

## 基于对立搜索和混沌变异的磷虾觅食优化算法

王磊<sup>a</sup>, 张汉鹏<sup>b</sup>, 张东宁<sup>a</sup>

(西南财经大学 a. 经济信息工程学院, b. 工商管理学院, 成都 610074)

**摘要:** 针对磷虾觅食算法存在容易陷入局部极值、收敛速度慢的问题, 提出一种新的改进算法. 首先, 给出启发式二次对立点的定义并证明其性能优势, 进而构造一种启发式二次对立搜索算子, 以加快算法的收敛速度, 提高全局探索能力; 然后, 采用分段线性混沌映射(PWLCM)混沌函数构造一种变尺度混沌变异算子, 以增强算法跳出局部极值的能力. 仿真实验表明, 所提出算法能有效避免陷入局部极值, 在收敛速度和寻优精度上得到大幅改善.

**关键词:** 磷虾觅食算法; 启发式二次对立点; 分段线性混沌映射混沌函数; 局部极值

中图分类号: TP18

文献标志码: A

## An improved krill herd algorithm based on oppositional searching and chaos mutation

WANG Lei<sup>a</sup>, ZHANG Han-peng<sup>b</sup>, ZHANG Dong-ning<sup>a</sup>

(a. School of Economics Information Engineering, b. School of Business Administration, Southwestern University of Finance & Economic, Chengdu 610074, China. Correspondent: WANG Lei, E-mail: wanglei.t@swufe.edu.cn)

**Abstract:** An improved algorithm based on oppositional searching and chaos mutation is proposed in order to deal with the deficiencies of the traditional krill herd optimization algorithm, including poor ability for avoiding local optimum and low convergence rate. The definition of the heuristic quasi-oppositional point is given, and its outstanding performance is proven. Then, a heuristic quasi-oppositional searching operator is designed for accelerating convergence rate and enhancing the global exploration ability of the algorithm. Meanwhile, the mutative scale chaos mutation operator based on the piecewise linear chaotic map(PWLCM) mapping function is constructed for boosting the ability of escaping local optimum. Simulation results on benchmark functions show that the proposed algorithm can avoid local optimum effectively, and achieves significant improvements in terms of convergence speed and accuracy.

**Keywords:** krill herd algorithm; heuristic quasi-oppositional point; PWLCM chaotic mapping function; local optimum

### 0 引言

最近, Gandomi等<sup>[1]</sup>在研究磷虾的觅食行为时, 发现磷虾个体的运动明显受到食物位置和虾群密度的影响, 从而形成稳定的虾群结构并不断朝食物位置移动. 他们采用Lagrangian模型来解释该行为, 进而提出一种新颖的仿生群体智能算法——磷虾觅食算法(KH).

KH算法同时考虑了磷虾个体的多种运动特性, 兼顾了全局探索能力与局部开采能力之间的平衡, 并具有控制参数少、易于实现等优点. 经仿真和实验测试, 其性能优于目前大多数群体智能算法<sup>[2-3]</sup>. 然而,

近期的研究表明, KH算法虽然具有优异的局部开采能力, 但全局探索能力不强, 特别在处理高维多峰函数优化问题时, 存在易陷入局部极值和收敛速度缓慢等不足. 对此, 文献[4]将DE算法的交叉和变异算子引入KH算法, 以增强算法的种群多样性和全局探索能力. 文献[5]利用混沌序列代替KH算法中的随机参数, 使得算法搜索行为具有一定混沌特性, 增强了全局探索能力. 文献[6]采用基于和声搜索的变异算子, 提高了算法的收敛速度和跳出局部极值的能力. 文献[7]提出一种SSC交叉算子, 通过贪婪地与当前最优解进行交叉, 加快了算法的收敛速度. 类似的研究还包括文献[8-9]. 上述工作在一定程度上克服了

收稿日期: 2014-06-26; 修回日期: 2015-01-14.

基金项目: 国家自然科学基金重大项目(91218301); 国家自然科学基金面上项目(71473201); 教育部人文社会科学研究一般项目(14XJC630010); 中央高校基本科研业务费专项资金项目(JBK130503, JBK150503).

作者简介: 王磊(1978-), 男, 副教授, 博士, 从事机器学习、模式识别等研究; 张汉鹏(1977-), 男, 副教授, 博士, 从事最优化理论、物流管理等研究.

KH算法的不足,但大都只针对算法的部分或单方面缺陷进行改进.因此,作为一种新型群体智能算法,进一步对KH的性能进行研究和完善十分必要.

本文提出一种基于对立搜索和混沌变异的改进磷虾觅食优化算法.它采用启发式二次对立搜索算子以提高算法的收敛速度和全局探索能力,并基于PWLCM混沌映射函数提出变尺度混沌变异算子,以增强算法跳出局部极值的能力,同时还利用搜索域动态收缩策略进一步提高算法的寻优效率.

## 1 基本的磷虾觅食算法

经研究发现,磷虾在觅食过程中通常形成一定的群体结构,食物吸引和虾群密度是影响该结构的两大主要因素<sup>[10]</sup>.磷虾个体的移动位置可用如下的Lagrangian模型描述:

$$\frac{d\mathbf{X}_i}{dt} = \mathbf{N}_i + \mathbf{F}_i + \mathbf{D}_i. \quad (1)$$

其中:  $\mathbf{X}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n}\}$  代表第  $i$  个磷虾个体的位置向量,其第  $k$  维变量  $x_{i,k}$  的搜索域范围为  $[l_k, u_k]$ ;  $\mathbf{N}_i$ 、 $\mathbf{F}_i$ 、 $\mathbf{D}_i$  分别代表受虾群运动影响、觅食活动和随机物理扩散而造成的 3 种运动向量.

算法中,运动向量  $\mathbf{N}_i$  由周围磷虾运动向量、当前最优个体的运动向量以及自身运动向量共同确定,即

$$\mathbf{N}_i = N^{\max}(\alpha_i^{\text{local}} + \alpha_i^{\text{best}}) + \omega^n \mathbf{N}_i^{\text{old}}. \quad (2)$$

其中:  $N^{\max}$  是最大移动步长,  $\omega^n$  是惯性权重,  $\alpha_i^{\text{local}}$  代表其周围邻居的运动向量的矢量和,  $\alpha_i^{\text{best}}$  是当前最优个体的运动向量.

觅食运动向量  $\mathbf{F}_i$  受当前估计的食物位置和上一次觅食活动及位置的影响,即

$$\mathbf{F}_i = \mathbf{V}^f \beta_i + \omega^f \mathbf{F}_i^{\text{old}}. \quad (3)$$

其中:  $\mathbf{V}^f$  是最大觅食速度向量,  $\omega^f$  是惯性权重,  $\beta_i = \beta_i^{\text{food}} + \beta_i^{\text{hisbest}}$  是由当前估计的食物位置和个体历史最优觅食位置共同计算出的吸引度值,  $\mathbf{F}_i^{\text{old}}$  是前一次的觅食运动向量.

物理扩散运动向量  $\mathbf{D}_i$  表示磷虾个体的随机搜索行为,可由下式模拟:

$$\mathbf{D}_i = D^{\max} \left(1 - \frac{G}{G^{\max}}\right) \delta. \quad (4)$$

其中:  $D^{\max}$  是最大扩散速度,  $\delta$  是随机扩散方向向量,  $G$  和  $G^{\max}$  是当前迭代次数和算法最大迭代次数.因此, KH 算法每轮迭代的位置更新公式为

$$\mathbf{X}_i(t + \Delta t) = \mathbf{X}_i(t) + \Delta t \frac{d\mathbf{X}_i}{dt}, \quad (5)$$

其中  $\Delta t$  是具体应用相关的时间间隔,即

$$\Delta t = \frac{C^t}{n} \|\mathbf{U} - \mathbf{L}\|_1. \quad (6)$$

时间常量  $C^t \in (0, 2]$  控制磷虾个体的运动幅度,搜索域上限  $\mathbf{U}$  和下限  $\mathbf{L}$  分别是  $u_k$  和  $l_k$  的向量形式.

由式(5)可知,磷虾个体的运动由周围邻居、最优个体、食物位置、自身位置等多个因素共同确定,因而KH算法的局部开采能力优异.但由于缺乏强的全局搜索算子(如变异),算法的全局探索能力不强,在一些多峰优化问题中容易陷入局部极值<sup>[4-9]</sup>.同时,每次迭代过程中均由多个因素共同决定个体的运动方向,难以迅速靠近全局最优值,收敛速度较慢<sup>[2]</sup>.

## 2 改进 KH 算法

### 2.1 启发式二次对立搜索算子

Tizhoosh<sup>[11]</sup>提出了对立点的概念,并从理论上证明利用对立搜索代替群体智能算法中的随机搜索,能够明显提高算法的收敛速度和改善全局探索能力.在此基础上,文献[12-13]分别提出了寻优效率更优的二次对立点和二次反射对立点,并应用于DE、BBO等算法中,它们的描述如图1和下述定义所示.

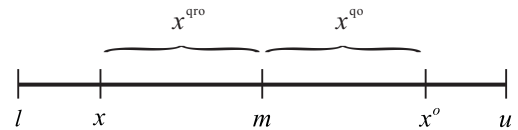


图 1 一维空间中 3 种对立点的示意图

**定义 1** (对立点)<sup>[11]</sup> 设  $x \in [l, u]$  是任意实数,它的对立点定义为

$$x^o = l + u - x. \quad (7)$$

**定义 2** (二次对立点)<sup>[12]</sup> 设  $x \in [l, u]$  是任意实数,  $x^o$  是其对立点,它的二次对立点定义为

$$x^{qo} = \text{rand}(m, x^o). \quad (8)$$

其中:  $m = (l + u)/2$  是区间  $[m, x^o]$  的中心点,  $\text{rand}(m, x^o)$  是区间内服从均匀分布的随机数.

**定义 3** (二次反射对立点)<sup>[13]</sup> 设  $x \in [l, u]$  是任意实数,它的二次反射对立点定义为

$$x^{qro} = \text{rand}(x, m). \quad (9)$$

分析可知,上述对立点在确定搜索位置时均未利用最优解的分布信息来提高搜索效率.例如:图1中若最优解靠近当前解  $x$ ,则在区间  $[x, m]$  搜索效率更高;若最优解靠近对立点  $x^o$ ,则在区间  $[m, x^o]$  搜索效率更高.因此,本文提出一种启发式二次对立点,它首先估计最优解的位置,然后作为先验信息启发式地确定搜索区间,从而提高搜索效率.其定义如下.

**定义 4** (启发式二次对立点) 设  $x \in [l, u]$  是任意实数,  $x^o$  是其对立点,则启发式二次对立点定义为

$$x^{\text{hqo}} = \begin{cases} \text{rand}(x, m), & d(x, x^*) \leq d(x^o, x^*); \\ \text{rand}(m, x^o), & d(x, x^*) > d(x^o, x^*). \end{cases} \quad (10)$$

其中距离函数  $d(\cdot, \cdot)$  用于估计最优解  $x^*$  与  $x$  (或  $x^o$ ) 之间的距离.

由于最优解的准确位置未知,在KH算法中可利

用种群中当前最优个体的位置  $x^{\text{best}}$  来近似  $x^*$ . 这样, 将一维空间中距离函数定义为  $d(x, x^*) \approx |x - x^{\text{best}}|$  和  $d(x^o, x^*) \approx |x^o - x^{\text{best}}|$ .

下面的定理1将表明采用启发式二次对立点进行搜索, 能够比二次对立点和二次反射对立点更快地搜索到最优解位置.

**定理1** 设  $x$  是区间  $[l, u]$  内的任意实数,  $x^{\text{qo}}$ 、 $x^{\text{qro}}$ 、 $x^{\text{hqo}}$  分别是其二次对立点、二次反射对立点和启发式二次对立点, 则有

$$\Pr[|x^{\text{hqo}} - x^*| < |x^{\text{qo}} - x^*|] > \frac{1}{2},$$

$$\Pr[|x^{\text{hqo}} - x^*| < |x^{\text{qro}} - x^*|] > \frac{1}{2}.$$

**证明** 图1中最优解  $x^*$  有4种可能分布位置: 1)  $x^* \in [l, x]$ ; 2)  $x^* \in [x, m]$ ; 3)  $x^* \in [m, x^o]$ ; 4)  $x^* \in [x^o, u]$ . 其中  $x$  和  $x^o$  关于  $m$  对称.

在情况1)和情况2)下,  $x^{\text{hqo}}$  在区间  $[x, m]$  内取值, 显然有

$$\Pr[|x^{\text{hqo}} - x^*| < |x^{\text{qo}} - x^*|] = \frac{1}{2}.$$

在情况3)下,  $x^{\text{hqo}}$  在区间  $[m, x^o]$  内取值并服从均匀分布, 可以证明

$$\Pr[|x^{\text{hqo}} - x^*| < |x^{\text{qo}} - x^*|] =$$

$$\Pr[x^{\text{hqo}} - x^* < x^* - x^{\text{qo}} | x^{\text{hqo}} \geq x^*] \times$$

$$\Pr[x^{\text{hqo}} \geq x^*] + \Pr[x^{\text{hqo}} > x^{\text{qo}} | x^{\text{hqo}} < x^*] \times$$

$$\Pr[x^{\text{hqo}} < x^*] > \frac{1}{2}.$$

在情况4)下,  $x^{\text{hqo}}$  在区间  $[m, x^o]$  内取值, 显然有

$$\Pr[|x^{\text{hqo}} - x^*| < |x^{\text{qo}} - x^*|] = 1.$$

因此, 下述不等式成立:

$$\Pr[|x^{\text{hqo}} - x^*| < |x^{\text{qo}} - x^*|] > \frac{1}{2}.$$

类似地, 可以证明

$$\Pr[|x^{\text{hqo}} - x^*| < |x^{\text{qro}} - x^*|] > \frac{1}{2}. \quad \square$$

进一步, 可以把一维空间中的启发式二次对立点扩展到任意维空间, 定义如下.

**定义5** ( $n$ 维启发式二次对立点) 设  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  是  $n$ 维空间的向量, 且  $x_k \in [l_k, u_k], k = 1, 2, \dots, n$ , 则  $\mathbf{X}^{\text{hqo}} = (x_1^{\text{hqo}}, x_2^{\text{hqo}}, \dots, x_n^{\text{hqo}})$  定义为它的  $n$ 维启发式二次对立点, 其中  $x_k^{\text{hqo}}$  按式(10)计算.

基于定义5, 本文设计如下的启发式二次对立搜索算子.

**定义6** (启发式二次对立搜索算子) 对于  $n$ 维向量  $\mathbf{X}$  执行如下的操作: 1) 利用定义5计算  $\mathbf{X}$  的启发式二次对立点  $\mathbf{X}^{\text{hqo}}$ ; 2) 使用适应度评价函数  $f(\cdot)$  评价  $\mathbf{X}^{\text{hqo}}$  的适应度值; 3) 若  $f(\mathbf{X}^{\text{hqo}}) < f(\mathbf{X})$ , 则替换

$$\mathbf{X} = \mathbf{X}^{\text{hqo}}.$$

显然, 相比于基于其他对立点的搜索方法, 本文提出的搜索算子在  $n$ 维空间更加高效地搜索最优解, 不仅能获得快速的收敛速度, 而且有利于增强种群的多样性和全局探索能力.

## 2.2 变尺度混沌变异算子

混沌是非线性系统特有的非周期运动现象, 能在一定范围内按自身规律不重复地遍历所有状态, 因而成为优化问题中避免陷入局部极值的一种有效方法<sup>[5,14]</sup>. 本文将采用轨道遍历性和参数不敏感性优异的PWLCM映射函数产生混沌序列, 构造一种变尺度混沌变异算子, 以增强KH算法跳出局部极值的能力.

PWLCM混沌映射函数的定义如下<sup>[15]</sup>:

$$z_{s+1} = \begin{cases} z_s/\rho, & z_s \in (0, \rho); \\ (1 - z_s)/(1 - \rho), & z_s \in [\rho, 1]. \end{cases} \quad (11)$$

其中参数  $\rho$  可以在  $(0, 1)$  内取值, 本文取  $\rho = 0.5$ . 显然, 混沌变量满足  $z_s \in (0, 1)$ .

考虑到迭代前期算法应具有较强的全局探索能力和跳出局部极值的能力, 迭代后期算法应具有较强的局部开采和精细搜索能力, 本文设计一种变尺度混沌变异算子, 其定义如下.

**定义7** (变尺度混沌变异算子) 设置混沌搜索次数  $s = 1$ , 对于  $n$ 维向量  $\mathbf{X}$ , 当  $s < S^{\text{max}}$  时重复执行如下的操作: 1) 将向量  $\mathbf{X}$  映射到区间  $(0, 1)$ , 即  $\mathbf{Z}_s = (\mathbf{X} - \mathbf{L})/(\mathbf{U} - \mathbf{L})$ ; 2) 利用式(11)对  $\mathbf{Z}_s$  的每一维变量进行混沌搜索, 得到  $\mathbf{Z}_{s+1}$ ; 3) 在原始向量空间中还原出混沌搜索位置  $\mathbf{X}^{\text{chaos}} = \mathbf{Z}_{s+1} \cdot (\mathbf{U} - \mathbf{L}) + \mathbf{L}$ ; 4) 计算新搜索位置  $\mathbf{X}' = (1 - \alpha_G)\mathbf{X} + \alpha_G\mathbf{X}^{\text{chaos}}$  及其适应度值; 5) 如果  $f(\mathbf{X}') < f(\mathbf{X})$ , 则执行变尺度混沌变异, 即重置  $\mathbf{X} = \mathbf{X}'$ ; 6) 更新  $s = s + 1$ .

定义7中,  $S^{\text{max}}$  是最大混沌搜索次数;  $\alpha_G$  是混沌参数, 它控制了混沌变异的尺度, 可按下式取值:

$$\alpha_G = 1 - \left(\frac{G - 1}{G^{\text{max}}}\right)^c. \quad (12)$$

参数  $c$  控制了尺度收缩的速度, 本文取  $c = 2$ .

在本文提出的改进算法中, 如果发现种群的当前最优个体  $\mathbf{X}^{\text{best}}$  连续  $w$  次迭代后仍未发生变化 ( $w$  是固定停滞参数), 则认为算法处于“收敛停滞”状态, 可能陷入局部最优. 此时, 算法执行变尺度混沌变异算子, 以增强跳出局部极值的能力.

## 2.3 改进算法——OCKH

为增强基本KH算法的全局探索能力和跳出局部极值的能力, 以及提高其收敛速度, 本文利用前面提出的两种算子对其进行改进, 获得一种快速高效的改进算法——基于对立搜索和混沌变异的磷虾觅食算法(OCKH), 其主要步骤如下.

**算法 1** 初始化: 种群规模 NP, 最大迭代次数  $G^{\max}$ , 最大混沌搜索次数  $S^{\max}$  及停滞参数  $w$ .

Step 1: 随机初始化 NP 个磷虾个体, 产生初始种群  $\{X_i\}_{i=1}^{NP}$ , 并令  $G = 1$ .

Step 2: 评价适应度值并确定当前最优个体  $X^{\text{best}}$ .

Step 3: 按式 (2)~(4) 计算所有个体的运动向量, 并按式 (5) 更新其位置.

Step 4: 重新评价适应度值及当前最优个体  $X^{\text{best}}$ .

Step 5: 每个个体执行定义 6 中的对立搜索算子.

Step 6: 若最优个体的位置  $X^{\text{best}}$  连续  $w$  次迭代未发生改变, 则对其执行定义 7 中的变异算子.

Step 7: 按下式动态调整当前迭代的搜索域范围  $[L^G, U^G]$ :

$$\begin{cases} l_k^G = \min(\{x_{i,k} \mid i = 1, 2, \dots, NP\}), \\ u_k^G = \max(\{x_{i,k} \mid i = 1, 2, \dots, NP\}). \end{cases} \quad (13)$$

其中:  $l_k^G$ 、 $u_k^G$  分别是  $L^G$ 、 $U^G$  的第  $k$  维变量.

Step 8: 若  $G < G^{\max}$ , 则令  $G = G + 1$ , 转 Step 2; 否则, 输出最优解  $X^* = X^{\text{best}}$ , 算法结束.

在算法中, 为了进一步提高算法的效率, 参照文献 [13, 17] 类似的方法, 本文采用式 (13) 所示的搜索域动态缩放策略以加快算法寻优速度.

此外, 算法的其他参数依据磷虾的 Lagrangian 行为模型经验地进行取值 (参见文献 [10] 的表 1), 包括: 最大移动步长  $N^{\max}$ , 最大觅食速度  $V^f$ , 最大扩散速度  $D^{\max}$ , 惯性权重  $\omega^n$  和  $\omega^f$ .

假设评价磷虾个体适应度值 1 次的时间复杂度为  $O(f)$ , 比较和算术运算的时间复杂度均为  $O(1)$ , 则 OCKH 算法执行 1 次迭代的时间复杂度为  $O(NP \cdot f) + O(NP)$ , 这与基本 KH 算法的开销在同一量级上. 其中: Step 4、Step 5、Step 7 的计算开销分别为  $O(NP \cdot f) + O(NP)$ 、 $O(NP \cdot f) + O(NP)$  和  $O(NP)$ , Step 6 所需的开销上限为  $O(S^{\max} \cdot f) + O(S^{\max})$ . 考虑到达到相同

收敛精度时 OCKH 算法所需的迭代次数更少 (参见表 2 的实验结果), 故它具有更快的寻优速度.

### 3 仿真实验及分析

为了检验本文提出的 OCKH 算法的性能, 实验选取 9 个标准测试函数进行仿真测试, 并与基本 KH 算法<sup>[1]</sup>、采用 Singer 混沌函数调整参数的 CKH 算法<sup>[5]</sup>以及采用 SSC 交叉算子的 SKH 算法<sup>[7]</sup>进行性能对比.

#### 3.1 标准测试函数及算法参数设置

实验采用的标准测试函数为: 1) 单峰连续优化函数  $f_1 \sim f_3$ , 依次是 Sphere、Rosenbrock 和 Schwefel 1.2, 它们常用于测试算法的收敛速度; 2) 多峰连续优化函数  $f_4 \sim f_9$ , 依次是 Ackley、Fletcher、Penalty#1、Penalty#2、Rastrigin 和 Schwefel 2.26, 它们存在许多局部极值, 且数量随维度指数增长, 常用于测试算法的逃离局部极值和全局探索能力. 函数的数学定义、搜索域范围按文献 [17] 获得, 全局最优值均为 0.

实验参数设置为: 1) 对于 4 种算法的共同参数, 依据文献 [10] 的模型进行如下设置: 最大移动步长  $N^{\max} = 0.01$ , 最大觅食速度  $V^f = 0.02$ , 最大扩散速度  $D^{\max} = 0.005$ , 时间常量  $C^t = 0.5$ ; 2) 对于惯性权重  $\omega^n$  和  $\omega^f$ , CKH 算法采用 Singer 混沌函数在 (0.1, 0.9) 内遍历取值, 其他算法则按迭代次数从 0.9 线性变化到 0.1; 3) 对于 SKH 算法, SSC 交叉算子采用单点交叉操作, 交叉概率为  $p_c = 1$ ; 4) 对于本文的 OCKH 算法, 经实验比选, 将最大混沌搜索次数设置为  $S^{\max} = 20$ , 停滞参数设置为  $w = 10$ .

此外, 种群个体采用实数编码, 种群规模设置为  $NP = 30$ , 测试函数的维度统一设置为 50.

#### 3.2 实验结果及分析

首先, 固定算法的最大迭代次数为  $G^{\max} = 2000$ , 并在每个测试函数上独立运行 30 次. 表 1 统计了各算法的平均最优适应度值 (MBF) 和标准差 (SD) 的结果.

表 1 不同算法的收敛精度及其标准差的比较 (固定 2000 次迭代)

算法	指标	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$
KH	MBF	2.48e-6	1.74e+1	2.64e+4	1.73e+0	1.41e+5	5.02e-1	6.44e+0	6.00e-3	4.86e+0
	SD	7.17e-7	6.93e+0	5.77e+3	6.42e-1	4.31e+4	2.06e-1	1.75e+0	1.32e-3	1.34e+0
CKH	MBF	7.84e-9	7.93e+0	6.02e-1	4.72e-3	9.58e+3	6.15e-2	8.51e-1	2.77e-6	2.99e-1
	SD	9.08e-9	5.20e+0	9.80e-1	5.33e-3	7.09e+3	8.01e-2	6.37e-1	3.65e-6	3.74e-1
SKH	MBF	0	2.18e+0	4.41e-10	5.85e-9	7.18e+2	1.01e-2	2.40e-2	3.85e-10	9.93e-1
	SD	0	8.82e-1	1.70e-10	2.90e-9	2.66e+2	3.23e-3	1.18e-2	1.72e-10	2.17e-1
OCKH	MBF	0	3.81e-3	1.23e-19	4.81e-14	4.48e+0	2.08e-8	2.91e-5	0	5.93e-3
	SD	0	1.17e-3	5.83e-20	2.09e-14	2.55e+0	6.85e-9	4.93e-6	0	2.18e-3

从表1的结果可知, 固定迭代次数情况下, 本文提出的OCKH算法在所有测试函数上均获得了最优的收敛精度. 对于形式简单的单峰函数 $f_1$ , OCKH算法能收敛到全局最优解0. 对于最优解附近陡峭而极难收敛的复杂单峰函数 $f_2$ , OCKH在2000次迭代后的收敛精度为 $3.81 \times 10^{-3}$ , 比其他算法至少提高3个数量级. 对于相对容易收敛的多峰函数 $f_4$ 、 $f_6$ 、 $f_8$ , OCKH的收敛精度均在 $10^{-8}$ 以上. 对于存在大量局部极值的复杂多峰函数 $f_5$ 、 $f_7$ 、 $f_9$ , OCKH仍能够收敛到最优解附近, 收敛精度显著优于另外3种算法. 此外, KH算法的收敛精度在所有函数上最差, SKH算法获得的收敛精度优于CKH算法, 但在除Sphere函数以外的8个函数上明显劣于本文算法.

其次, 为了定量比较算法的收敛速度, 实验度量了在固定收敛精度情况下, 不同算法所需的(适应度)函数评价次数. 实验在每个函数上独立运行30次, 表2给出了不同算法的平均函数评价次数, 其中, 算法运行的最大函数评价次数限定为 $1 \times 10^5$ .

表2 固定收敛精度时算法的函数评价次数均值

函数	固定收敛精度	KH	CKH	SKH	OCKH
$f_1$	$10^{-10}$	21990	3127	181	42
$f_2$	$10^{-2}$	—	75669	47411	2306
$f_3$	$10^{-8}$	—	19482	839	104
$f_4$	$10^{-8}$	—	20064	1327	152
$f_5$	$10^0$	—	—	91552	7552
$f_6$	$10^{-6}$	—	67220	12822	1468
$f_7$	$10^{-4}$	—	72393	10566	1736
$f_8$	$10^{-10}$	896017	14330	3519	84
$f_9$	$10^{-2}$	—	55891	11892	670

从表2的结果可知, OCKH算法达到固定收敛精度所需的函数评价次数远低于其他算法, 因而具有最快的收敛速度. 例如, 在Rastrigin多峰函数上, 只需要84次函数评价就可以达到 $10^{-10}$ 量级的收敛精度. 而基本KH算法的收敛速度最慢, 在7个函数上无法达到固定收敛精度. SKH算法所需的函数评价次数比KH和CKH大幅减少, 但仍显著多于OCKH算法.

此外, 为了更加直观地解释本文算法的收敛速度及寻优性能优势, 图2~图5给出了算法的最优适应度值随迭代次数(固定为2000次)的进化曲线. 限于篇幅, 这里只给出在4个函数上的实验结果.

观察图2~图5可知, OCKH算法在迭代前期的收敛速度远快于其他算法. 例如, 在Ackley上只需要314次迭代便可获得 $10^{-12}$ 的寻优精度. 同时, 在迭代中后期OCKH算法仍保持良好的探索和寻优能力, 进化曲线具有明显的锯齿状下降趋势. 分析原因,

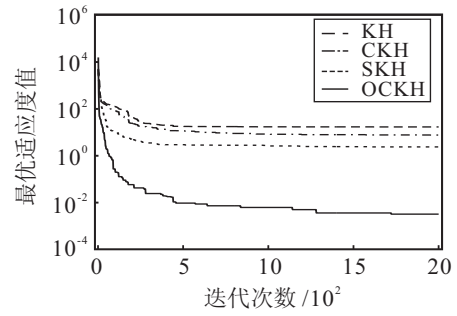


图2 各算法在 $f_2$ 函数上的进化曲线

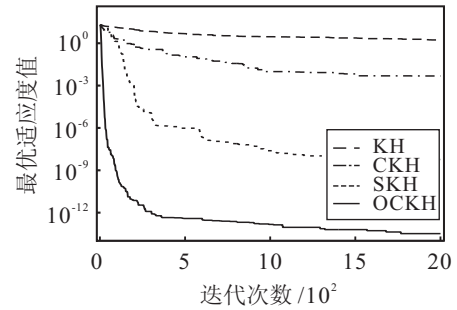


图3 各算法在 $f_4$ 函数上的进化曲线

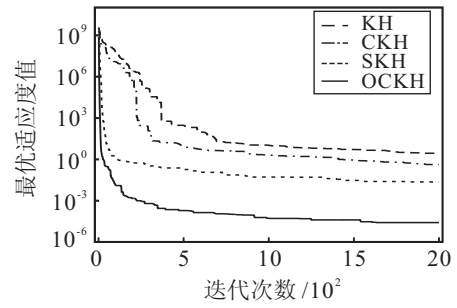


图4 各算法在 $f_7$ 函数上的进化曲线

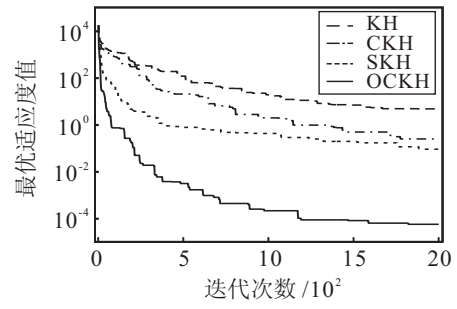


图5 各算法在 $f_9$ 函数上的进化曲线

OCKH算法采用的启发式二次对立搜索算子具有优异的寻优效率, 使得算法迅速地搜索到最优值附近, 因而收敛速度非常快. 同时, 在迭代后期, 算法的变尺度混沌变异算子的作用不断增强, 使得算法具有很强的跳出局部极值的能力, 能获得非常高的收敛精度.

与之相比较, KH算法的进化曲线下降缓慢, 并在 $f_2$ 和 $f_4$ 函数上明显陷入了局部极值, 收敛精度最低. 此现象验证了其固有的全局探索能力不强、收敛速度慢的缺陷. CKH算法简单地使用混沌序替代KH的随机参数, 虽然使得寻优过程表现出一定的混沌特点, 但其寻优机制并未发生根本变化, 仍难以克服KH

的固有缺陷,因而实验中CKH算法的收敛速度和精度改善不明显. SKH算法利用SSC交叉算子使得种群快速向最优解靠近,因而算法前期的收敛速度很快;但由于缺乏高效的避免局部极值的机制,算法后期收敛速度趋缓,难以快速收敛到全局最优.

综上,OCKH算法在收敛速度和收敛精度方面显著优于KH及其改进的CKH和SKH算法,是一种快速有效的全局优化算法.

#### 4 结 论

为了改善磷虾觅食优化算法容易陷入局部极值、收敛速度慢的不足,本文首先提出了一种启发式二次对立点并证明其搜索效率优势,用于提高KH算法的收敛速度和全局探索能力;然后,采用PWLCM混沌映射函数设计了一种变尺度混沌变异算子,增强了KH算法跳出局部极值的能力. 基于9个标准测试函数的实验结果表明,本文提出的OCKH算法在全局寻优精度和收敛速度方面均显著优于基本KH算法及最新的CKH算法和SKH算法,具有良好的工程应用前景.

#### 参考文献(References)

- [1] Gandomi A H, Alavi A H. Krill herd: A new bio-inspired optimization algorithm[J]. *Communications in Nonlinear Science and Numerical Simulation*, 2012, 17(12): 4831-4845.
- [2] Singh G P, Singh A. Comparative study of krill herd, firefly and cuckoo search algorithms for unimodal and multimodal optimization[J]. *Int J of Intelligent Systems and Applications*, 2014, 6(3): 35-49.
- [3] Gandomi A H, Talatahari S, Tadbiri F, et al. Krill herd algorithm for optimum design of truss structures[J]. *Int J of Bio-Inspired Computation*, 2013, 5(5): 281-288.
- [4] Mandal B, Roy P K, Mandal S. Economic load dispatch using krill herd algorithm[J]. *Int J of Electrical Power and Energy Systems*, 2014, 57(5): 1-10.
- [5] Wang G G, Guo L H, Gandomi A H, et al. Chaotic krill herd algorithm[J]. *Information Sciences*, 2014, 274(8): 17-34.
- [6] Wang G G, Guo L H, Wang H Q, et al. Incorporating mutation scheme into krill herd algorithm for global numerical optimization[J]. *Neural Computing and Applications*, 2014, 24(3/4/5): 853-871.
- [7] Wang G G, Gandomi A H, Alavi A H. Stud krill herd algorithm[J]. *Neurocomputing*, 2014, 128(5): 363-370.
- [8] Wang G G, Gandomi A H, Alavi A H. An effective krill herd algorithm with migration operator in biogeography-based optimization[J]. *Applied Mathematical Modelling*, 2014, 38(9/10): 2454-2462.
- [9] Li J P, Tang Y G, Hua C C, et al. An improved krill herd algorithm: Krill herd with linear decreasing step[J]. *Applied Mathematics and Computation*, 2014, 234(5): 356-367.
- [10] Hofmann E E, Haskell A G, Klinck J M, et al. Lagrangian modelling studies of antarctic krill (*Euphasia superba*) swarm formation[J]. *ICES J of Marine Science*, 2004, 61(4): 617-631.
- [11] Tizhoosh H R. Opposition-based learning: A new scheme for machine intelligence[C]. *Proc of Int Conf on Computational Intelligence for Modelling, Control and Automation*. Vienna: IEEE, 2005: 695-701.
- [12] Rahnamayan S, Tizhoosh H R, Salama M M. Quasi-oppositional differential evolution[C]. *Proc of IEEE Congress on Evolutionary Computation*. Singapore: IEEE, 2007: 2229-2236.
- [13] Ergezer M, Simon D, Du D W. Oppositional biogeography-based optimization[C]. *Proc of IEEE Int Conf on System, Man and Cybernetics*. San Antonio: IEEE, 2009: 1009-1014.
- [14] 吴定海, 张培林, 李胜, 等. 基于混沌变异的自适应双粒子群优化[J]. *控制与决策*, 2011, 26(7): 1083-1086. (Wu D H, Zhang P L, Li S, et al. Adaptive double particle swarms optimization algorithm based on chaotic mutation[J]. *Control and Decision*, 2011, 26(7): 1083-1086.)
- [15] Baranovsky A, Daems D. Design of one-dimensional chaotic maps with prescribed statistical properties[J]. *Int J of Bifurcation and Chaos*, 1995, 5(6): 1585-1598.
- [16] Wang H, Wu Z J, Rahnamayan S, et al. Enhancing particle swarm optimization using generalized opposition-based learning[J]. *Information Sciences*, 2011, 181(20): 4699-4714.
- [17] Jamil M, Yan X S. A literature survey of benchmark functions for global optimization problems[J]. *Int J of Mathematical Modelling and Numerical Optimization*, 2013, 4(2): 150-194.

(责任编辑: 李君玲)