治平,林亚平.基于兴趣挖掘的非结构化 P2P 搜索机制研究与实现[J].计算机应用, 2006, 26(5): 1164-1166.
- [14] HSIAO H C ,SU H W.On optimizing overlay topologies for search in unstructured peer-to-peer networks [J].IEEE Transactions on Parallel and Distributed Systems, 2012, 23(5): 924-935.
- [15] BUCKLEY C.Implementation of the smart information retrieval system [R].Ithaca, NY, USA: Cornell University, 1985.
- [16] STIGLER S M.Francis Galton's account of the invention of correlation [J].Statistical Science, 1989, 4(2): 73-79.
- [17] 刘璇.非结构化 P2P 网络基于马尔科夫链的资源搜索算法研究[D].北京: 北京交通大学, 2015.
- [18] Multidisciplinary Optimization, 2008, 37(4): 395-413.
- [12] 伊进田,刘云连,刘丽,等.一种高效的混合蝙蝠算法[J].计算机工程与应用, 2014, 50(7): 62-66.
- [13] 王丽,王晓凯.一种非线性改变惯性权重的粒子群算法[J].计算机工程与应用, 2007, 43(4): 47-48.
- [14] 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.
- [15] LAAMARI M A ,KAMEL N.A hybrid bat based feature selection approach for intrusion detection [M]//Bio-inspired computing: theories and applications. Berlin: Springer, 2014: 230-238.

(上接第 67 页)

optimization algorithm: theoretical foundations, analysis, and applications [M]//Foundations of computational intelligence. [s.l.]: [s.n.], 2009: 23-55.

- [10] PAN T S ,KIEN D ,NGUYEN T T ,et al.Hybrid particle swarm optimization with bat algorithm [C]//Proceeding of the eighth international conference on genetic and evolutionary computing. Nanchang, China [s.n.], 2015: 37-47.
- [11] WANG Yong ,CAI Zixing ,ZHOU Yuren ,et al.Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique [J]. Structural and