

用于多峰函数优化的小生境人工蜂群算法

毕晓君, 王艳娇

(哈尔滨工程大学信息与通信工程学院, 黑龙江 哈尔滨 150001)

摘要: 针对人工蜂群算法(artificial bee colony algorithm, ABC)在多峰优化时只能找到一个最优解, 难以找到所有全局和局部最优解的问题, 提出一种小生境 ABC 算法。利用小生境技术维持种群的多样性, 使算法不再单单收敛于全局最优解; 借助 ABC 算法具有较高寻优精度的特点精确寻找各个峰值点。对标准测试函数进行仿真, 结果表明, 小生境 ABC 算法能够准确、有效地找到多峰问题的全局和局部最优解, 具有良好的多峰优化性能。

关键词: 人工蜂群算法; 小生境技术; 多峰函数优化

中图分类号: TP 301.6

文献标志码: A

DOI: 10.3969/j.issn.1001-506X.2011.11.41

Niche artificial bee colony algorithm for multi-peak function optimization

BI Xiao-jun, WANG Yan-jiao

(College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China)

Abstract: Since it is difficult to find all the optimal solutions for artificial bee colony algorithm (ABC) used in multimodal optimization problems, which can only find a global optimal solution, this paper presents a niche ABC. Niche technology can maintain the diversity of population, so that the algorithm will not converge to the single global optimal solution. The ABC has higher accuracy to find the characteristics of the various peaks accurately. The simulation results of five standard test functions show that the niche ABC can effectively find all of the global and local optimal solutions for the multi-peak problems, and it is a good multi-peak optimization algorithm.

Keywords: artificial bee colony algorithm (ABC); niche technology; multimodal function optimization

0 引言

现实世界中,会遇到许多这样的问题,全局最优解很难满足,而往往选择那些已经能达到工程需要的有意义的局部最优解,正基于此,使得多峰优化问题成为人们一直关注的焦点。对于这类问题,传统的收敛于全局最优的进化算法如遗传算法、粒子群算法、克隆选择算法、免疫算法、人工鱼群算法、蚁群算法等都不能直接应用^[1-6],通常结合小生境技术,其中以排挤机制^[7]和适应值共享机制^[8]最为流行。但由于这些算法本身易陷入局部最优,参数设置困难,使得多峰函数通常不会得到精确解,多样性(搜索到的峰的数目)也不足。

人工蜂群(artificial bee colony, ABC)算法是 2005 年提出的种群优化方法,具有设置参数少、收敛速度快、解精度高的优点。文献[9]中指出与差分算法、粒子群算法相比,ABC 算法更容易获得全局最优解。对于这样一种优秀的算法,在处理多峰函数优化时,也仅能找到一个全局最优解,而不能搜索到所有全局和局部最优解。本文结合小生境技术对 ABC 算法的结构、选择策略等进行修改,提出一

种解决多峰函数优化问题的小生境人工蜂群(niche artificial bee colony, NABC)算法。对多峰函数优化领域广泛采用的问题进行测试,结果表明本文提出的算法能够有效、准确地找到全部峰值,而且具有较少的最小估计量,成功解决了多峰函数优化问题。

1 ABC 算法

ABC 算法模拟实际蜜蜂采蜜机制处理函数优化问题,将人工蜂群分为 3 类:引领蜂、跟随蜂和侦查蜂。引领蜂、跟随蜂用于蜜源的开采,侦查蜂避免蜜源种类过少。

函数优化问题可以抽象为蜜蜂寻找最优蜜源的过程,具体实现方式为:

步骤 1 产生初始种群

按式(1)随机产生 $2N$ 个位置。

$$V_{ij} = x_{jL} + \text{rand}(x_{jH} - x_{jL}) \quad (1)$$

式中, V_{ij} 为第 i 个蜜蜂第 j 维对应的搜索后的位置; x_{jL} 和 x_{jH} 代表第 j 维变量的上下界。选取适应度值优的 N 个组成蜜源集 P 。

步骤 2 引领蜂开采新蜜源组成蜜源集 P_1

引领蜂按式(2)搜索新蜜源。

$$V'_{ij} = x_{ij} + R_{ij}(x_{ij} - x_{kj}) \quad (2)$$

式中, V'_{ij} 为新的蜜源位置; x_{ij} 为蜜源 i 的第 j 维位置; x_{kj} 为随机选择的不等于 i 的蜜源 k 的第 j 维位置; R_{ij} 为 $[-1, 1]$ 间的随机数。

步骤 3 确定标记蜜源集 P_2

比较 P 和 P_1 中对应个体优劣, 若搜索后蜜源个体有所改善, 则代替先前蜜源, 构成蜜源集 P_2 。

步骤 4 跟随蜂搜索蜜源群体 P_3

跟随蜂根据处于蜜源处的引领蜂释放的花蜜信息(抽象为适应度), 按轮盘赌方式选择蜜源, 并在其附近按式(2)产生新蜜源。

步骤 5 最终蜜源集 P 的确定

取蜜源集 P_2 和 P_3 中适应值较优的 N 个作为蜜源集 P 。

步骤 6 判断是否出现侦查蜂, 设置参数 $lim\ it$, 若某些蜜源经 $lim\ it$ 次循环不变, 放弃该蜜源, 相应引领蜂变成侦查蜂, 按式(1)随机产生新蜜源, 并更新蜜源集 P 。

步骤 7 用新确定的蜜源集 P , 从步骤 2 开始重新搜索, 直到满足终止条件。

ABC 算法的程序流程如图 1 所示。

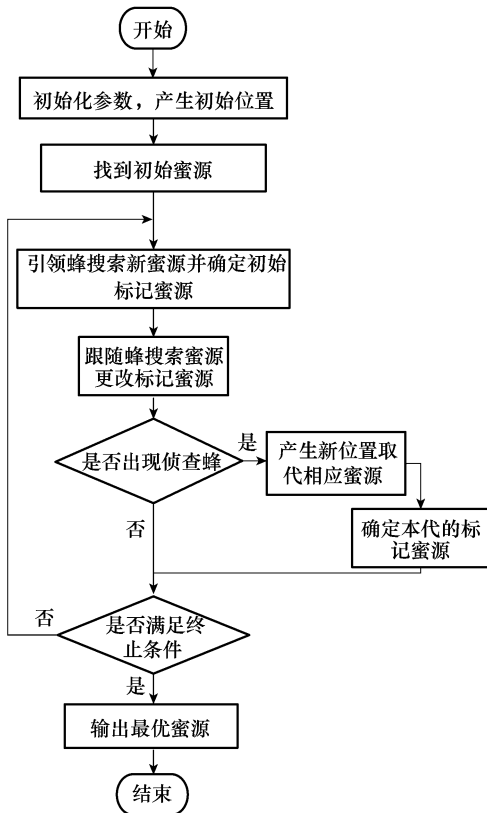


图 1 ABC 算法流程图

2 小生境技术

为了更好的理解本文提出的解决多峰优化问题的 ABC 算法, 下面简单介绍与之相关的小生境技术。

2.1 适应值共享小生境

文献[10]提出的适应值共享小生境技术引入“分享机制”使得高、低峰以平等机会被选中, 可以有效的避免漏峰, 并且不会全部聚集到某一个峰上。非常适合解决多峰问题。但传统的共享机制存在预设半径的难题, 调试复杂、效果不佳。文献[11]引入小生境识别技术(Niche identification techniques, NIT)动态识别确定小生境的边界、半径等信息, 避免预设半径, 是目前效果较好的一种适应值共享小生境方法, 也使适应值共享的方法得以有效应用。

2.2 排挤小生境

为了避免算法收敛于单个的最优解, 避免种群多样性的降低, 通常选用排挤策略。方法为: 对于每个子代个体, 从父代中选择 CF 个个体, 根据某种距离定义选择最近的一个个体, 若优于该个体则替换, 否则, 维持子代个体不变^[12]。

3 NABC 算法

ABC 算法只能锁定一个全局最优解, 不符合多峰函数优化的要求, 为此本文对 ABC 算法进行五方面的修改。

3.1 标记蜜源 P_2 的确定方式的改进

在 ABC 算法中, 标记蜜源 P_2 是由父代与子代蜜源直接对比适应度值取优确定的, 这样的方式有利于算法较快的向全局最优解靠近, 但不适合解决多峰函数优化问题, 为此, 进行相关修改: 子代和父代个体考虑地形信息进行两两配对, 比较每对个体对应的适应度值是否有所改善, 若改善, 则用子代个体替换父代个体确定标记蜜源。这样有利于个体向距离自己最近的峰靠近, 而不是向全局最优的峰靠近。具体的配对方式为: 计算子代和父代个体间的欧式距离, 选取与第一个父代个体最相近的子代个体, 配成一对, 第二个父代个体从余下的子代个体中选择与之最为相近的个体配对, 依此方式一一配对。

3.2 选择策略的调整

与函数优化问题一样, 运用 ABC 算法求解多峰优化问题时也存在易陷入局部最优——不能精确搜索到某些峰值点的现象。这是由跟随蜂选择蜜源时依据的轮盘赌方式过于贪婪、种群多样性降低过快造成的。

自由搜索算法^[13]提出了一个重要概念——灵敏度, 通过与信息素(与优化问题的适应度值有关)配合选择区域, 具体为:

- (1) 计算出 N 个个体的适应度 $f(X)$;
- (2) 计算第 i 个区域的信息素 $nf(i)$, 如式(3)所示;

$$nf(i) = \begin{cases} \frac{f(i) - f_{\min}}{f_{\max} - f_{\min}}, & f_{\max} \neq f_{\min} \\ 0, & \text{其他} \end{cases} \quad (3)$$

- (3) 随机产生第 j 个个体的灵敏度 $S(j) \sim U(0, 1)$;
- (4) 找出配合第 j 个个体灵敏度的区域 i ; 随机找出 i , 满足 $nf(i) \geq S(i)$ 。

由上述的“灵敏度—信息素”模型可以看出: 由于灵敏度的随机性, 理论上任何区域都可以被搜索到, 这就在很大

程度上避免了陷入局部最优;必须满足步骤(4),就使算法有了明确方向,决定了目标函数在搜索空间中的收敛和发散。这种区域选择的方式与跟随蜂选择蜜源的方式是类似的,所以可以考虑用灵敏度与信息素配合的方式代替轮盘赌方式选择蜜源。

3.3 跟随蜂选择蜜源的评价方式的改进

在 ABC 算法中,跟随蜂依据蜜源的适应度值选择合适蜜源,这样会使跟随蜂向最佳蜜源靠近。但对于多峰优化问题中的不等峰问题,如果仅以适应度值作为标准,那些函数值较小的峰就很容易被漏掉。而赋予每个峰值点同样的最大适应度值的共享适应度值方法可以克服这一缺点,可以使跟随蜂以均等的机会向各个峰值点移动。有效的避免了多样性的损失,因此本文以文献[11]中介绍的共享适应度值作为新的评价函数。

3.4 每次迭代的最终种群 P 的确定

ABC 算法中,每代确定的最终迭代种群是由标记蜜源和跟随蜂中适应度值较优的个体组成的,这种方式也是有利于向全局最优解靠近,而不适合求解多峰函数优化问题。为此,本文选用排挤方式确定种群。但传统的排挤方式是依据个体的适应度值确定种群的,对于不等峰问题,有可能将峰值点较低的个体排挤掉,从而出现漏峰的现象,显然是不合适的,为此,本文将个体适应度值改为共享适应度值参与排挤策略。

3.5 设置外部种群

运用小生境时有时会造成丢失已得峰值点的现象,为此设置一个外部种群用于存储已搜索到的峰值点。方法为:将第一次迭代的峰值点直接存储于外部种群,从第二次迭代开始,将本次迭代中的峰值点与外部种群中的峰值点进行小生境判断,取每一个小生境中的最优个体更新外部种群。

3.6 NABC 算法的操作流程及复杂度分析

本文将引入小生境技术做上述修改的 ABC 算法称为 NABC 算法。NABC 算法步骤为:

- 步骤 1 初始化参数:引领蜂、跟随蜂数目,迭代次数,外部种群等;
- 步骤 2 产生初始蜜源 P;
- 步骤 3 引领蜂按式(2)搜索产生新蜜源 P1;
- 步骤 4 按照 3.1 节方法在 P 和 P1 中挑选个体作为标记蜜源 P2;
- 步骤 5 跟随蜂按照 3.3 节的评价方式和 3.2 节的选择策略选择 P2 中合适蜜源进行搜索开探新蜜源 P3;
- 步骤 6 按照 3.4 节方式由 P2 和 P3 确定本次迭代的最终标记蜜源 P;
- 步骤 7 按照 3.5 节方法更新外部种群;
- 步骤 8 判断是否满足终止条件:若不满足则转到步骤 3,否则转到下一步;
- 步骤 9 输出外部种群。

算法的复杂度分析:设引领蜂、跟随蜂数目都为 n 。NABC 算法的主要计算时间有:计算一次适应度函数的时间 t_1 ,计算一次两个个体间距离的时间 t_2 ,计算一次共享函数的时间 t_3 ,计算一次式(2)的时间 t_4 。所以,在每一次进

化代中,各过程所需时间及相应的时间复杂度如表 1 所示。

表 1 时间复杂度分析

过程	时间	时间复杂度
种群适应度	$n \times t_1$	$O(n)$
小生境识别	$n(n-1)t_2/2$	$O(n^2)$
种群配对	$(2n+(2n-2)\cdots+4)t_2/2$	$O(n^2)$
种群共享适应度	$n(n-1)t_3/2$	$O(n^2)$
产生新蜜源	$2 \times n \times t_4$	$O(n)$

由表 1 可以看出,本文算法总的时间复杂度为 $O(n^2)$ 。

4 仿真试验及结果分析

4.1 测试函数

为验证本文提出的 NABC 算法的性能优劣,选取 5 个在多峰函数优化领域被广泛采用的标准测试函数进行试验。

测试函数 1^[14]

$$f_1(x) = \sin^6(5.1\pi x + 0.5)$$

式中, $x \in [0, 1]$, 含有 5 个等距等高的峰,峰值点为 1、1、1、1、1。

测试函数 2^[14]

$$f_2(x) = e^{-4 \ln 2(x-0.0667)^2/0.64} \sin^6(5.1\pi x + 0.5)$$

式中 $x \in [0, 1]$, 含有 5 个等距不等高的峰,峰值点为 1、0.847 2、0.515 3、0.225 1、0.070 6。

测试函数 3^[14]

$$f_3(x) = x \sin^6(5.1\pi(x^2 - 0.57))$$

式中, $x \in [0, 1]$, 含有 5 个不等距不等高的峰,峰值点为 0.929 7、0.817 6、0.687 2、0.525 8、0.285 9。

测试函数 4

$$f_4(x) = x(x+1) \sin[(2x-0.5)^2\pi - 1]$$

式中, $x \in [-1.5, 1]$ 。

测试函数 5^[5]

$$f_5(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$$

式中, $x, y \in [-6, 6]$ 。4 个峰值均为 200。

函数 1~函数 3 一般用来考察算法解决多峰优化问题的执行能力,而对于函数 4、函数 5 这样的复杂函数一般方法很难精确找到所有峰值点,所以可以用来考察算法的搜索能力。

4.2 性能评价标准

本文在评价 NABC 算法性能时选用目前常用的多峰函数测试标准^[15]。

有效峰数:如果种群中至少有一个个体达到某峰值的 99%时,则代表该峰被搜索到,称作一个有效峰。所有有效峰的总数为有效峰数。

最大峰值比:实际搜索到的各个不同的峰值之和与理论最优峰之和的比称为最大峰值比。该比值用来衡量算法的局部寻优能力,当算法识别的有效峰数越多且对应的实际峰越精确,最大峰值比越大,最大值为 1,表明准确找到所有峰值点。

最小计算量:计算量可以用评价个体的累计次数来估计,对个体评价一次是指对一个个体循环一次,算法的最小计算量约为“种群规模×进化代数”。

精确度:实际峰值与算法运行得来的有效峰值之差的

平方和。精确度越接近于 0,结果越精确。

4.3 实验结果及分析

以下实验仿真都是在 CPU Intel Centrino Duo T7250、2.0 GHz 主频、1 G 内存、Widows7 操作系统的计算机上实现的,程序采用 Matlab7.5 语言实现。

4.3.1 主要参数对算法性能的影响

为了测试 NABC 算法的主要参数(即引领蜂、跟随蜂数目)变化对算法性能的影响,改变种群数目大小,以有效峰数、最大峰值比及运行时间作为性能评价标准对函数一进行测试,参数变化对算法的性能影响如表 2 所示。

表 2 引领蜂数目对算法的影响

引领蜂数目	有效峰数	最大峰值比	运行时间/s
10	4.7	0.939	1.065 0
15	4.9	0.979	1.336
20	5	0.999	2.155
30	5	1	2.640
40	5	1	3.611
50	5	1	4.993
100	5	1	13.945

NABC 算法其他参数设置为:共享参数取常用值 2;拥挤因子(crowding factor, CF)为 20;迭代次数为 100。每个测试函数独立运行 10 次,取其性能平均值。

从表 2 可以看出:当引领蜂即种群数目较小时($n=10$),并不是每次都能搜索到全部峰值点,而当种群数目增加时,精确搜索到全部峰值点的可能性增大,但也增加了运行时间。也就是说,为了平衡解的精确性和运行效率需要针对具体问题选取合适的种群数目。如在这一函数寻优时,当种群数目取为 30 时就可以得到满意的结果。

4.3.2 标记蜜源确定方式的有效性

为了验证本文算法确定标记蜜源 P_2 方式的有效性,将本文算法与采用原有确定方式的基本算法对函数一进行测试,结果如表 3 所示。

表 3 改进前后的性能比较结果

方法	(自变量,函数值)
原有方法	(0.062 8,0.987 6);(0.259 4,0.990 7);(0.459 2,1); (0.659 5,0.985 1);(0.851 2,1)
改进方法	(0.066 8,1);(0.262 9,1);(0.459 0,1); (0.655 1,1);(0.851 1,1)

从表中数据可以看出:本文算法可以精确锁定各个峰值点,而原有方法却不能,这是由于原有确定方式使种群快速的向全局最优解靠近,使大部分个体都分布在最优点附近,而降低了对其他峰的搜索能力,因而不利于寻找所有峰值点。而本文提出方法可以使各个点向离自己最近的峰值点靠近,很好的保持了种群多样性,所以可以获得很好的探索峰的能力。

4.3.3 算法比较

为测试本文提出的 NABC 算法解决多峰函数优化问题效果的好坏,将本文算法与优育子群法^[14]、简单子群法^[14]和文献[15]的方法进行对比。鉴于公平性原则,同其他文献一样,独立进行 5 次实验,取最好结果。算法性能的

比较结果如表 4 所示,其中,种群规模(population number, PN)、进化代数(evolutional generation, EG)、有效峰数(effective peak number, EPN)、精确度(accuracy, A)和最小估计量(minimum estimation, ME)。

表 4 各种方法性能比较结果

测试函数	F_1	F_2	F_3	
简单子群法	PN	40	100	100
	EG	200	400	500
	EP	5	4	4
	A	4.0×10^{-3}	2.4×10^{-3}	4.6×10^{-3}
	ME	8 000	40 000	50 000
优育子群法	PN	10	10	10
	EG	50	150	180
	EP	5	5	5
	A	1.2×10^{-5}	4.1×10^{-5}	5.1×10^{-5}
	ME	500	1 500	1 800
文献[15]中方法	PN	40	50	50
	EG	10	10	15
	EP	5	5	5
	A	1.1×10^{-5}	7.7×10^{-6}	3.6×10^{-6}
	ME	400	500	750
本文算法	PN	30	30	50
	EG	10	15	15
	EP	5	5	5
	A	4.74×10^{-6}	1.6×10^{-7}	1.0×10^{-7}
	ME	300	450	750

注:上述对比算法的实验结果取自相应参考文献。

从表 4 中数据可以看出:在搜索到的有效峰方面,除简单子群法外,其他算法都可以搜索到全部的峰值;在精确度方面,无论处理上述哪种函数,本文算法都获得了最精确的结果;在最小估计量方面,本文算法的最小估计量最少。即与其他算法相比,本文方法的多峰函数优化效果更好。这是因为本文方法一方面设计了新的标记种群确定方式使各个点有方向性的靠近离自己最近的峰值点,配合小生境技术的使用,有效的保持物种多样性,避免漏峰现象的产生;另一方面,改进了跟随蜂选择蜜源的方式,加快了具有较强搜索能力的人工蜂群算法的收敛。这样就会在很少的迭代次数内精确搜索到各个峰,必然使得最小估计量小、精度高。

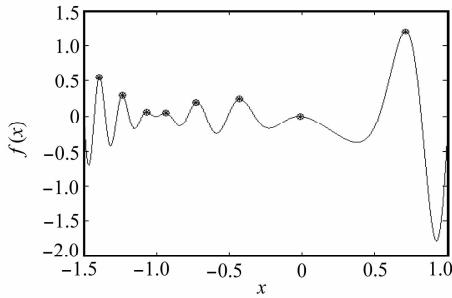
为了更进一步说明本文方法具有较高的寻优精度,对较复杂的函数 4 和函数 5 进行测试,表 5、表 6 和图 2 给出测试结果。

表 5 函数 4 的优化结果

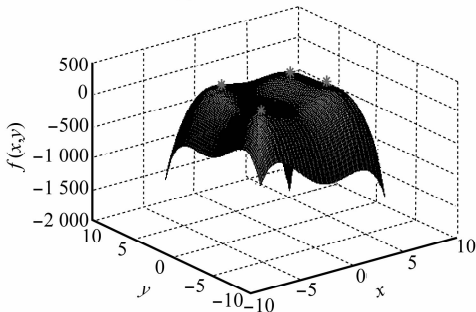
最优解号	最优解点	函数值
1	0.716 8	1.213 4
2	-1.396 4	0.551 9
3	-1.238 3	0.292 5
4	-0.426 3	0.244 4
5	-0.723 3	0.199 3
6	-1.068 8	0.066 7
7	-0.940 1	0.049 9
8	-0.016 2	0.001 7

表 6 函数 5 的优化结果

序号	最优解点	函数值
1	2.999 999 884 705 3 1.999 999 908 184 7	199.999 999 999 999 2
2	-2.805 118 029 293 6 3.131 312 812 255 4	199.999 999 999 996 4
3	-3.779 307 929 967 1 -3.283 184 122 812 8	199.999 999 999 654 8
4	3.588 114 660 871 9 -1.857 877 300 626 2	199.998 135 610 526 9



(a) 函数4的寻优效果



(b) 函数5的寻优效果

图 2 NABC 算法寻优效果直观图

从表 5、表 6 和图 2 可以看出,本文算法可以精确找到复杂多峰函数的全部峰值,寻优精度极高。

综上所述,本文算法无论处理上述的哪种测试函数,都能在较少的最小估计量的情况下得到全部峰值数,并且较准确地锁定各个峰值点,即本文算法在处理多峰函数优化上是一种计算量少、多样性好、精确度高的算法。

5 结束语

针对多峰函数优化问题,本文提出了一种小生境人工蜂群算法——NABC 算法。算法中的每个个体都能够逐渐靠近离其最近的峰;利用动态边界识别的小生境技术保证跟随蜂在各个峰左右进行开采,精确定位各个峰的位置;利用排挤机制确定蜜源位置,使算法不止收敛于单个的最优峰,增加了算法的多样性;增设外部种群,避免了多样性的降低。另外,算法本身不需额外设置参数,使得算法易于实施。实验表明:该算法对于多峰函数优化问题具有计算量少、多样性好、精度高的优点,是一种有效的多峰处理方法,较为成功的拓宽了人工蜂群算法的应用领域。

参考文献:

[1] Alfayo A A, Josiah M L. Optimization of multimodal models in mechanical design by a niche hybrid genetic algorithm[C] // *Proc. of the IEEE AFRICON*, 2009; 1-6.

[2] Muhammad R, Abdul R B, Kashif Z. Niching with sub-swarm based particle swarm optimization[C] // *Proc. of the International Conference on Computer Technology and Development*, 2009; 181-183.

[3] 叶文, 欧阳中辉, 朱爱红, 等. 求解多峰函数优化的生境克隆选择算法[J]. 系统工程与电子技术, 2010, 32(5): 1100-1104. (Ye W, Ouyang Z H, Zhu A H, et al. Niche clonal Selection Algorithm for Multi-Modal Function Optimization[J]. *Systems Engineering and Electronics*, 2010, 32(5): 1100-1104.)

[4] 薛文涛, 吴晓蓓, 徐志良. 用于多峰函数优化的免疫粒子群网络算法[J]. 系统工程与电子技术, 2009, 31(3): 705-709. (Xue W T, Wu X B, Xu Z L. Immune particle swarm network algorithm for multimodal function optimization[J]. *Systems Engineering and Electronics*, 2009, 31(3): 705-709.)

[5] 张梅凤, 邵诚. 多峰函数优化的生境人工鱼群算法[J]. 控制理论与应用, 2008, 25(4): 773-776. (Zhang M F, Shao B. Niche artificial fish swarm algorithm for multimodal function optimization[J]. *Control Theory & Application*, 2008, 25(4): 773-776.)

[6] Meng X Y, Huang S. A new method to design section area curve of ship based on niche ACO[C] // *Proc. of the 8th World Congress on Intelligent Control and Automation*, 2010; 3230-3235.

[7] Ling Q, Wu G, Yang Z Y, et al. Crowding clustering genetic algorithm for multimodal function optimization[J]. *Applied Soft Computing*, 2008, 8(1): 88-95.

[8] Li M Y, Hu J, Li L. Multipath planning based on neural network optimized with adaptive niche in unknown environment[C] // *Proc. of the International Conference on Intelligent Computation Technology and Automation*, 2010; 761-764.

[9] Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm[J]. *Applied Soft Computing*, 2008, 8(1): 687-697.

[10] Sareni B, Krahenbuhl L. Fitness sharing and niching methods revisited[J]. *IEEE Trans. on Evolutionary Computation*, 1998, 2(3): 97-106.

[11] 陆青, 梁昌勇, 杨善林, 等. 面向多模态函数优化的自适应小生境遗传算法[J]. 模式识别与人工智能, 2009, 22(1): 91-100. (Lu Q, Liang C Y, Zhang E Q, et al. A adaptive niche genetic algorithm for multimodal function optimization[J]. *Pattern Recognition and Artificial Intelligence*, 2009, 22(1): 91-100.)

[12] Thomsen R. Multimodal optimization using crowding-based differential evolution[C] // *Proc. of the Congress on Evolutionary Computation*, 2004; 1382-1389.

[13] Penev K, Littlefair G. Free search—a comparative analysis[J]. *Information Sciences*, 2005, 172(1): 173-193.

[14] 刘洪杰, 王秀峰. 多峰搜索的自适应遗传算法[J]. 控制理论与应用, 2004, 21(2): 302-310. (Liu H J, Wang X F. Adaptive genetic algorithm for multi-peak searching[J]. *Control Theory & Application*, 2004, 21(2): 302-310.)

[15] Lu Q, Liang C Y, Zhang E Q. A dynamic sharing scheme-based multimodal niche genetic algorithm[C] // *Proc. of the 7th World Congress on Intelligent Control and Automation*, 2008; 5333-5338.