

基于免疫算法的细菌觅食优化算法

刘小龙*, 赵奎领

(华南理工大学 工商管理学院, 广州 510640)

(*通信作者电子邮箱 810963@qq.com)

摘要:针对细菌觅食优化算法经常出现的速度较慢、步长一致的缺陷,赋予细菌灵敏度的概念,对细菌游动的步长进行调节以提高收敛速度。采用免疫算法中的克隆选择思想,对精英细菌群体进行克隆、高频变异和随机交叉,引导算法提高搜索精度。典型高维函数测试表明,改进算法的搜索速度和精度得到极大提升,算法改造后可适用于多维、约束等实际工程问题中的优化。

关键词:灵敏度;免疫算法;细菌觅食;全局优化

中图分类号: TP183; TP301.6 **文献标志码:** A

Bacteria foraging optimization algorithm based on immune algorithm

LIU Xiao-long*, ZHAO Kui-ling

(School of Business Administration, South China University of Technology, Guangzhou Guangdong 510640, China)

Abstract: To correct the defects such as slower speed, step consistence in bacteria foraging optimization algorithm, this paper presented the concept of the sensitivity of bacteria to increase convergence speed by adjusting the step size of bacterial swimming. The clonal selection ideas in immune algorithm were used to achieve bacterial cloning, high-frequency variation and random crossover of the elite group, and to guide the search algorithm to improve accuracy. A number of typical high-dimensional function tests show that the improved algorithm has been greatly improved in terms of search speed and accuracy, and is more appropriate to solve practical engineering optimization problems such as high dimensionality, constraints.

Key words: sensitivity; immune algorithm; bacterial foraging; global optimization

0 引言

自20世纪40年代开始,越来越多的工程实际问题呈现出多峰值、高维度和非线性且不可微分的特征,为解决这些问题,许多学者致力于从生物系统来寻找灵感,并设计和构造了大批的仿生优化算法。进入21世纪后,仿生优化算法又有了新的发展,比较有代表性的如 Müller 等^[1]提出的细菌趋药性(Bacterial Chemotaxis, BC)算法和 Passino^[2]提出的标准细菌觅食优化(Bacterial Foraging Optimization, BFO)算法。上述两种算法的灵感都源于微生物的行为特性,其中BC是源于单个细菌的趋药性行为,而BFO则来源于细菌的群体行为特性。BFO作为一种基于群体的搜索技术,通过群体细菌之间的竞争和协作,实现搜索的优化。此外,有关文献表明,BFO具有避免局部最优的可能,并逐渐成为仿生优化的一个研究热点。在BFO的研究中,相关学者主要从算法的参数调整和算法融合上进行研究,以提高其性能。

在参数调整方面,Liu等^[3]改进了大肠杆菌间的相互作用机制,并对BFO算法的收敛性进行了初步分析;Mishra^[4]提出了基于TS的模糊规则系统改进的MBFO算法;Majhi等^[5]将设计出的自动趋化步长BFO模型应用到了神经网络训练当中;Datta等^[6]设计出了一种根据自适应的增量调制原理实现自适应趋化步长的BFO算法;Chen等^[7]提出了基于生物的自适应搜索策略的自适应协同型的菌群觅食优化算法。在现行有关细菌觅食优化算法的研究中,Majhi、Datta和Chen等从趋化步长的角度,根据细菌在觅食生命周期内的获取能量

大小,使细菌能够自适应地调节趋化步长,在不增加算法复杂性的情况下,提高细菌寻优效率。

在算法的融合设计方面,Biswas等^[8]将粒子群算法与BFO算法相结合,提出了一种应用于多峰函数优化的混合优化算法(以下简称PSOBFO算法);Bakwad等^[9]、Tang等^[10]和Chu等^[11]将粒子群优化(Particle Swarm Optimization, PSO)算法的基本思想引入到BFO算法中,分别提出了细菌群和快速细菌群算法等。Biswas、Dasgupta等^[12]将差分进化算法中的交叉与变异操作引入BFO算法,提出了一种混合型全局优化算法;Kim等^[13]将遗传算法中的交叉、变异算子引入到BFO算法中,构造出GABFO算法;Luh等^[14]则从细菌进化的视角,设计出了细菌进化算法(Bacterial Evolutionary Algorithm, BEA)。

Biswas等在细菌的趋化算子当中嵌入了粒子群的方法,同时使细菌具有感知全局极值的能力,通过粒子群算法的迭代公式对细菌的位置进行更新,在这个过程中,趋化算子并未发挥作用,且该方法嵌入粒子群位置更新公式增加了时间复杂性。Dasgupta等在细菌进化中引入了交叉、变异算子,虽然使种群的多样性得到增加,但这样也极易导致优秀个体缺失情况的发生。

鉴于以上情况,本文通过对细菌赋予灵敏度的概念,使细菌在趋化的过程中能够依据能量自适应地改变步长,从而实现对环境适应。继而在趋化周期完成后,采用免疫算法中的克隆选择算法思想,对挑选的精英细菌进行克隆繁殖,并经过克隆群体的高频变异和随机交叉,取代细菌群体中较差的

收稿日期:2011-09-15;修回日期:2011-12-18。

基金项目:国家自然科学基金资助项目(71071057);中央高校基本科研业务费专项资金资助项目(2012ZM0031)。

作者简介:刘小龙(1977-),男,湖南永州人,讲师,博士,主要研究方向:智能优化算法、管理决策;赵奎领(1986-),男,河南禹州人,硕士研究生,主要研究方向:生产系统优化。

低能量或者高适应度的个体。相关实验表明,免疫算法有效地提高了细菌群体的搜索精度和适应度。

1 基于免疫算法的改进细菌觅食优化算法

1.1 细菌觅食优化算法

细菌觅食优化算法通过模拟大肠杆菌的趋药性行为,即根据趋化、繁殖和迁移这3个算子的迭代进行优化,该算法的本身并不涉及到群体细菌相互之间的影响和相互作用的机制,主要用于解决局部优化问题。在BFO算法中,用趋化算子来模拟细菌随机地向营养丰富区域集中的过程,具体的行为主要有翻转和游动两种。在趋化过程中,经过固定周期的搜索,为了缩短搜索时间和提高搜索精度,以趋化周期的细菌能量,即适应值的累加和作为标准,使能量较高的一半细菌繁殖,用分裂出来的细菌对能量较低的另一半细菌进行替代。在经过了固定周期繁殖之后,细菌会由于一定的原因进行区域的迁移,此时BFO算法设定一个迁移的概率,然后使每个细菌都能够进行随机的迁移,从而提高算法的全局寻优能力。

可以看出,BFO算法中细菌局部搜索能力的实现,主要是通过趋化算子对邻域的随机搜索并根据适应度进行方向调整来获得。很明显,由于缺乏对群体周边细菌信息的利用机制,造成算法收敛速度较慢,容易陷入局部最优的特征。在趋化周期结束后,算法根据细菌能量进行繁殖,这对寻优精度的提高有一定作用,但算法如已陷入局部最优,则早熟、不熟等问题也随之而生。

1.2 免疫算法

1993年,Mori等^[15]最早提出了免疫算法;1994年,Forbes等^[16]使用T细胞的自体-非自体识别思想,提出了免疫选择算法,并应用于模式识别等诸多领域;2000年,Castro等^[17]提出了克隆选择算法(Clonal Selection Algorithm, CSA),该算法不考虑抗体(群体)之间的亲和度(适应度),只关注抗体和抗原的亲和度,由于算法的高记忆和极强搜索能力,在模式识别和优化领域得到了广泛的应用。虽然免疫算法的原理较为复杂,但有研究表明,免疫算法具有非常好的融合性^[18],和主流的遗传算法、鸟群算法、蚁群算法等都有融合算法,但针对细菌觅食优化与免疫算法的融合鲜有研究。

1.3 基于免疫算法的改进细菌觅食算法

根据Dasgupta的研究,免疫算法可以分成3种类型,即:基于人工网络的免疫算法、负选择算法和克隆选择算法。本文借鉴免疫算法中的克隆选择思想,融合入细菌觅食优化中的繁殖算子中,进行克隆选择性繁殖,从而提高问题的求解精度。针对繁殖算子的流程改进如图1所示。

1.4 算法实现的基本流程

算法的主要实现步骤描述如下:

第1步 参数初始化。 N_{ed} 为迁移次数、 N_{re} 为繁殖次数、 N_e 为趋化次数; P_{ed} 为迁移概率; S 为细菌数; N_s 为游动次数。

第2步 初始化细菌位置。采用式(1)产生初始化位置,计算细菌群的初始化适应度 J 值。

$$X = x_{\min} + rand * (x_{\max} - x_{\min}) \quad (1)$$

第3步 迁移循环 $n_{ed} = 1:N_{ed}$;繁殖循环 $K = 1:N_{re}$;趋化循环 $j = 1:N_e$ 。

第4步 执行细菌趋化循环。

1) 翻转。细菌随机变换单位向量 $\varphi(i)$ 的方向,并按照式(2)更新细菌位置,其中 $C(i)$ 为细菌翻转的步长。

$$P(i, j+1, k, l) = P(i, j, k, l) + C(i)\varphi(i) \quad (2)$$

2) 游动。如果翻转的适应值改善,则按照翻转方向前行,

直至适应值不再改善或达到设定的最大移动步数 N_s 为止,称为游动。游动步长为 $C * V$,其中 V 为灵敏度, Δ 值为6000。

$$V(i) = (J_{\max} - J_i) / (J_{\max} - J_i + \Delta) \quad (3)$$

第5步 繁殖循环。趋化周期完成后,对当前细菌群体执行克隆免疫算法繁殖。

1) 计算所有细菌的适应度,从当前细菌群体中找出 n 个精英细菌,形成待克隆群体。其中, $n = n_{ed}$,随着算法的迭代,精英细菌越来越多,为了保持群体的多样性,在最后的精英细菌数目 n 为群体的一半。

2) 进行抗体的克隆繁殖,形成克隆群体 NC 。

① 对 n 个细菌按照适应度进行排序,最好的细菌排在最前面,并将适应度标准化。其中,精英细菌的克隆繁殖数目按式(4)进行,适应度标准化采用式(5)进行, NC_size 公式使得适应度越小的细菌形成的克隆群体越多。 β 是对细胞抗体抗原之间亲和度度量的递增函数,在此简称为克隆扩张系数 β ,本文 β 取值0.8,意味着总共产生80个克隆群体。

$$NC_size(i) = \sum_{i=1}^n \text{round}\left(\frac{\beta \times S}{i}\right) \quad (4)$$

$$\text{bestpop_fit} = \text{bestpop_fit} / \max(\text{bestpop_fit}) \quad (5)$$

② 对 NC 高频变异,随后进行随机交叉。其中,高频变异按照式(6)进行, δ 是高频变异概率,按照式(7)计算,适应度越大的细菌,变异的概率越大,搜索的空间越宽。

$$NC(i) = NC(i) + \delta * \text{randn}() * X_{\text{best}} \quad (6)$$

$$\delta = \exp(-\text{bestpop_fit}(n-i+1)) \quad (7)$$

③ 对 NC 群体进行随机交叉,产生 NS 个细菌后代,作为免疫细胞注入 NC 群体。交叉采用式(8)进行,其中 a, b, c, d 为克隆群体的互不相同的随机细菌。

$$\text{Cross_X} = NC(:, a) - NC(:, b) + NC(:, c) - NC(:, d) \quad (8)$$

④ 从 $NC + NS$ 的细菌克隆变异随机交叉后的群体中,选出适应度最好的 n 个个体,替代 S 个细菌群体中能量较差的 n 个个体。

第6步 迁移循环。繁殖算子完成后,将所有细菌按照能量进行排序,适应度较高的细菌群体 ($S * P_{ed}$),在解空间的定义域内随机初始化。

第7步 循环结束条件判断,满足则结束,输出结果。

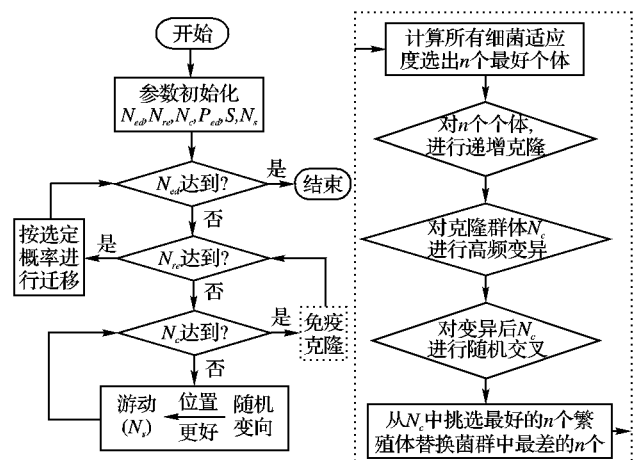


图1 基于免疫优化的细菌觅食优化算法流程

2 实验及结果分析

2.1 检验函数

选取进化计算中较难的5个测试函数(Rosenbrock、Rastrigin、Griewank、Ackley和Schwefel 2.26)^[19]作为检验函

数,这些函数都是 NP 难问题,都有全局极小,用这几个函数来检验算法改进的性能比较合适。定义它们分别为 $f_i(x)$:

$$f_1(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2];$$

$$x_i \in [-5, 10]$$

$$f_2(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10];$$

$$x_i \in [-5.12, 5.12]$$

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1;$$

$$x_i \in (-600, 600)$$

$$f_4(x) = 20 + e - 20 \exp\left(-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) -$$

$$\exp\left(-\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right);$$

$$x_i \in [-15, 30]$$

$$f_5(x) = 418.9829 * n - \sum_{i=1}^n x_i \sin \sqrt{|x_i|};$$

$$x_i \in [-100, 500]$$

以上这几个测试函数,优化过程的难度会随着测试函数的维度增加而出现急剧上升。对于维数较低的单峰函数,应用细菌觅食优化算法进行优化可以得到比较满意的结果,而对于维数较高的多峰函数,应用 BFO 算法优化的效果不怎么理想。现有针对 PSO、GA 等优化方法的改进,在以上测试函数中实现了一定的优化效果,但这些方法改进多是基于大量的群体规模和漫长的迭代步骤。此外,许多改进算法在利用函数 $f_1(x)$ 进行测试时,经常会出现局部最优,算法改进只是在精度上部分实现;在利用函数 $f_5(x)$ 进行测试时,由于全局最优的范围较狭窄,且距离局部最优值很远,导致算法改进较难,有些算法甚至并不针对函数 $f_5(x)$ 的测试情况进行比较。

2.2 实验参数设置

从算法流程上分析,本文算法仅仅在最后的克隆群体再选择阶段,进行精英筛选去替换现有群体中的较差细菌,这就涉及到适应度函数的计算而影响计算时间。当 β 为0.8时,繁殖群体为80个,相当于现有群体的2倍,现有参数设置的80次繁殖周期中,为了保证收敛时间,将迁移周期 N_{ed} 设置为10,经过测试,算法能够很快收敛,即本文算法基本没有依赖于过大的收敛代数来获得性能的成熟。

另外,基于免疫算法的克隆选择算子,具有一系列的参数,主要有高频变异概率 δ 、扩张系数 β 、克隆母体 n 。经过测试,扩张系数 β 取值从2,1.5,1,0.8,0.5,0.3进行设置,算法性能改变不大。迭代中精英群体 n 的适应度差值越来越小,导致克隆群体的变异概率 δ 越来越小且趋同,使得算法的局部

最优的可能越来越大。因此,精英细菌数目 n 随迭代而改变有利于算法的全局寻优。

最后,从改进克隆算法繁殖算子的再选择机制可以发现,现有替代机制并不考虑克隆体高频变异后的细菌针对现有细菌群体的改变,仅仅是从克隆体里面找出最好的 n 个个体进行替换,这种做法如果 β 过大,可能导致找出的克隆体全部是某个细菌的克隆体,从而降低了群体的多样性,如果出现在算法的早期,则容易产生早熟收敛,但却有利于提高算法的问题寻优精度。

因此,在基于免疫算法繁殖的 BFO 参数中,维度设置为30维,种群数目30,其他参数基本不变,总迭代时间近似相等。免疫算子中, $\beta = 0.8, n = \text{round}(n_{ed} * 3/2)$ 。对每个函数测试40次。

为验证算法性能,将本文算法与 BFO 标准算法、PSOBFO 算法和文献[20]中的 PSO 改进算法(以下简称 PSOPC 算法)进行对比。其中,PSOPC 算法和 PSOBFO 算法中 $S = 80, \omega$ 从0.9线性递减至0.7, C_1 从5线性递减至0.1, C_2 从0.5线性递减至0.1,最大迭代6000次。标准 BFO 算法中, $S = 30, N_{ed}$ 迁移6次,迁移概率为0.25, N_{re} 繁殖4次, N_c 趋化40次,迭代次数用 $N_{ed} * N_{re} * N_c$ 衡量,约960次。其他参数设置见文献[2,20-21]。

2.3 实验结果及分析

在3个函数测试40次的优化结果如表1所示。由于算法的概率收敛特性,本文选取最优值、最差值、平均值、标准差4个指标来评价算法性能度量。

从表1可看出:针对测试函数 f_3, f_4 ,标准 BFO 算法基本上能够在现有的迭代步骤内找到全局最优值,但是精度并不高。本文算法在现有迭代步骤内,求解的精度较高,进行对比的 PSOPC 算法和 PSOBFO 算法在较长的迭代步骤内也获得了较高的求解精度,都优于基本的 BFO 算法。其中,对于测试函数 f_3 ,本文算法能够快速收敛且获得全局最优解,算法精度较高,40次测试中全部获得全局最优,其最优解的最差 X 维度值在E-07等级;当迭代次数在600~850时,对数取值出错,此时算法已经收敛获得了全局最优。本文认为免疫 BFO 针对问题 f_3 具有适用性且能够稳定获得全局最优。算法在测试函数 f_4 上的随机测试的最优值为 $8.88\text{E}-16$,位置为E-15级别。算法收敛曲线如图2所示。

测试函数 f_5 极易发生早熟收敛,现有研究中,PSOPC、标准 BFO 等算法都容易陷入局部最优。本文算法在 f_5 上的随机测试最优值是0.9489,各个维度的最优位置均在421左右,仅有11维处于420左右,此时没有完全成熟的维度占比36.6%。另外,免疫 BFO 在测试问题函数 f_5 上的算法性能较为稳定,最优值在0.6左右,40次鲁棒性测试仅有一次算法

表1 测试函数的算法测试结果

测试函数	指标	PSOPC 算法 ^[20]	标准 BFO 算法 ^[2]	PSOBFO 算法 ^[8]	本文算法
f_1	平均(标准差)	24.378(4.326)	29.63(0.22)	2.544(0.16)	28.644(0.18)
	[最差,最优]	[15.56,28.71]	[29.31,29.91]	[2.513,2.641]	[28.03,28.676]
f_2	平均(标准差)	28.633(7.821)	2.26(0.37)	0.0159(0.003)	0
	[最差,最优]	[36.296,21.87]	[1.29,3.04]	[0,0.0351]	[0,0]
f_3	平均(标准差)	0.0083(0.006)	1.004(0.016)	1.56E-03(0.004)	0
	[最差,最优]	[0.8E-05,0.03]	[0.926,1.253]	[0,2.122E-02]	[0,0]
f_4	平均(标准差)	3.8E-13(4.7E-13)	0.393(0.042)	8.9E-15(8.9E-16)	3.1E-15(9.8E-16)
	[最差,最优]	[6.5E-015,2.6E-012]	[0.29,0.43]	[8.9E-15,8.9E-15]	[8.8E-16,4.4E-15]
f_5	平均(标准差)	288.447(227.18)	807.31(56.71)	969.7(107.45)	8.346(6.07)
	[最差,最优]	[80.017,669.167]	[687.7,938.6]	[882.9,1347.1]	[0.43,2.18+E02]

失效,其他基本找出了问题测试函数的全局最优,且性能稳定。免疫 BFO 性能优于 PSOPC,40 次测试中,除去一次不稳定,其他 39 次的测试较为稳定,最优值在 0.43 左右,但相对于基本 BFO 改善不大。本文算法的迭代曲线如图 3 所示。

针对函数 f_1 , 本文算法随着迭代,很快进入局部最优区域(此时适应值为 28),在迭代的第 120 次左右算法局部收敛,后续迭代基本围绕该点进行局部优化。40 次测试数据表明,本文算法相对于标准 BFO 的性能改进不是很明显,但本

文算法的鲁棒性能较好,寻优结果更加趋于稳定。因此,本文算法不太适应于 f_1 类问题的全局最优搜索。

而对于测试函数 f_2 , 本文算法很快获得全局最优,其在问题 f_2 上的最优解为 0,优化位置 X 的级别达到 E-09 级,且测试结果在第 800 次左右便接近了全局最优,所以本文算法优于标准 BFO 算法、PSOBFO 算法和 PSOPC 算法,相对性能非常稳定,且每次都能求得问题的全局最优解,非常适合于该类问题的求解。相关的曲线如图 4 所示。

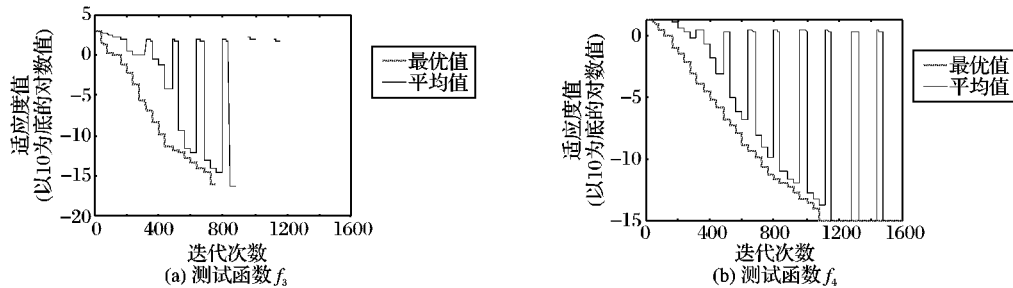


图2 基于30维 f_3 、 f_4 的本文算法收敛性能测试曲线

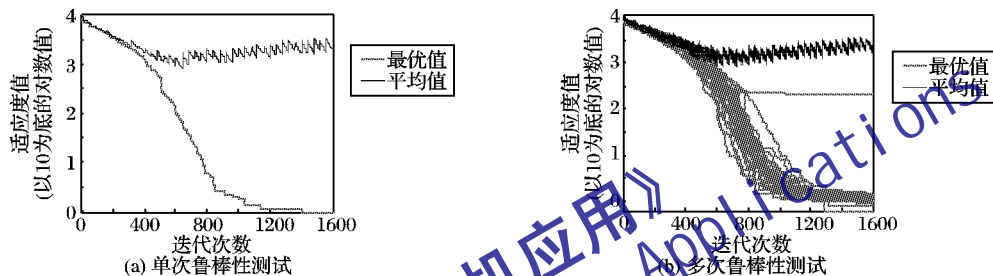


图3 基于30维 f_5 的本文算法单次和多次鲁棒性能测试曲线

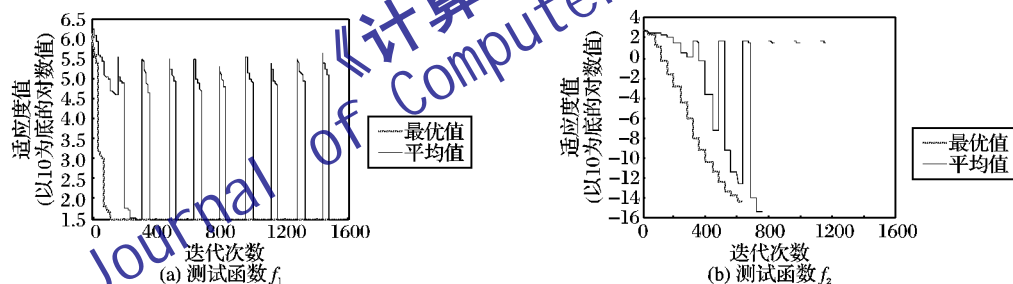


图4 基于30维 f_1 、 f_2 的本文算法收敛性能测试曲线

3 结语

由以上算法性能的测试可知,对于问题测试函数 f_2 、 f_3 、 f_4 , 它们虽然是多峰函数,全局极值附近有许多拐点,但极值区域的通道稍微宽敞,本文算法具有较强的适应性,每次都能非常准确地找到全局最优解,并获得很高的求解精度;但针对 f_5 的测试表明,算法虽然找到了全局最优区域,但并不能完全成熟收敛;对于问题函数 f_1 , 这是一种香蕉型函数,全局极值在一条非常狭窄的通道中,本文算法基于精英繁殖,不能有效跳出局部最优的通道,算法虽获得了较好的稳定性,却陷入了局部最优。因此,本文算法对于多峰函数 f_2 和 f_4 , 具有很强的问题适用性,可以有效地找到全局最优,在很短的迭代周期极大提高求解精度。另外,从以上 5 个不同类型的测试问题可知,免疫 BFO 不会降低原算法的求解性能。

另外,免疫克隆是一种单亲变异繁殖,虽然通过随机交叉产生了部分细菌群体,但每次这些群体的适应度如果不能很快得到改善,那么在用繁殖精英细菌去替代较差的部分细菌时,随机交叉根本就不起作用,这就能很好地解释算法在 f_1 上陷入局部收敛的原因,对于这个问题的解决,将进行进一步

的相关研究。

参考文献:

- [1] MILLER S, AIRAGHI S, MARCHELO J, *et al.* Optimization algorithms based on a model of bacterial chemotaxis [C]// Proceedings of the 6th International Conference on Simulation of Adaptive Behavior. Boston: MIT Press, 2000: 375-384.
- [2] PASSINO K M. Biomimicry of bacterial foraging for distributed optimization and control [J]. IEEE Control Systems Magazine, 2002, 22(3): 52-67.
- [3] LIU Y, PASSINO K M, POLYCARPOU M M. Stability analysis of m-dimensional asynchronous swarms with a fixed communication topology [J]. IEEE Transactions on Automatic Control, 2003, 48(1): 76-95.
- [4] MISHRA S. A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(1): 61-73.
- [5] MAJHI R, PANDA G, MAJHI B, *et al.* Efficient prediction of stock market indices using Adaptive Bacterial Foraging Optimization (ABFO) and BFO based techniques [J]. Expert Systems with Applications, 2009, 36(6): 10097-10104.

(下转第 653 页)

$v = (0.108\ 8, 0.226\ 5, 0.132\ 9, 0.146\ 0, 0.161\ 7, 0.224\ 0)$

所以这6个备选方案的排序为 $x_2 > x_6 > x_5 > x_4 > x_3 > x_1$,即根据各位专家的评估和考虑这4个属性的情况下,应该选择第2个方案。从原始数据本身可以看到,对于大部分属性,每个专家对方案 x_2 给出的属性估计值相对较大,其他方案次之,符合最终给出的决策结果,这也说明上述决策过程的科学性和可行性。

将上述结果与文献[11]比较发现,本文方法更具有灵活性,针对不同方案和属性,每个专家的权重并不完全相同,也符合具体问题具体分析的原则。同时,文献[11]处理的是有偏好信息,本文处理的是对专家权重完全未知,依据权重确定准则和所给数据挖掘出权重信息,同时结合文献[3]中属性权重的确定方法,可以方便地给出多专家多属性群决策的方法。

4 结语

在不同对象和属性下,当某专家的估计值与所有专家估计值的均值越靠近时,则其专家权重就越大;反之就越小。基于此准则提出了一种新的专家权重的确定方法,通过建立非线性规划模型,利用拉格朗日乘数法得到专家权重的计算公式。然后利用区间算术平均算子将决策矩阵集成为综合决策矩阵,结合属性权重将其转化为综合属性值,利用可能度建立所有对象的比较可能度矩阵,再用排序向量法进行排序。实例也证明该方法易于理解,切实可行。

参考文献:

- [1] 张文修,梁怡,吴伟志. 信息系统与知识发现[M]. 北京:清华大学出版社,2003:12-56.
- [2] QINAN Y H, LIANG J DANG Y. Interval ordered information systems[J]. Computers and Mathematics with Applications, 2008, 56(8):1994-2009.
- [3] 徐泽水. 不确定多属性决策方法及应用[M]. 北京:清华大学出版社,2004:62-124.
- [4] XU Z S. On multi-period multi-attribute decision making[J]. Knowledge-Based Systems, 2008, 21(2):164-171.
- [5] 徐泽水. 直觉模糊信息集成理论及应用[M]. 北京:科学出版社,2008:25-165.
- [6] 徐泽水. 区间直觉模糊信息的集成方法及其在决策中的应用[J]. 控制与决策, 2007, 22(2):215-249.
- [7] 杨威,庞永峰. 一个基于不确定动态几何加权平均算子的多属性决策方法[J]. 数学认识与实践, 2011, 41(8):14-18.
- [8] CHEN ZHIPING, YANG WEI. A new multiple attribute group decision making method in intuitionistic fuzzy setting[J]. Applied Mathematical Modelling, 2011, 35:4424-4437.
- [9] 陈根忠,刘湘伟,郭世杰. 基于区间数灰色关联分析的雷达对抗目标选择方法[J]. 电子信息对抗技术, 2010, 25(7):57-60.
- [10] 万树平. 区间型多属性群体专家权重的确定方法[J]. 应用数学与计算数学学报, 2008, 22(2):109-116.
- [11] 陈晓红,刘益凡. 基于区间数群决策矩阵的专家权重确定方法及其算法实现[J]. 系统工程与电子技术, 2010, 32(10):2128-2131.
- [12] 兰继斌,胡明明,叶新苗. 基于相似度的区间数排序[J]. 计算机工程与设计, 2011, 32(4):1419-1421.
- [13] NAYAGAMA V L G, MURALIKRISHNAN S, SIVARAMAN G. Multi-criteria decision-making method based on interval-valued intuitionistic fuzzy sets[J]. Expert Systems with Applications, 2011, 38(3):1464-1467.
- [14] NAYAGAMA V L G, SIVARAMAN G. Ranking of interval valued intuitionistic fuzzy sets[J]. Applied Soft Computing, 2011, 11(4):3368-3372.
- [15] CHEN TING-YU, WANG HSIAO-PIN, LU YEN-YU. A multicriteria group decision-making approach based on interval-valued intuitionistic fuzzy sets: A comparative perspective[J]. Expert Systems with Application, 2011, 38(6):7647-7658.
- [6] DATTA T, MISRA I S, MANGARAJ B B, et al. Improved adaptive bacteria foraging algorithm in optimization of antenna array for faster convergence[J]. Progress in Electromagnetics Research C, 2008, 1:143-157.
- [7] CHEN H, ZHU Y, HU K. Self-adaptation in bacterial foraging optimization algorithm[C]// Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering. Piscataway, NJ: IEEE Press, 2008:1026-1031.
- [8] BISWAS A, DASGUPTA S, DAS S, et al. Synergy of PSO and bacterial foraging optimization — A comparative study on numerical benchmarks[EB/OL]. [2010-05-10]. http://www.softcomputing.net/hais07_2.pdf.
- [9] BAKWAD K M, PATTNAIK S S, SOHI B S, et al. Hybrid bacterial foraging with parameter free PSO[C]// NaBIC 2009: World Congress on Nature & Biologically Inspired Computing. Piscataway, NJ: IEEE Press, 2009:1077-1081.
- [10] TANG W J, WU Q H, SAUNDERS J R. A bacterial swarming algorithm for global optimization[C]// Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press, 2007:1207-1212.
- [11] CHU Y, MI H, LIAO H, et al. A fast bacterial swarming algorithm for high-dimensional function optimization[C]// Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press, 2008:3135-3140.
- [12] DASGUPTA A, DASGUPTA S, DAS S, et al. A synergy of differential evolution and bacterial foraging optimization for global optimization[J]. Neural Network World, 2007, 17(6):607.
- [13] KIM D H, ABRAHAM A, CHO J H. A hybrid genetic algorithm and bacterial foraging approach for global optimization[J]. Information Sciences, 2007, 177(18):3918-3937.
- [14] LUH G C, LEE S W. A bacterial evolutionary algorithm for the job shop scheduling problem[J]. Journal of the Chinese Institute of Industrial Engineers, 2006, 23(3):185-191.
- [15] MORI K, TSUKIYAMA M, FUKUDA T. Immune algorithm with searching diversity and its application to resource allocation problem[J]. Transactions-Institute of Electrical Engineers of Japan, 1993, 113C(10):872-878.
- [16] FORREST S, PERELSON A S, ALLEN L. Self-nonself discrimination in a computer[C]// IEEE Symposium on Research in Security and Privacy. Piscataway, NJ: IEEE Press, 1994:202-212.
- [17] CASTRO L N, ZUBEN F J. Artificial immune system: Part I: basic theory and applications[R]. Campinas, Brazil: State University of Campinas, School of Computing and Electrical Engineering, 1999.
- [18] 莫宏伟. 人工免疫系统原理与应用[M]. 哈尔滨:哈尔滨工业大学出版社, 2002.
- [19] YAO X, LIU Y, LIN G. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2):82-102.
- [20] HE S, WU Q H, WEN J Y, et al. A particle swarm optimizer with passive congregation[J]. Biosystems, 2004, 78(1/2/3):135-147.

(上接第637页)