Journal of Computer Applications
计算机应用,2020,40(6):1731 - 1737
ISSN 1001-9081
CODEN JYIIDU
2020-06-10
http://www.joca.cn

文章编号:1001-9081(2020)06-1731-07
DOI:10.11772/j.issn.1001-9081.2019101725

基于非均匀消除-扩散概率分布的情绪化细菌觅食算法

董　海[1],齐新娜[2*]
(1. 沈阳大学 应用技术学院,沈阳 110044; 2. 沈阳大学 机械工程学院,沈阳 110044)
(*通信作者电子邮箱1205344726@qq.com)

摘　要:针对传统的细菌觅食算法在优化过程中存在的趋化步长的不确定性及消除-扩散概率的恒定性不足的问题,提出一种基于非均匀消除-扩散概率的情绪化细菌觅食算法,以解决高维度工程优化问题。首先,在趋化步骤中利用古斯分布搜索机制对细菌个体位置进行更新,以解决细菌因以随机方式在每个维度上游动或翻转而导致的搜索能力差及易陷入局部最优的问题,引入情绪感知因子,利用情绪智能的突变来实现自适应趋化步长,从而避免算法过早收敛;其次,针对细菌个体在消除-扩散过程中概率的恒定性,提出利用线性和非线性概率分布代替传统的常数分布以此实现非均匀分布的构想,通过引入动力因子随机值,限制未定义的搜索空间中的细菌个体,从而节省算法的计算成本。通过六个基准测试函数进行测试,测试结果表明,在计算成本较低的情况下,除针对Rosenbrock函数外,所提算法针对所有函数均具有较低的迭代次数及良好的优化质量,且算法收敛性对比结果表明所提的算法具有较好的收敛性。

关键词:细菌觅食算法;情绪突变;古斯分布;动量因子;非均匀概率分布
中图分类号:TP301.6　　**文献标志码:**A

Emotional bacterial foraging algorithm based on non-uniform elimination-diffusion probability distribution

DONG Hai[1], QI Xinna[2*]
(1. *College of Applied Technology*, *Shenyang University*, *Shenyang Liaoning* 110044 *China*;
2. *College of Mechanical Engineering*, *Shenyang University*, *Shenyang Liaoning* 110044 *China*)

Abstract: In view of the uncertainty of chemotaxis step length and the lack of constancy of elimination diffusion probability in the optimization process of traditional bacterial foraging algorithm, in order to solve the problem of high-dimensional engineering optimization, an emotional bacteria foraging algorithm based on non-uniform elimination-diffusion probability was proposed. Firstly, in the chemotaxis step, the Gus distribution search mechanism was used to update the bacteria individual positions, so as to solve the problem of poor search ability and easy to fall into local optimum caused by bacteria swimming or flipping on each dimension in a random way. The emotion perception factor was introduced, and the sudden change of emotional intelligence was used to realize the adaptive chemotaxis step size, so as to avoid premature convergence of the algorithm. Secondly, in view of the probability constancy of bacterial individuals in the process of elimination-diffusion, the idea of using linear and non-linear probability distributions to replace the traditional constant distribution to realize non-uniform distribution was proposed. By introducing the random value of dynamic factor, the bacterial individuals in the undefined search space were limited, so as to save the calculation cost of the algorithm. Six benchmark functions were used in the test, and the test results show that: in the case of low calculation cost, except on Rosenbrock function, the proposed algorithm has low iteration times and good optimization quality on all functions, and the algorithm convergence comparison results show that the proposed algorithm has good convergence.

Key words: bacterial foraging algorithm; sudden change of emotion; Gus distribution; momentum factor; non-uniform probability distribution

0　引言

自20世纪40年代以来,针对工程存在的具有高维度、多峰值、非线性且不可微分等特征的实际问题,许多仿生优化演进算法被广泛用于优化多式联运、单一方式、多变量、多层面、多目标工程问题的基准功能[1]。初始阶段应用较广泛的搜索最优解的优化技术是Mirjalili[2]提出的遗传算法(Genetic Algorithm, GA),GA是通过模拟自然进化过程搜索最优解的方法,该算法被广泛应用于解决许多复杂的最优解和满意解等现实问题。Eberhart等[3]开发了粒子群优化(Particle Swarm Optimization, PSO)算法,该算法拥有更好的收敛性和更精确

收稿日期:2019-10-12;**修回日期:**2019-11-28;**录用日期:**2019-12-19。
基金项目:国家自然科学基金资助项目(71672117);国家社会科学基金资助项目(16BZX024)。
作者简介:董海(1971—),男,辽宁沈阳人,教授,博士,主要研究方向:网络化制造;　齐新娜(1994—),女,辽宁阜新人,硕士研究生,主要研究方向:车间调度。

的优化效果,但网络权重编码的选择较麻烦,计算成本高。刘晓阳等[4]提出了蚁群优化(Ant Colony Optimization, ACO)算法,该算法是用来在图中寻找优化路径的几率型算法,适用于图上路径搜索问题。李晓磊等[5]提出鱼群优化(Fish Swarm Optimization, FSA)多用于精度要求不高的情况,可快速得到一个可行解,具有局限性。继 Müller 等[6]提出的基于单个细菌运动行为的细菌趋药性(Bacterial Chemotaxis, BC)算法之后,Passion[7]在 2002 年提出细菌觅食优化算法(Bacterial Foraging Optimization Algorithm, BFOA),研究表明:尽管 BFOA 的搜索性能优于 GA,但不及 GA 成熟。

为了提高 BFOA 的性能,一些学者从算法融合和算法参数上进行了研究。在算法融合方面:刘小龙等[1]将分布估计算法的思想引入 BFOA 的繁殖过程,提出了基于高斯分布估计的细菌觅食算法;刘小龙等[8]将免疫算法与 BFOA 进行融合,提出了基于免疫算法的细菌觅食优化算法;王振东等[9]将混沌优化算法融入 BFOA,提出了混沌优化细菌觅食算法;徐玉韬等[10]将差分进化算法中的交叉与变异操作引入 BFOA,提出了一种混合型全局优化算法。在算法参数调整方面:易军等[11]提出了基于变邻域趋化操作的 BFOA,利用邻域搜索提高局部最优解的精确度;章国勇等[12]提出了一种具有量子行为的细菌觅食算法,以细菌群体信息为依据建立量子化的势能阱模型;谢平平等[13]提出了基于社会学习自适应 BFOA,将社会学习机制及自适应步长策略引入到标准 BFOA 中进行算法改进;蔡昌春等[14]提出了基于神经网络改进的细菌觅食算法;王新刚等[15]对趋化操作的运动步长以及翻转方向进行了改进,提出了一种改进型细菌觅食算法。

综上所述,上述研究提出的改进型 BFOA,其自适应趋化步长存在不适用于多模情况等局限性,且忽略了消除-扩散过程中概率的非均匀性[16]。针对上述问题,本文将基于情绪智能的突变引入标准的 BFOA,并结合符合非均匀概率分布的消除-扩散操作,提出了一种基于非均匀消除-扩散概率分布的情绪化细菌觅食算法,通过仿真及算法对比,验证了算法具有良好的优化性能。

1 算法设计

标准的细菌觅食优化算法(BFOA)通过趋化、群集、繁殖和消除-扩散四个步骤进行优化。在 BFOA 中,通过游动和翻转进行优化。在游动的过程中,如果细菌获取到最佳状态,会朝最佳状态食物功能评估,并改变其在最大食物方向上的位置。

首先,在任何优化技术中,最优解的精度和收敛性都取决于步长。由于 BFOA 的步长尺寸固定[17],存在以下缺点:

1)如果步长尺寸较小,可能无法获取到全局最优值;

2)如果步长尺寸较大,有可能在局部极小处捕获细菌,虽可快速获得最优解,但最优解的精度不高;

3)如果步长固定,细菌在接近最佳值时振荡,无法收敛于最佳值。

针对上述步长存在的不确定性,本文利用自适应或动态的步长来修正 BFOA,以获得低计算成本的最优解;与此同时,为了减少计算时间,提高经典 BFOA 中最优解的精度,本文拟从以下三个方面对 BFOA 进行改进。首先,对其位置进行更新,在趋化步骤中提出古斯分布搜索机制;其次,对其步长大小进行改进,在初始阶段,固定的步长用于局部搜索,当生成的数量增加时,利用自适应或动态步长进行全局搜索,并利用情绪智能的突变来实现自适应步长;第三,标准的 BFOA 在消除-扩散过程中,所有细菌个体都服从恒定的消除-扩散概率,这项操作会使即将接近最优位置的细菌替换为远离最优解的细菌,从而影响其收敛速度,本文利用线性和非线性概率分布代替传统的常数分布来实现非均匀分布。具体算法设计如下。

1.1 改进的趋化操作

传统的 BFOA 由于趋化操作,其搜索性能较差,发现每一种细菌都通过随机翻滚来调整自己的游动和翻转方向,使每一种细菌都能在特定的环境中生存。在整个生命周期内,以随机方式在每个维度上游动或翻转,从而使细菌具有较差的搜索能力,如果整个搜索空间很大,则容易陷入局部最优。此外,每个细菌的运动都是孤立的,在趋化操作中缺乏信息交换能力。为了解决细菌运动的基本趋向步骤,本文提出了基于古斯分布的搜索机制,对趋化性细菌的运动规律进行调整[17]。假设细菌 i 的位置由 $\theta^i(j,k,l)$ 表示,其中:j 为趋化步骤,k 为繁殖步骤,l 为消除-扩散步骤。$C(i)$ 表示每次游动或翻转的趋化步长大小,那么对于每次趋化步骤,细菌 i 的运动可表示为:

若 $\theta^i_{\mathrm{pbest}}(j,k,l) \neq \theta^i_{\mathrm{gbest}}(j,k,l)$,则:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \times N\left[\mu^i(j,k,l), \sigma^i(j,k,l)\right] \tag{1}$$

否则:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \times N\left[\mu'(j,k,l), \sigma'(j,k,l)\right] \tag{2}$$

其中:

$$\mu^i(j,k,l) = \frac{\theta_{\mathrm{pbest}}(j,k,l)/i - \theta_{\mathrm{gbest}}(j,k,l)}{2} - \theta^i(j,k,l) \tag{3}$$

$$\sigma^i(j,k,l) = \left|\theta^i_{\mathrm{pbest}}(j,k,l) - \theta(j,k,l)\right| \tag{4}$$

$$\mu'(j,k,l) = \frac{\sum_{i=1}^{p} \frac{\theta_{\mathrm{pbest}(j,k,l)}}{i} - \theta_{\mathrm{gbest}}(j,k,l)}{2} - \theta^i(j,k,l) \tag{5}$$

$$\sigma'(j,k,l) = \left|\sum_{i=1}^{p} \frac{\theta_{\mathrm{pbest}}(j,k,l)}{i} - \theta_{\mathrm{gbest}}(j,k,l)\right| \tag{6}$$

$$C(i) = \frac{1}{N_s + 1} \tag{7}$$

式中:N 表示游动或翻转的次数;p 表示局部细菌个体总数。

1.2 群集操作

接近最优路径的细菌个体可跟随其他较低梯度的细菌个体进行觅食,以便更快地收敛。菌群的群集行为使细菌个体形成群体,并以同心圆为轨迹进行运动,菌群密度高,则食物梯度高。该群集机制可建模型,如式(8)表示。

$$J_{cc}(\theta, P(j,k,l)) = \sum_{i=1}^{s} J^i_{cc}\left(\theta, \theta^i(j,k,l)\right) =$$

$$\sum_{i=1}^{n}\left[-d_{\text{attract}}\exp\left(-w_{\text{attract}}\sum_{m=1}^{q}\left(\theta_m-\theta_m^i\right)^2\right)\right]+$$
$$\sum_{i=n}^{s}\left[-h_{\text{repelent}}\exp\left(-w_{\text{repelent}}\sum_{m=1}^{q}\left(\theta_m-\theta_m^i\right)^2\right)\right] \tag{8}$$

其中:$J_{cc}(\theta,P(j,k,l))$表示细菌个体之间的吸引和排斥作用值;$\theta\in\{\theta_1,\theta_2,\cdots,\theta_q\}$表示$q$维搜索域中的一个点;$s$表示细菌的总数;$q$表示优化变量的个数;$n$表示维度内某节点;$d_{\text{attract}}$、$w_{\text{attract}}$分别表示细菌的引诱因子深度和宽度的度量系数;$h_{\text{repelent}}$、$w_{\text{repelent}}$分别表示细菌驱避高度和宽度系数。

1.3　情绪化突变

基于情绪智能突变使细菌的位置发生动态变化,在基于非均匀消除-扩散概率分布的情绪化细菌觅食算法中,突变发生在繁殖步骤之前。在突变中,采用自适应步长可避免过早收敛,此过程假设细菌有心理,每种细菌都有两种情绪:快乐和悲伤,每种细菌都会根据自己的感受对当前的位置作出反应,不同的位置被用来刺激下一代的细菌。利用著名的韦伯-费克纳定律计算情绪感知因子。由韦伯-费克纳定律描述的刺激强度P大小之间的关系[16],如式(9)所示:

$$P=k\ln(S/S_0) \tag{9}$$

其中:k为常数因子;S_0为刺激阈值;S为刺激函数。刺激和感知之间使用对数关系。如果一个刺激依据几何表达式变化,并且相应的知觉在对数关系的数学表达式中改变。细菌的全局感知r_g和历史感知r_h分别通过方程式(10)~(11)来描述:

$$r_g=-k\frac{S[f(\theta_{\text{global}})-f(\theta(i,j,k,l))]}{S_0} \tag{10}$$

$$r_h=-k\ln\frac{S[f(\theta_{\text{pbest}})-f(\theta(i,j,k,l))]}{S_0} \tag{11}$$

其中,$f(\theta(i,j,k,l))$表示如果细菌远离全局最佳位置,将对刺激产生强烈的反应,这些刺激将与其所经历的历史相比较。细菌的快乐和悲伤两种情绪会动态地改变细菌的位置,使其步幅变小或变大,细菌的情绪表现如方程式(12)~(13)所示。

快乐的细菌个体为:

$$v(i,j,k,l)=w(k)\times v(i,j,k,l)+$$
$$c\times r_g\times r_h\times rand\times[f(\theta_{\text{global}})-f(\theta(i,j,k,l))] \tag{12}$$

悲伤的细菌个体为:

$$v(i,j,k,l)=w(k)\times v(i,j,k,l)+c\times\frac{r_h}{r_g}r_h\times$$
$$rand\times[f(\theta_{\text{global}})-f(\theta(i,j,k,l))] \tag{13}$$

其中:θ_{global}表示全局最佳位置;$f(\theta_{\text{global}})$表示全局最佳位置适应度;$f(\theta(i,j,k,l))$表示实时的适应度;$v(i,j,k,l)$表示细菌速度;$w(k)$表示线性减小惯性重量;$c$表示加速度系数;$rand$表示随机函数。

根据情绪感知因子更新细菌速度:如果随机函数$rand$小于情绪感知因子e_s,依据式(12)更新细菌速度;否则,依据等式(13)更新细菌速度。

根据式(14)可推断出全局最佳位置$\boldsymbol{v}(i,j+1,k,l)$,以更新每种细菌的当前位置:

$$\boldsymbol{v}(i,j+1,k,l)=\theta_{\text{global}}\times m+v(i,j,k,l) \tag{14}$$

其中,m表示动力因子。

为了防止细菌在游泳或翻转后离开脱水区域,使用动量因子。动量因子限制了未定义的搜索空间中的细菌,因此不必在每一代之后检查细菌的位置,从而节省计算成本。动量因子的选择为随机值而不是常量值,因为所有细菌每次都会被拉入具有不同限制的搜索空间,从而导致无法逃离全局最优值。在整个搜索空间,搜索实验的初始阶段步长非常大,当细菌接近全局最优值时,步长会动态地缩减,因此,自适应步长$f(\theta(i,j,k,l))$将更接近$f(\theta_{\text{global}})$。

1.4　繁殖操作

在基于非均匀消除-扩散概率的分布的细菌觅食算法中,细菌的繁殖类似于经典的BFOA,随着营养素的不断吸收,细菌逐渐生长。在适当的条件下,每个健康的个体会分裂成两个子细菌,但是,对于营养不良的个体这些细菌会被清除。为了保留细菌趋化过程中的良好经验,加快算法的收敛速度,本文将细菌适应度在历次趋化过程中的历史最优值作为健康适应度函数J_{health}[13],再现操作排序如式(15)所示:

$$J_{\text{health}}=\sum_{j=1}^{N_c}J(i,j,k,l) \tag{15}$$

其中,N_c为趋向步骤次数。

1.5　非均匀概率分布的消除-扩散操作

为了避免在局部最优条件下存储细菌的可能性,提高细菌的搜索能力,一些细菌被消除,而一些则根据概率分散在搜索空间中内,即在该过程中,具有最佳适应度值和最差适应度值的细菌均以相同概率被消除。因此,会将接近最优位置的细菌个体替换为远离最优位置的细菌个体,从而影响算法的性能[16]。据此,本文提出依据非均匀概率分布消除细菌,该方法是通过将细菌重新定位到环境中的随机位置来实现该过程的。线性消除概率定义为:

$$P_e(i)=\frac{i-1}{s/2-1};\quad i=1,2,\cdots,\frac{s}{2} \tag{16}$$

根据式(16)可知,种群内的消除-扩散概率与种群内的细菌指数成线性正比,其概率步长增加为$s/2$,在该情况下,获得最佳适应度的细菌被消灭的可能性最低,适应度最差的个体因被分配了等同的概率而被淘汰。另一种非均匀分布概率可用于定义非线性行为,如式(17)所示。

$$P_e{}^*(i)=\left[\frac{i-1}{s/2-1}\right]^2;\quad i=1,2,\cdots,\frac{s}{2} \tag{17}$$

在该概率分布中,最优细菌个体将具有最低的概率,而最差的细菌个体将完全被淘汰,在这种情况下概率与细菌指数成线性正比。

消除-扩散概率(P_e)决定了搜索过程中的探索和开发的量:P_e值越大,消除作用越强,对细菌的迁移作用越好,可避免陷入局部极小值,并且探索过程越容易实现;P_e值越小,越可保证最佳细菌的发展,并可加强对局部区域的研究,然而这种细菌将有更少的机会被从局部区域释放,可能会造成局部极小值。因此,控制P_e值至关重要。在研究初期本文采取探索整个搜索空间,相反,在搜索结束时进行开发(增强),该操作可通过分配小的P_e值来实现。对于线性分布及非线性分布可

推广为包含消除–扩散指数,分别定义如式(18)、(19)所示:

线性为:

$$P_e(i,l)=P_e(i)+\left[i-\left(\frac{s}{4}+0.5\right)\right]\frac{l-1}{(N_{ed}-1)(s/2-1)} \tag{18}$$

非线性为:

$$P_e^*(i,l)=P_e^*(i)\left[\frac{l-1}{N_{ed}-1}\right]\left[2\left(\frac{i-1}{s/2-1}\right)^2-1\right]+P_e^*(i) \tag{19}$$

1.6 算法设计

本文提出的基于非均匀消除–扩散概率的情绪化细菌觅食算法流程如图1所示。

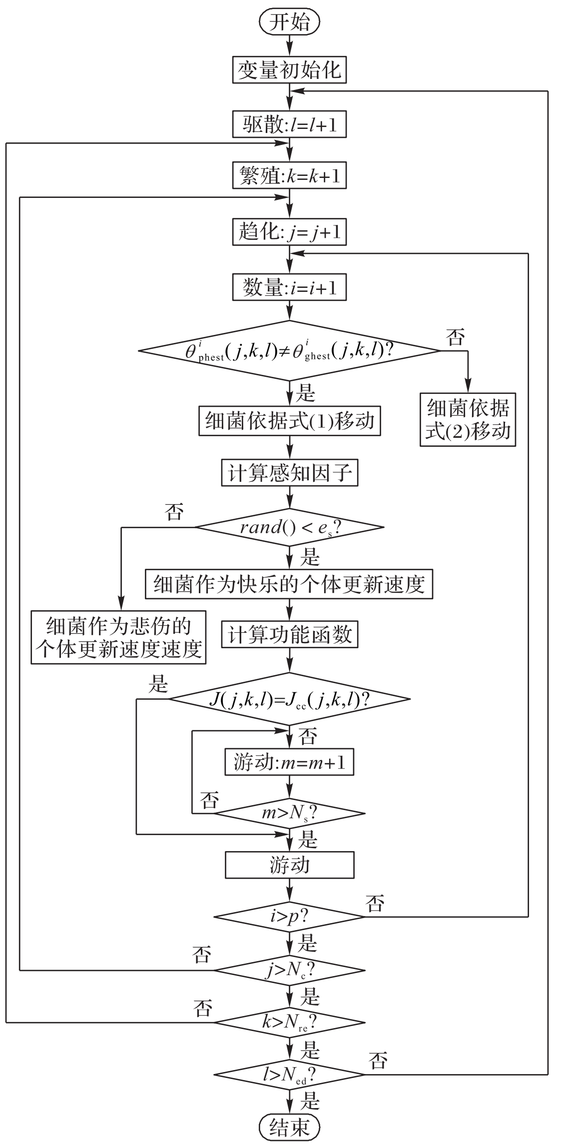

图1 本文算法流程

Fig. 1 Flowchart of proposed algorithm

2 仿真实验与结果分析

2.1 仿真设计

表1给出Rosenbrock、Sphere、Griewank、Schaffer's、Shekel's Foxholes、Rastrigin六个基准函数,测试本文提出的基于非均匀消除–扩散概率分布的情绪化细菌觅食算法,并验证算法的性能。其中:$f_1(x)$与$f_2(x)$为单峰函数,解的空间有非常多狭窄的局部极小通道;$f_3(x)$、$f_4(x)$、$f_5(x)$与$f_6(x)$为多峰函数,算法很容易陷入通向全局最优点的局部最优上。表2为全局最优搜索范围及初始化范围的参数,表中d表示无限大。表3为算法参数设置。

表1 模拟基准函数

Tab. 1 Simulated benchmark functions

基准函数名称	数学表达式	函数类型
Rosenbrock	$f_1(x)=\sum_{i=1}^{d}\left(100\left(x_{i+1}-x_i^2\right)^2\right)$	单峰
Sphere	$f_2(x)=\sum_{i=1}^{d}x_i^2$	单峰
Griewank	$f_3(x)=\frac{1}{4\,000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{2}\cos\frac{x_i}{\sqrt{i}}+1$	多峰
Schaffer's	$f_4(x)=0.5+\frac{\left(\sin\sqrt{x^2+y^2}\right)^2-0.5}{\left(1.0+0.001\left(x^2+y^2\right)\right)^2}$	多峰
Shekel's Foxholes	$f_5(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right]^{-1}$	多峰
Rastrigin	$f_6(x)=\sum_{i=1}^{d}\left(x_i^2-10\cos 2\pi x_i+10\right)$	多峰

表2 搜索范围及初始化范围

Tab. 2 Search range and initialization range

函数	全局最优	搜索范围	初始化范围
$f_1(x)$	0.000	$[-100]^d$~$[100]^d$	$[16]^d$~$[32]^d$
$f_2(x)$	0.000	$[-100]^d$~$[100]^d$	$[50]^d$~$[100]^d$
$f_3(x)$	0.000	$[-500]^d$~$[500]^d$	$[250]^d$~$[500]^d$
$f_4(x)$	0.000	$[-100]^d$~$[100]^d$	$[16]^d$~$[32]^d$
$f_5(x)$	0.996	$[-65.536]^2$~$[65.536]^2$	$[0]^2$~$[65.536]^2$
$f_6(x)$	0.000	$[-10]^d$~$[10]^d$	$[2.56]^d$~$[5.12]^d$

2.2 算法对比分析

为测试新算法的性能,本文对具有量子行为的细菌觅食算法(Bacterial Foraging Algorithm with Quantum Behavior, BFAQB)、差分进化改善的细菌觅食算法(Improved Bacterial Foraging Algorithm Based on Differential Evolution, IBFABDE)、基于高斯分布的细菌觅食算法(Bacterial Foraging Algorithm Based on Gaussian Distribution, BFABGD)及本文提出的基于非均匀消除–扩散概率分布的情绪化细菌觅食算法(Non-uniform Probability Emotional Bacteria Foraging Algorithm, NPEBFA)在上述六个基准函数进行测试,实验对比结果如表4所示。

表4中,NPEBFA的性能优于BFAQB和IBFABDE,但对Rosenbrock函数的性能不如BFABGD。与NPEBFA相比,BFABGD获得的Rosenbrock函数的平均迭代效率优于NPEBFA。

Sphere函数的模拟结果显示,NPEBFA和比较算法执行

于球体函数上,标准差分析结果表明,球面函数拟合度的参数　值彼此接近。

表3　算法参数设置

Tab. 3　Algorithm parameter setting

算法	趋化步骤次数 N_c	游动长度 N_s	繁殖次数 N_{re}	消除-扩散事件次数 N_{ed}	消除-扩散事件概率 P_{ed}	趋化步长 $C(i)$
BFAQB	50	125	2	4	0. 25	0. 05
IBFABDE	50	125	2	4	0. 25	0. 10
BFABGD	50	125	2	4	0. 25	0. 15
NPEBFA	50	125	2	4	0. 25	0. 20

表4　不同算法函数仿真结果对比

Tab. 3　Comparison of functional simulation results of different algorithms

函数	指标	BFAQB	IBFABDE	BFABGD	NPEBFA
$f_1(x)$	平均值	5. 678 33E+1	5. 363 22E+1	2. 267 16E+1	2. 943 82E+1
	最大值	1. 576 10E+2	1. 570 03E+2	2. 373 41E+1	8. 098 31E+1
	最小值	3. 987 835	3. 987 033	2. 091 77E+1	2. 869 67E+1
	中位数	6. 537 462E+1	6. 085 728E+1	2. 281 630E+1	3. 145 870E+1
	标准差	3. 822 346 1E+1	3. 584 537 2E+1	0. 945 7	4. 350 0
	平均迭代次数	2 000. 00	2 000. 00	475. 26	1 000. 00
$f_2(x)$	平均值	0. 597 46E-4	0. 595 14E-4	0. 295 18E-4	0. 264 87E-4
	最大值	0. 975 651E-4	0. 974 320E-4	0. 481 523E-4	0. 306 134E-4
	最小值	0. 149 71E-4	0. 679 00E-5	0. 121 99E-4	0. 139 70E-6
	中位数	0. 733 32E-4	0. 667 00E-4	0. 324 30E-4	0. 278 95E-4
	标准差	0. 298 376E-4	0. 278 803E-4	0. 112 231E-4	0. 839 900E-5
	平均迭代次数	306. 20	106. 72	15. 19	9. 14
$f_3(x)$	平均值	0. 352 573 0E-1	0. 142 989 1E-1	0. 831 540 0E-4	0. 737 000 0E-5
	最大值	0. 173 849 487	0. 833 066 9E-1	0. 985 470 0E-4	0. 124 730 0E-4
	最小值	0. 109 55E-3	0. 746 29E-4	0. 537 22E-4	0. 131 00E-5
	中位数	0. 181 359 8E-1	0. 985 728 7E-2	0. 870 460 0E-4	0. 589 700 0E-4
	标准差	0. 475 030 5E-1	0. 185 053 3E-1	0. 143 870 0E-4	0. 231 700 0E-4
	平均迭代次数	1 383. 87	922. 71	109. 08	16. 02
$f_4(x)$	平均值	0. 267 92E-2	0. 220 41E-2	0. 440 30E-3	0. 532 10E-4
	最大值	0. 971 57E-2	0. 971 48E-2	0. 972 08E-2	0. 189 95E-2
	最小值	0. 230E-4	0. 170E-4	0. 390E-5	0. 390E-5
	中位数	0. 793E-4	0. 763E-4	0. 523E-4	0. 286E-4
	标准差	0. 424 41E-2	0. 387 31E-2	0. 193 22E-2	0. 284 90E-4
	平均迭代次数	321. 41	249. 30	44. 73	36. 77
$f_5(x)$	平均值	9 980. 386 3E-4	9 980. 332 7E-4	980. 265 0E-4	9 980. 359 9E-4
	最大值	9 980. 950 1E-4	9 980. 890 9E-4	9 980. 948 1E-4	9 980. 639 9E-4
	最小值	9 980. 038 0E-4	9 980. 038 0E-4	9 980. 038 0E-4	9 980. 038 0E-4
	中位数	9 980. 324 7E-4	9 980. 184 0E-4	9 980. 145 2E-4	9 980. 234 4E-4
	标准差	0. 326 9E-4	0. 310 8E-4	0. 265 2E-4	0. 000 302 4
	平均迭代次数	90. 07	43. 71	40. 10	22. 12
$f_6(x)$	平均值	7. 730 617E+1	5. 149 903E+1	0. 821 000E-4	0. 572 000E-4
	最大值	1. 457 385 9E+2	7. 959 657 1E+1	0. 997 000 0E-4	0. 638 000 0E-4
	最小值	3. 310 945 0E+1	3. 239 033 0E+1	0. 522 000 0E-4	0. 490 000 0E-5
	中位数	7. 169 040 5E+1	4. 875 794 0E+1	0. 836 000 0E-4	0. 590 000 0E-4
	标准差	2. 805 810 1E+1	1. 155 063 4E+1	0. 133 000 0E-4	0. 102 000 0E-4
	平均迭代次数	2 000. 00	1 973. 00	251. 15	55. 12

Griewank 函数的模拟结果显示,该方法可以提高算法的精度和收敛性,通过分析 NPEBFA 计算结果,证明其能够以较低的计算成本获得精确的最优解。

Schaffer's 函数的模拟结果显示,NPEBFA 以较低的计算成本提供了较好的优化质量。

Shekel's Foxholes 函数的模拟结果显示,其全局最小值不是零。对于这种函数,NPEBFA 技术在节省计算成本方面的性能也很好,同时与其他方法相比,拟合参数彼此接近。

Rastrigin 函数的模拟结果显示,NPEBFA 以最小的计算成本获得精确的最优值。与 BFAQB、IBFABDE 和 BFABGD 相比,NPEBFA 对 Rastrigin 函数的代数要求较少。

对单个最优函数和多个局部最优函数的实验结果表明,该算法除针对 Rosenbrock 函数外,在计算成本较低的情况下,具有较好的优化质量,且对于求解多模态函数等复杂问题更

为有效，优化精度高。

不同算法收敛性的比较如图2所示，仿真结果表明本文提出的NPEBFA在最优均值范围不同的情况下均具有较好的收敛性，即该算法较BFABGD、IBFABDE、BFAQB更适用于不同的搜索空间范围，该优势可确保算法优化过程中种群的多样性，并避免范围内待优化因子的丢失。

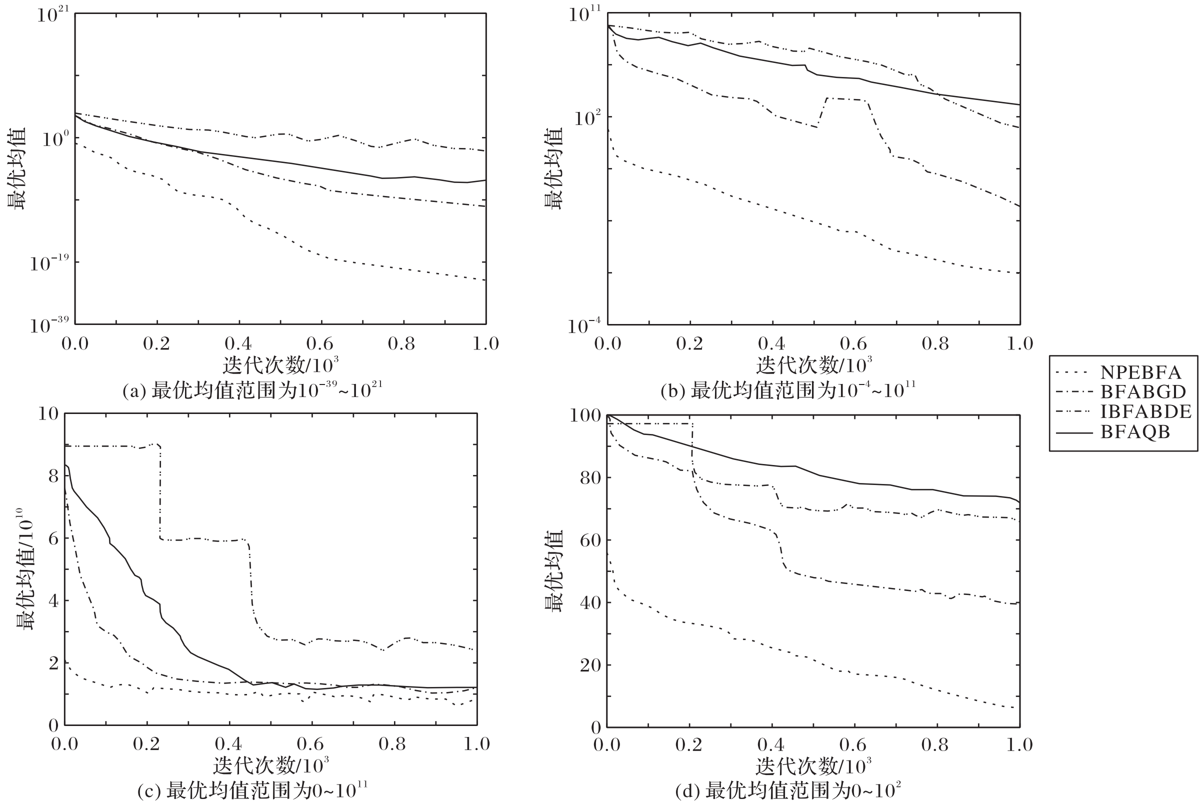

图2 不同算法收敛性比较

Fig. 2 Convergence comparison of different algorithms

3 结语

本文通过赋予细菌个体情绪感知因子的概念来改变细菌的趋化步长，并将古斯分布搜索机制嵌入趋化步骤中，同时在个体消除-扩散过程中，利用线性和非线性概率分布代替传统的常数分布来实现非均匀分布，进而提出了一种改进的细菌觅食优化算法，算法中的待优化变量的搜索空间在开发能力和探索能力上取得提升。仿真实验结果表明，本文提出的NPEBFA的性能及收敛性优于传统的BFO算法，并且避免了其他改进细菌觅食算法存在的部分缺陷，适用于解决高维度工程优化问题。

参考文献(References)

[1] 刘小龙，李荣钧，杨萍．基于高斯分布估计的细菌觅食优化算法[J]．控制与决策，2011，26(8)：1233-1238.（LIU X L, LI R J, YANG P. Bacterial foraging optimization algorithm based on estimation of distribution[J]. Control and Decision, 2011, 26(8): 1233-1238.）

[2] MIRJALILI S. Evolutionary Algorithms and Neural Networks [M]. Cham: Springer, 2019: 1-11.

[3] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory [C]// Proceedings of the 6th International Symposium on Micro Machine and Human Science. Piscataway: IEEE, 1995: 39-43.

[4] 刘晓阳，刘恩福，靳江燕．基于蚁群算法的异步并行装配序列规划方法[J]．机械工程学报，2019，55(9)：107-119.（LIU X Y, LIU E F, JIN J Y. Asynchronous parallel assembly sequence planning based on ant colony algorithm[J]. Journal of Mechanical Engineering, 2019, 55(9): 107-119.）

[5] 李晓磊，邵之江，钱积新．一种基于动物自治体的寻优模式：鱼群算法[J]．系统工程理论与实践，2002，22(11)：32-38.（LI X L, SHAO Z J, QIAN J X. An optimal model based on animal autonomous body: fish swarm algorithm[J]. System Engineering Theory and Practice, 2002, 22(11): 32-38.）

[6] MÜLLER S D, MARCHERLO J, AIRAGHI S, et al. Optimization algorithms based on a model of bacterial chemotaxis [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 16-29.

[7] PASSINO K M. Biomimicry of bacterial foraging for distributed optimization and control [J]. IEEE Control Systems Magazine, 2002, 22(3): 52-67.

[8] 刘小龙，赵奎领．基于免疫算法的细菌觅食优化算法[J]．计算机应用，2012，32(3)：634-637，653.（LIU X L, ZHAO K L. Bacterial foraging optimization algorithm based on immune algorithm [J]. Journal of Computer Applications, 2012, 32(3): 634-637, 653.）

[9] 王振东，陈峨霖，胡中栋．混沌优化细菌觅食算法的传感器节点部署策略[J]．传感技术学报，2018，31(1)：110-118.（WANG Z D, CHEN E L, HU Z D. Sensor node deployment strategy of chaotic optimization of bacterial foraging algorithm [J]. Chinese Journal of Sensors and Actuators, 2018, 31(1): 110-118.）

[10] 徐玉韬,谈竹奎,吕黔苏,等. 基于差分进化的改进细菌觅食算法在智能配电网故障定位方法中的应用[J]. 电力大数据,2018,21(5):1-7.(XU Y T, TAN Z K, LYU Q S, et al. Application of improved bacteria feeding algorithm based on differential evolution in fault location of active distribution network [J]. Power Systems and Big Data, 2018, 21(5): 1-7.)

[11] 易军,李太福. 求解作业车间调度的变邻域细菌觅食优化算法[J]. 机械工程学报,2012,48(12):178-183.(YI J, LI T F. Bacterial foraging optimization algorithm based on variable neighborhood for job-shop scheduling problem [J]. Journal of Mechanical Engineering, 2012, 48(12): 178-183.)

[12] 章国勇,伍永刚,谭宇翔. 一种具有量子行为的细菌觅食优化算法[J]. 电子与信息学报,2013,35(3):614-621.(ZHANG G Y, WU Y G, TAN Y X. Bacterial foraging optimization algorithm with quantum behavior [J]. Journal of Electronic and Information Science, 2013, 35(3): 614-621.)

[13] 谢平平,李银红,刘晓娟,等. 基于社会学习自适应细菌觅食算法的互联电网AGC最优PI/PID控制器设计[J]. 中国电机工程学报,2016,36(20):5440-5448.(XIE P P, LI Y H, LIU X J, et al. Optimal PI/PID controller design of AGC based on social learning adaptive bacteria foraging algorithm for interconnected power grids [J]. Proceedings of the CSEE, 2016, 36(20): 5440-5448.)

[14] 蔡昌春,程述成,邓志祥,等. 基于模糊RBF神经网络的直流微电网并网等效建模[J]. 电网技术,2016,40(11):3446-3452.(CAI C C, CHENG S C, DENG Z X, et al. DC microgrid equivalent modeling based on fuzzy-RBF artificial neural network [J]. Power System Technology, 2016, 40(11): 3446-3452.)

[15] 王新刚,衣鹏飞. 改进型细菌觅食算法求解FJSP问题[J]. 计算机与数字工程,2018,46(6):1094-1098.(WANG X G, YI P F. Improved bacteria foraging optimization algorithm for solving flexible job-shop scheduling problem [J]. Computer and Digital Engineering, 2018, 46(6): 1094-1098.)

[16] SAHIB M A, ABDULNABI A R, MOHAMMED M A. Improving Bacterial foraging algorithm using non-uniform elimination-dispersal probability distribution [J]. Alexandria Engineering Journal, 2018, 57(4): 3341-3349.

[17] MUSAL U, BAKARE G A, MATI A A, et al. Radial distribution network enhancement with D-statcom using bacterial foraging algorithm [J]. Arid Zone Journal of Engineering, Technology and Environment, 2018, 14(2): 249-260.

This work is partially supported by the National Natural Science Foundation of China (71672117), the National Social Science Foundation of China (16BZX024).

DONG Hai, born in 1971, Ph. D., professor. His research interests include networked manufacturing.

QI Xinna, born in 1994, M. S. candidate. Her research interests include job shop scheduling.