

# 基于余弦相似度反向策略的自然计算方法

孙小琳, 季伟东, 王 旭

哈尔滨师范大学大学计算机科学与信息工程学院, 黑龙江 哈尔滨 150025

基金项目: 国家自然科学基金(31971015); 黑龙江省自然科学基金(LH2021F037); 哈尔滨市科技局科技创新人才研究专项项目(2017RAQXJ050); 哈尔滨师范大学计算机科学与信息工程学院科研项目(JKYKYY202001)

通信作者: 季伟东, kingjwd@126.com 收稿/录用/修回: 2021-07-21/2021-09-29/2022-02-25

## 摘要

现有的基于反向策略的优化算法大多根据初始种群适应度值大小进行反向择优, 没有充分考虑迭代过程中的反向且存在收敛速度慢的问题。针对此问题, 提出一种基于余弦相似度反向策略的快速收敛自然计算方法, 通过计算每个粒子与区域中心粒子的余弦相似度, 将粒子划分为相似子群与非相似子群, 对非相似子群按照相似程度进行加权反向, 进而加快收敛速度, 同时引入柯西扰动提高种群多样性。将该策略应用到三种不同的自然计算方法中, 对收敛性进行分析, 并采用12个经典测试函数验证其性能, 对实验数据进行非参数检验。分析结果表明, 应用余弦相似度反向策略的方法在大多数测试函数上表现优异, 说明提出的方法具有很好的普适性和有效性。

## 关键词

自然计算  
快速收敛  
余弦相似度  
加权反向  
中图法分类号: TP301  
文献标识码: A

## Natural Computation Method Based on Cosine Similarity Opposition Strategy

SUN Xiaolin, JI Weidong, WANG Xu

College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China

## Abstract

Most of the existing optimization algorithms based on the opposite learning strategy are under the initial population fitness value. The problem of opposition and slow convergence is not fully considered. Thus, a fast convergent natural computation method based on a cosine similarity opposition strategy is proposed to solve the aforementioned problem. The particles are divided into similar and nonsimilar subgroups by calculating the cosine similarity between each particle and the particle in the regional center. The nonsimilar subgroups are weighted opposition according to the similarity degree, thus the convergence speed is accelerated. The Cauchy disturbance is then introduced to improve population diversity. The strategy is applied to three different natural computing methods, and 12 classical test functions are used to analyze the convergence and verify its performance. Finally, nonparametric tests are performed on the experimental data. Experiments show that the method performs well in most test functions and has good universality and effectiveness.

## Keywords

natural computation;  
fast convergence;  
cosine similarity;  
weighted opposition

## 0 引言

自然计算(natural computation)通常是指那些试

图从信息处理的角度来理解世界, 受自然启发而形成的智能算法<sup>[1]</sup>。自然计算方法能够借鉴自然界的功能与作用机理, 抽象出计算模型, 如进化计

算、生物分子计算、量子计算、模糊计算和情感计算等<sup>[2]</sup>。每种自然计算方法都对应各自实际的启发源<sup>[3]</sup>, 如植物生长算法对应植物的生长过程, 灰狼算法、鲸鱼算法对应动物捕食行为等。自然计算方法具有自适应、自组织、自学习的特点<sup>[4]</sup>, 目前已广泛应用于许多领域, 如模式识别与图像处理领域中基于遗传算法和自组织映射的模糊混合智能彩色图像分割方法<sup>[5]</sup>, 基于反向混沌变异的文本聚类特征选择算法<sup>[6]</sup>等; 身份识别领域中交通拥堵识别的模糊识别算法<sup>[7]</sup>, 基于多目标的空间搜索算法识别模糊推理系统<sup>[8]</sup>等; 医学领域中利用遗传算法的无符号基因排序<sup>[9]</sup>, 使用多目标进化策略的 DNA 核酸编码设计<sup>[10]</sup>等。除此以外还有基于灰狼算法的目标威胁评估<sup>[11]</sup>, 基于量子遗传的视觉目标跟踪<sup>[12]</sup>, 基于粒子群优化的室内定位<sup>[13]</sup>和基于萤火虫算法增强图像对比度<sup>[14]</sup>等应用。

虽然各种自然计算方法模拟的生物机理不同, 但它们都是基于种群的优化方法<sup>[15]</sup>。针对算法收敛速度慢、易陷入局部最优等问题, 为提高算法性能, 学者们提出了很多解决方案, 反向学习 (opposition-based learning, OBL)<sup>[16]</sup>就是其中之一。尽管反向学习的思想简单, 但在很多算法中都表现出了很好的性能。Rahnamayan 等<sup>[17]</sup>首次将 OBL 与差分进化 (differential evolution, DE) 算法进行结合。受此启发, Wang 等<sup>[18]</sup>将 OBL 应用于粒子群 (particle swarm optimization, PSO) 算法。而 Iqbal 等<sup>[19]</sup>则将 OBL 应用于遗传算法 (genetic algorithm, GA) 解决函数优化问题。Ewees 等<sup>[20]</sup>将 DE 与 OBL 相结合, 并将其应用于鲸优化算法进行多目标优化。Chen 等<sup>[21]</sup>使用 OBL 初始化种群并利用正余弦加速系数 (sine cosine acceleration coefficients, SCAC) 指导函数寻优。上述算法中, 利用 OBL 种群进行初始化时, 是在搜索空间中先计算解的反向值, 然后反向解与原解比较, 选择最优元素进行后续迭代寻优, 从而选出最优解。而 Feng 等<sup>[22]</sup>将 OBL 应用于果蝇算法 (fruit fly optimization algorithm, FOA) 中, 并设计了动态收缩参数策略调整反向学习强度。Dhargupta 等<sup>[23]</sup>将 OBL 与灰狼算法 (grey wolf optimization, GWO) 相结合, 每次迭代中使用皮尔逊相关系数选择反向候选解。Xing 等<sup>[24]</sup>提出了动态广义反向学习和即时策略有效优化狼群算法。除此之外, 孙辉等<sup>[25]</sup>提出了混合均值中心反向学习粒子群优化算法, 对混合均值中心进行 OBL, 从而扩大搜索区域。Rahnamayan 等<sup>[26]</sup>以整个种群重心作为参考点计算反向解, 提出了重心反向学习 (centroid opposition-based learning,

COBL)。受 COBL 的启发, 周凌云等<sup>[27]</sup>使用邻域重心作为参考点计算反向。邵鹏等<sup>[28]</sup>提出折射原理反向学习模型。罗强等<sup>[29]</sup>对全局最优粒子逐维进行重心反向学习。这些改进算法均产生了很好的效果, 在全局探索能力、收敛速度、计算精度上都有显著提高, 表明反向学习策略是一种十分有效的方法。但它们大多是针对某一特定算法在种群初始化时使用反向学习策略, 且依赖算法具体实现过程, 普适性不强。

受以上方法启发, 本文提出一种普适的基于余弦相似度反向学习 (cosine similarity opposition-based learning, CSOBL) 的快速收敛进化计算方法。首先, 在迭代过程中, 计算每个个体与区域中心个体的余弦相似度, 根据其值将种群划分为相似种群与非相似种群。其次, 根据非相似种群中每一个体与区域中心个体的相似程度对其进行加权反向计算, 在加快收敛速度的同时增加了种群多样性。最后对种群最优个体进行柯西扰动, 增加变异范围, 进一步提高种群多样性。将本策略运用到 3 种不同的自然计算方法中, 采用 12 个标准测试函数验证其性能, 并与目前主流反向优化算法进行对比, 用以验证基于余弦相似度反向策略的普适性和有效性。

## 1 余弦相似度与反向学习

### 1.1 余弦相似度

余弦相似度<sup>[30]</sup>是一种相关性度量方法, 它是通过测量两个向量夹角的余弦值来度量它们之间的相似性, 从而确定两个向量是否大致指向相同的方向。余弦相似度注重的是两个向量在方向上的差异, 因此, 对于高维空间, 余弦相似度能够很好地度量两个向量之间的相关性。其相关定义为<sup>[31]</sup>: 设  $D$  维空间中的两个可行解  $\mathbf{X}$ 、 $\mathbf{Y}$  对应的向量分别为  $\mathbf{X} = (x_1, x_2, \dots, x_D)$ ,  $\mathbf{Y} = (y_1, y_2, \dots, y_D)$ , 那么它们之间的余弦相似度由式 (1) 确定, 3 维空间中两个向量的余弦相似度图像如图 1 所示。

$$S = \cos \theta = \frac{\langle \mathbf{X}, \mathbf{Y} \rangle}{\|\mathbf{X}\| \|\mathbf{Y}\|} = \frac{\sum_{i=1}^D x_i y_i}{\sqrt{\sum_{i=1}^D x_i^2} \sqrt{\sum_{i=1}^D y_i^2}} \quad (1)$$

其中,  $\theta$  为向量  $\mathbf{X}$ 、 $\mathbf{Y}$  之间的夹角,  $x_i (i = 1, 2, \dots, D)$  为向量  $\mathbf{X}$  在第  $i$  维的属性值,  $y_i (i = 1, 2, \dots, D)$  为向量  $\mathbf{Y}$  在第  $i$  维的属性值,  $D$  为问题空间的维度。  $\cos \theta \in [-1, 1]$ , 余弦相似度越靠近 1 表明两个向量的夹角越接近 0, 即两个向量越相似。反之, 余弦相似度越靠近 -1, 表明两个向量越不相似。

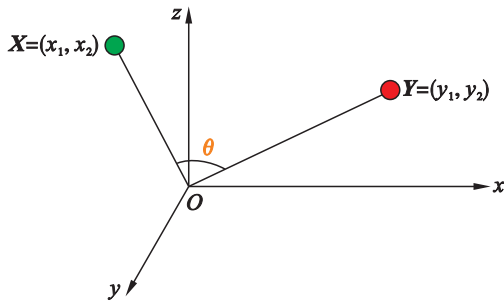


图1 3 维空间中两个向量的余弦相似度  
Fig.1 The cosine similarity of two vectors in the three-dimensional space

1.2 反向学习策略

如果某个体离最优解很远,那么它的反向个体可能具有更好的适合度值<sup>[16]</sup>。反向学习定义为:假设  $\mathbf{X} = (x_1, x_2, \dots, x_D)$  是  $D$  维空间中的某一可行解对应的向量,且  $x_j \in [a_j, b_j] (j=1, 2, \dots, D)$ , 则其对应的反向解向量  $\mathbf{X}^* = (x_1^*, x_2^*, \dots, x_D^*)$  由式(2)确定, 1 维空间中反向解的位置如图 2 所示。

$$x_j^* = a_j + b_j - x_j \quad (2)$$

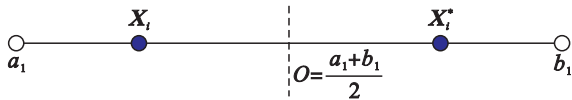


图2 1 维空间反向  
Fig.2 Opposition in the one-dimensional space

2 余弦相似度反向自然计算方法

2.1 余弦相似度反向策略

目前,数据规模庞大且大都具有多维特性<sup>[32]</sup>,因此,本算法采用适用于多维数据的余弦相似度反向来度量问题空间中某一个体与全部个体的平均位置的相似程度。相关定义为:设问题空间中个体  $\mathbf{X}_i$  对应的向量为  $(x_{i1}, x_{i2}, \dots, x_{iD})$ , 其中  $i=1, 2, \dots, N$ 。每次迭代过程中全部个体的平均位置对应的中值向量  $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_D)$  如式(3)所示:

$$\mu_j = \sum_{i=1}^N x_{ij} / N \quad (3)$$

其中,  $N$  表示问题空间中全部个体的数目,  $j=1, 2, \dots, D$ 。

个体  $\mathbf{X}_i$  的余弦相似度可以定义为此个体对应的向量  $\mathbf{X}_i$  与中值向量  $\boldsymbol{\mu}$  之间的余弦相似度,余弦相似度越接近 1,说明此个体与平均位置的相似程度越高,计算方法如式(4)所示。

$$S(\mathbf{X}_i) = \cos(\mathbf{X}_i, \boldsymbol{\mu}) = \frac{\sum_{j=1}^D (x_{ij}\mu_j)}{\sqrt{\sum_{j=1}^D x_{ij}^2} \sqrt{\sum_{j=1}^D \mu_j^2}} \quad (4)$$

其中,  $D$  为问题空间的维度,  $\mu_j$  为中值向量第  $j$  维的值,  $x_{ij}$  为个体  $\mathbf{X}_i$  第  $j$  维的值。

根据个体的余弦相似度,对个体进行相关性选择,相关性选择的分类阈值分别设置为  $-1/2, 0, 1/2$ 。前期实验表明,分类阈值为 0 时寻优结果最佳,故本文选取 0 为相关性选择分类阈值。若个体的余弦相似度系数小于 0,表明这些个体与中值向量的相似度很小,将这些个体归类为非相似种群,并根据余弦相似度系数对其进行加权反向。反向候选解  $\mathbf{X}_i^*$  的各维属性值按式(5)计算:

$$x_{ij}^* = (a_j + b_j - x_{ij}) |S(\mathbf{X}_i)| \quad (5)$$

其中,  $a_j$  为问题空间的下界,  $b_j$  为问题空间的上界,  $S(\mathbf{X}_i)$  为个体  $\mathbf{X}_i$  的余弦相似度。

在基于余弦相似度的反向过程中,使用了动态收缩策略调整反向学习权重,任一个体与中值个体的相似程度值越小,表示个体与中值个体越不相似,向量之间的夹角越大,相似程度值的绝对值越大,则进行反向加权收缩后,反向粒子的位置离中值粒子越近。由于余弦相似度是基于方向性的相似程度度量,所以非相似种群中的个体能够避免不必要的反向,快速向中值个体靠拢,同时保证了种群的多样性。为直观描述描述反向过程,本文仅画出处于非相似子群中的 3 个个体,它们反向后的具体位置如图 3 所示,其中,  $\boldsymbol{\mu}$  表示中值向量个体的位置,  $\mathbf{X}_i (i=1, 2, \dots, N)$  表示任一个体,  $\mathbf{X}_i^* (i=1, 2, \dots, N)$  表示余弦反向后的个体。

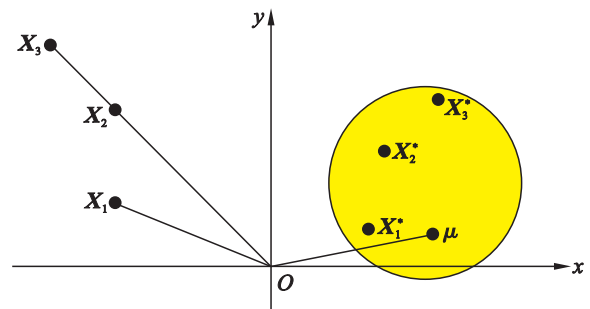


图3 非相似子群个体经余弦相似度反向学习后的位置  
Fig.3 Positions of individuals in non-similar subgroups after cosine similarity opposition-based learning

2.2 柯西扰动

当个体陷入局部最优时,由图 4 可知,柯西分布的两端延伸能够产生距中心较远的随机数,柯西算子能以更高的概率跳出局部最优,且变异后的个体与原个体具有较大差异。因此,为了使个体具有更强的搜索能力,本算法使用柯西变异机制<sup>[33]</sup>对全局最优个体位置进行变异扰动。

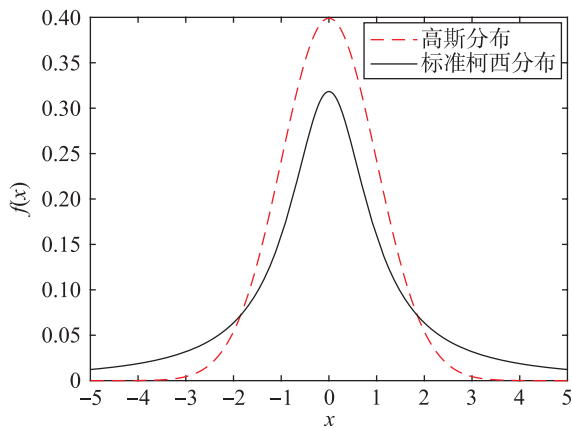


图4 高斯分布和标准柯西分布概率密度函数曲线  
Fig.4 Probability density function curves of Gaussian distribution and standard Cauchy distribution

标准柯西分布概率密度如下<sup>[34]</sup>:

$$f(x) = \frac{1}{\pi} \frac{1}{x^2 + 1}, \quad x \in (-\infty, +\infty) \quad (6)$$

本算法的变异概率随迭代次数增加而减小, 减少变异值以避免最优值的动荡, 从而更可能探寻新区域, 帮助提高种群的多样性, 使搜索能够跳出局部最优区域, 增大发现最优解的概率。对最优个体的柯西变异如下:

$$g_b^i = g_b^i + C \quad (7)$$

其中,  $g_b^i$  为第  $i$  代的全局最优个体,  $g_b^i$  为当前最优个体经柯西扰动后的新值,  $C$  为生成的柯西变异随机数。

### 2.3 余弦相似度反向策略

余弦相似度反向策略流程如图 5 所示。

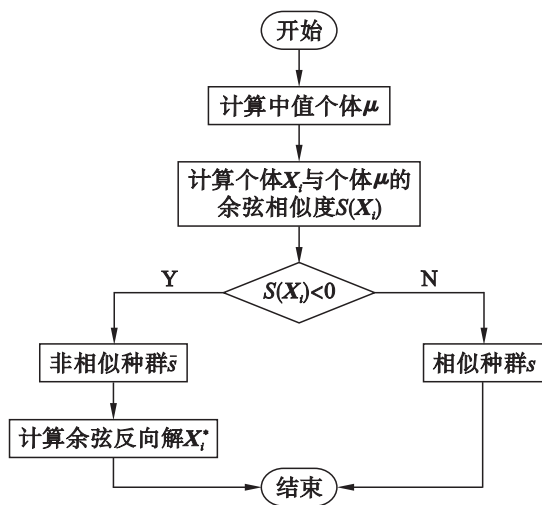


图5 余弦相似度反向策略流程图  
Fig.5 The flow chart of cosine similarity opposition-based strategy

基于余弦相似度反向策略的算法步骤见算法 1。

### 算法 1 基于余弦相似度反向的自然计算方法

输入: 相关参数;

输出: 最优解

1. 在给定区域内随机初始化种群 pop, 种群规模  $N$ , 维数  $D$  等参数
2. 计算当前种群 pop 中个体的适应度值 fitness;
3. 根据相应公式产生下一代种群 pop<sub>1</sub>;
4. 根据式(3)计算中值向量  $\mu$  位置, 式(4)计算个体  $j$  与  $\mu$  的相似程度;
5. 若相似程度  $< 0$ , 则将 pop<sub>1</sub> 划分为相似子群  $s$  与非相似子群  $\bar{s}$ ;
6. 根据式(5), 计算  $\bar{s}$  的余弦反向;
7. 根据式(6)以一定概率对全局最优柯西扰动;
8. 判断算法是否达到结束条件, 若结束, 则返回最优解; 否则, 转步骤 3。

## 3 算法时间复杂度与收敛性分析

### 3.1 算法时间复杂度分析

从算法 1 的描述中可以看出, 基于余弦相似度反向的计算方法主要包含以下部分: 初始化种群并计算适应度值、速度与位置更新操作、余弦相似度反向策略、柯西变异操作。设种群规模为  $N$ , 目标函数维数为  $D$ , 最大迭代次数为  $G_{\max}$ 。易知, 初始化种群并计算适应度值的时间复杂度为  $O(ND)$ , 速度与位置更新操作以及柯西变异的时间复杂度为  $O(G_{\max}ND)$ , 余弦相似度反向策略分为两个部分, 分别是计算余弦相似度和根据余弦相似度权重反向。其中计算余弦相似度需要的时间复杂度为  $O(G_{\max}((N+1)D)) = O(G_{\max}ND)$ ; 进行位置更新时条件判断以及权重反向的时间复杂度也是  $O(N)$  级别的时间复杂度。综上, 本算法的总的时间复杂度为  $O(N)$  级别。

### 3.2 算法的收敛性分析

**定义 1** 个体在可行解空间中的随机运动位置称为个体状态, 在迭代中的所有可能状态称为个体状态空间, 记为  $X^{[35]}$ 。

**定义 2** 对于问题  $\langle S, f \rangle$ , 采用某随机优化算法  $M$ , 对于个体  $x$ , 第  $t$  次迭代的结果为  $x_t$ , 则  $t+1$  代为  $x_{t+1} = M(x_t, \zeta)$ , 其中,  $S$  表示解空间,  $f$  表示适应度函数,  $\zeta$  表示曾搜索过的解。那么在 Lebesgue 空间中, 搜索下界:

$$\sigma = \inf\{t \mid v(x \in S \mid f(x) < t) > 0\} \quad (8)$$

其中,  $v(X)$  为集合  $X$  上的 Lebesgue 测度, 则最优解可定义为

$$R_{\xi} = \begin{cases} \{x \in S \mid f(x) < \sigma + \xi\}, & \sigma \text{ 有限} \\ \{x \in S \mid f(x) < -c\}, & \sigma = -\infty \end{cases} \quad (9)$$

其中,  $c$  表示充分大的正数,  $\xi > 0$ , 若算法  $M$  在迭代过程中找到了  $R_\xi$  中的某个解, 那么就认为算法  $M$  找到了可以接受的全局最优解<sup>[36]</sup>。

**条件 1**  $f(M(x, \zeta)) \leq f(x)$ , 且若  $\zeta \in S$ ,  $f(M(x, \zeta)) \leq f(\zeta)$ 。

**条件 2**<sup>[37]</sup> 对  $\forall A \in S$ , s.t.  $v(A) > 0$ , 有:

$$\prod_{i=0}^{\infty} (1 - v_i(A)) = 0 \quad (10)$$

其中,  $v_i(A)$  表示解在空间  $A$  上, 算法  $M$  在第  $t$  次迭代过程中能够找到解的概率测度。

**定理 1** 对于任意渐近收敛算法, CSOBL 算法不会破坏原算法的渐近收敛性。

**证明** 对于任意渐近收敛算法, 符合条件 1, 即渐近收敛算法中每个个体的第  $t+1$  次最优位置一定优于第  $t$  次最优位置, 同时符合条件 2, 即在任意多次迭代后, 算法以概率为 1 收敛。对于 CSOBL 算法, 余弦相似度取值在  $-1$  到  $1$  之间, 所以每个个体以概率  $P=1/2$  反向, 个体位置更新公式:

$$x_{i+1}^* = \begin{cases} (\text{lb} + \text{ub} - x_i) \frac{\sum_{i=1}^D (x_{ii} \mu_{ii})}{\sqrt{\sum_{i=1}^D x_{ii}^2} \sqrt{\sum_{i=1}^D \mu_{ii}^2}}, & x_i \frac{\sum_{i=1}^D (x_{ii} \mu_{ii})}{\sqrt{\sum_{i=1}^D x_{ii}^2} \sqrt{\sum_{i=1}^D \mu_{ii}^2}} < 0 \\ x, & \frac{\sum_{i=1}^D (x_{ii} \mu_{ii})}{\sqrt{\sum_{i=1}^D x_{ii}^2} \sqrt{\sum_{i=1}^D \mu_{ii}^2}} \geq 0 \end{cases} \quad (11)$$

其中,  $x_{i+1}^*$  为个体位置更新的中间结果, 对于任意渐近收敛算法, 个体在一次迭代后的位置为

$$x_{i+1} = \begin{cases} x_{i+1}^*, & f(x_{i+1}^*) \leq f(x_i) \\ x_i, & f(x_{i+1}^*) > f(x_i) \end{cases} \quad (12)$$

即 CSOBL 算法符合条件 1, CSOBL 算法不会破坏原始渐近收敛算法的渐近收敛性。定理 1 得证。

**定理 2** CSOBL 算法满足渐近收敛准则。

**证明** 由定理 1 可知, 在 CSOBL 算法中, 总是有  $f(x_{i+1}) \leq f(x_i)$ , 在迭代次数足够多时,  $f(x_i)$  会逐渐逼近最优值, 即  $\lim f(x_i) = f(g_i^{\text{best}})$ , 其中  $g_i^{\text{best}}$  为算法第  $t$  迭代中的全局最优值位置。因此可以得到:

$$\prod_{i=0}^{\infty} \frac{x_i - g_i^{\text{best}}}{|g_i^{\text{best}}|} = 0 \quad (13)$$

所以有:

$$\prod_{i=0}^{\infty} (1 - P(x_i = g_i^{\text{best}})) = 0 \quad (14)$$

即 CSOBL 算法满足条件 2, 所以 CSOBL 算法最终以概率为 1 收敛至迭代最优值  $g_{\text{best}}$ 。定理 2 证毕。

## 4 仿真实验与结果分析

实验仿真环境为 Intel® Core™, CPU@3.30 GHz, RAM 16, Windows 10, Matlab 2020a。本实验把 CSOBL 策略分别应用于标准粒子群算法<sup>[38]</sup>、标准遗传算法<sup>[39]</sup>和灰狼算法<sup>[40]</sup>中, 得到 CSOPSO、CSOGA 和 CSOGWO, 并与 PSO、GA、GWO 和几个经典基于反向优化算法进行对比。为了测试应用了本文所提出策略的算法性能, 选取了表 1 中的 12 个经典测试函数进行仿真实验分析, 表 1 给出了经典测试函数的表达式、搜索范围和理论最优值。单峰函数  $F_1 \sim F_4$ 、 $F_{12}$  只有一个全局最优点, 因此适合作为算法开发的基准。多峰函数  $F_5 \sim F_{11}$  具有大量的局部最优点, 求解难度较高。

### 4.1 CSOBL 策略与标准算法的实验结果分析

#### 4.1.1 参数设置

为了比较结果的公平性, 实验参数设置如下: PSO 算法和 CSOPSO 算法的参数设置为: 加速因子  $c_1 = c_2 = 2$ , 惯性权重  $\omega$  由 0.9 到 0.4 线性递减, 最大迭代次数  $\text{FEs} = 1\ 000$ , 种群规模  $N = 50$ , 维数  $D = 30$ ; GA 算法和 CSOGA 算法的参数设置为: 交叉概率  $P_c = 0.70$ , 变异概率  $P_m = 0.05$ , 最大迭代次数  $\text{FEs} = 1\ 000$ , 种群规模  $N = 50$ , 维数  $D = 30$ ; GWO 算法和 CSOGWO 算法的参数设置为: 种群规模  $N = 100$ , 最大迭代次数  $\text{FEs} = 1\ 000$ , 维数  $D = 30$ 。

#### 4.1.2 实验结果及分析

各算法对 12 个标准测试函数分别独立运行 10 次, 记录各算法的平均值、最优值、标准差等结果, 用于评价算法的性能。将改进算法与其对应的标准算法进行比较, 结果如表 2 所示, 其中加黑部分表示结果更接近最优解。从表 2 的结果可以看出, 应用了本文所提出策略的综合算法在大部分测试函数中的表现明显优于标准算法的结果。对于得到的优化结果来说, 在 10 个函数上 CSOPSO 比 PSO 的寻优结果更接近最优值, 3 个函数优势不明显。在 8 个函数上 CSOGA 比 GA 的寻优结果更接近最优值, 4 个函数优势不明显。在 5 个函数上 CSOGWO 比 GWO 的寻优结果更接近最优值, 3 个函数优势不明显, 4 个函数二者表现相同。比较结果充分说明了余弦相似度反向策略较标准算法更具优势。这是因为余弦相似度反向策略充分利用了余弦相似度的方向特性, 保证了种群的多样性, 提高了计算精度, 一定程度上避免了陷入局部最优。另外, 本文对各测试函数进行了收敛性测试, 并给出了它们的收敛比

较图, 横坐标表示种群迭代次数, 纵坐标表示平均适应度值, 如图 6 所示。从图 6 的收敛走势图中可以看出, 函数  $F_1 \sim F_7$ 、 $F_{11}$  的收敛速度十分迅速, 同时,  $F_1$ 、 $F_4$ 、 $F_6$ 、 $F_7$ 、 $F_{11}$ 、 $F_{12}$  的收敛速度迅速下降且并未趋于稳定, 在迭代次数较少的情况下即可找到全局最优解, 这是因为使用余弦相似度调整反向学习权

重能够在保证多样性的情况下迅速向最优解靠拢, 从而进一步说明了余弦相似度反向策略能够强化局部区域的搜索能力, 避免不必要的反向, 从而使算法具有更快的收敛速度。但在函数  $F_{11}$  中, 收敛速度提高不大, 这是由于函数本身是具有多个局部最优值导致的, 使得算法收敛速度相对较慢。

表 1 经典测试函数  
Tab.1 Classical test function

函数名称	测试函数	搜索范围	最优解
Sphere	$F_1 = \sum_{i=1}^D x_i^2$	$[-100, 100]$	0
Rosenbrock	$F_2 = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-5, 10]$	0
dixon-price	$F_3 = (x_1 - 1)^2 + \sum_{i=1}^D i(2x_i^2 - x_{i-1})^2$	$[-10, 10]$	0
sum of different powers	$F_4 = \sum_{i=1}^D  x_i ^{i+1}$	$[-1, 1]$	0
Ackley	$F_5 = -a \exp\left(-b \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(cx_i)\right) + a + \exp(1)$ $a=20, b=0.2, c=2\pi$	$[-32.76, 32.76]$	0
Rastrigin	$F_6 = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0
Griewank	$F_7 = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \frac{x_i}{\sqrt{i}} + 1$	$[-600, 600]$	0
Powell	$F_8 = \sum_{i=1}^{D/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	$[-4, 5]$	0
Levy	$F_9 = \sin^2(\pi\omega_1) + \sum_{i=1}^{D-1} (\omega_i - 1)^2 [1 + 10 \sin^2(\pi\omega_i + 1)] + (\omega_D - 1)^2 (1 + \sin^2(2\pi\omega_D))$ $\omega_i = 1 + \frac{\omega_i - 1}{4}$	$[-10, 10]$	0
Schwefel	$F_{10} = 418.9829D - \sum_{i=1}^D x_i \sin \sqrt{ x_i }$	$[-500, 500]$	0
Alpine	$F_{11} = \sum_{i=1}^D  x_i \sin x_i + 0.1x_i $	$[-10, 10]$	0
sum squares	$F_{12} = \sum_{i=1}^D ix_i^2$	$[-10, 10]$	0

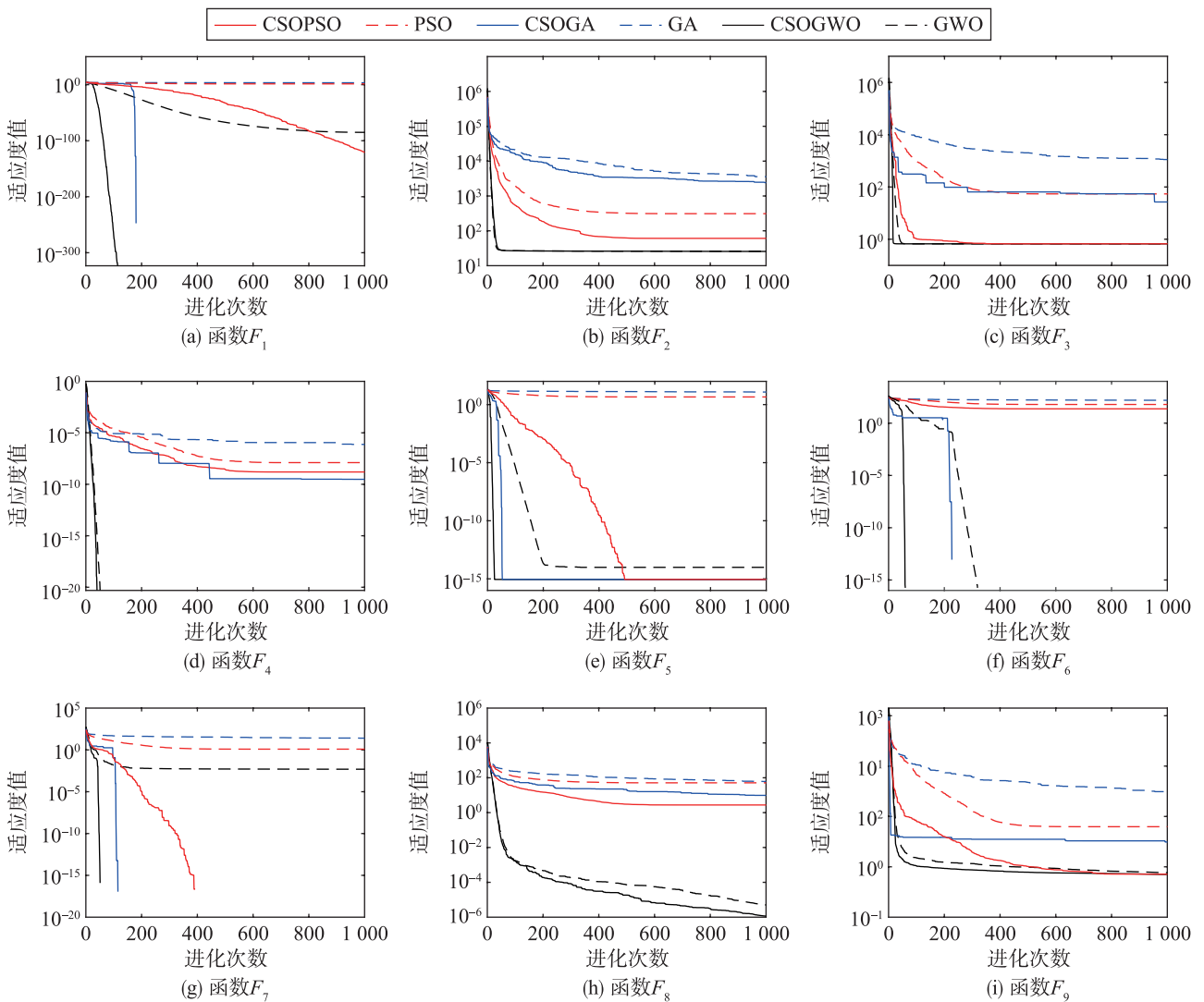
表 2 各算法对 12 个标准测试函数优化结果  
Tab.2 The optimization results of various algorithms on 12 test functions

函数	特征	平均结果及标准方差					
		PSO	CSOPSO	GA	CSOGA	GWO	CSOGWO
$F_1$	均值	$1.57 \times 10^1$	<b>0</b>	$1.94 \times 10^3$	<b>0</b>	$1.32 \times 10^{-84}$	<b>0</b>
	方差	$9.28 \times 10^0$	0	$3.69 \times 10^2$	0	$2.34 \times 10^3$	$1.84 \times 10^3$
$F_2$	均值	$8.95 \times 10^1$	<b><math>2.63 \times 10^1</math></b>	$2.27 \times 10^3$	<b><math>1.60 \times 10^3</math></b>	$2.71 \times 10^1$	<b><math>2.50 \times 10^1</math></b>
	方差	$7.20 \times 10^1$	$1.10 \times 10^1$	$2.03 \times 10^3$	$1.23 \times 10^2$	$5.35 \times 10^4$	$3.72 \times 10^4$
$F_3$	均值	$5.57 \times 10^0$	<b><math>6.67 \times 10^{-1}</math></b>	$5.48 \times 10^2$	<b><math>8.01 \times 10^{-1}</math></b>	<b><math>6.67 \times 10^{-1}</math></b>	<b><math>6.67 \times 10^{-1}</math></b>
	方差	$2.59 \times 10^0$	$3.58 \times 10^{-2}$	$6.81 \times 10^2$	$7.35 \times 10^1$	$4.60 \times 10^4$	$3.96 \times 10^4$
$F_4$	均值	$6.09 \times 10^{-10}$	<b>0</b>	$1.25 \times 10^{-9}$	<b>0</b>	<b>0</b>	<b>0</b>
	方差	$1.79 \times 10^{-8}$	$2.38 \times 10^{-8}$	$1.13 \times 10^{-6}$	$1.90 \times 10^{-9}$	$3.25 \times 10^{-2}$	$1.56 \times 10^{-2}$

表2 各算法对12个标准测试函数优化结果(续)

Tab.2 The optimization results of various algorithms on 12 test functions (continued)

函数	特征	平均结果及标准方差					
		PSO	CSOPSO	GA	CSOGA	GWO	CSOGWO
$F_5$	均值	$4.16 \times 10^0$	<b><math>8.88 \times 10^{-16}</math></b>	$1.08 \times 10^1$	<b><math>8.88 \times 10^{-16}</math></b>	<b><math>7.99 \times 10^{-15}</math></b>	<b><math>8.88 \times 10^{-16}</math></b>
	方差	$6.22 \times 10^{-1}$	0	$8.43 \times 10^{-1}$	0	$2.01 \times 10^0$	$6.48 \times 10^{-1}$
$F_6$	均值	$3.94 \times 10^1$	<b>0</b>	$1.18 \times 10^2$	<b>0</b>	<b>0</b>	<b>0</b>
	方差	$1.96 \times 10^1$	$1.53 \times 10^1$	$1.80 \times 10^1$	0	$4.91 \times 10^1$	$2.71 \times 10^1$
$F_7$	均值	$1.04 \times 10^0$	<b>0</b>	$1.93 \times 10^1$	<b>0</b>	<b>0</b>	<b>0</b>
	方差	$1.53 \times 10^{-1}$	0	$3.54 \times 10^0$	0	$2.28 \times 10^1$	$2.25 \times 10^1$
$F_8$	均值	$1.21 \times 10^0$	<b><math>1.06 \times 10^{-3}</math></b>	$3.13 \times 10^1$	<b><math>7.16 \times 10^{-1}</math></b>	<b><math>8.35 \times 10^{-7}</math></b>	$1.10 \times 10^{-6}$
	方差	$4.30 \times 10^1$	$2.92 \times 10^0$	$2.22 \times 10^1$	$9.31 \times 10^0$	$5.24 \times 10^2$	$4.14 \times 10^2$
$F_9$	均值	$6.01 \times 10^{-1}$	<b><math>4.58 \times 10^{-6}</math></b>	$1.31 \times 10^1$	<b><math>2.23 \times 10^0</math></b>	$1.16 \times 10^0$	<b><math>6.59 \times 10^{-1}</math></b>
	方差	$7.77 \times 10^{-1}$	$3.53 \times 10^{-1}$	$1.33 \times 10^1$	$6.21 \times 10^{-2}$	$7.57 \times 10^1$	$3.31 \times 10^1$
$F_{10}$	均值	$3.86 \times 10^3$	<b><math>1.76 \times 10^2</math></b>	$7.08 \times 10^3$	<b><math>6.79 \times 10^3</math></b>	$6.06 \times 10^3$	<b><math>5.92 \times 10^3</math></b>
	方差	$6.99 \times 10^2$	$1.86 \times 10^2$	$3.16 \times 10^2$	$1.12 \times 10^3$	$9.86 \times 10^2$	$1.30 \times 10^3$
$F_{11}$	均值	$8.92 \times 10^{-1}$	<b><math>2.68 \times 10^{-70}</math></b>	$1.05 \times 10^1$	<b>0</b>	$7.28 \times 10^{-48}$	<b>0</b>
	方差	$1.95 \times 10^0$	$2.97 \times 10^0$	$1.81 \times 10^0$	0	$3.82 \times 10^0$	$1.94 \times 10^0$
$F_{12}$	均值	$3.36 \times 10^0$	<b>0</b>	$1.83 \times 10^2$	<b>0</b>	$2.55 \times 10^{-86}$	<b>0</b>
	方差	$6.13 \times 10^0$	0	$4.62 \times 10^1$	0	$4.14 \times 10^2$	$1.94 \times 10^0$



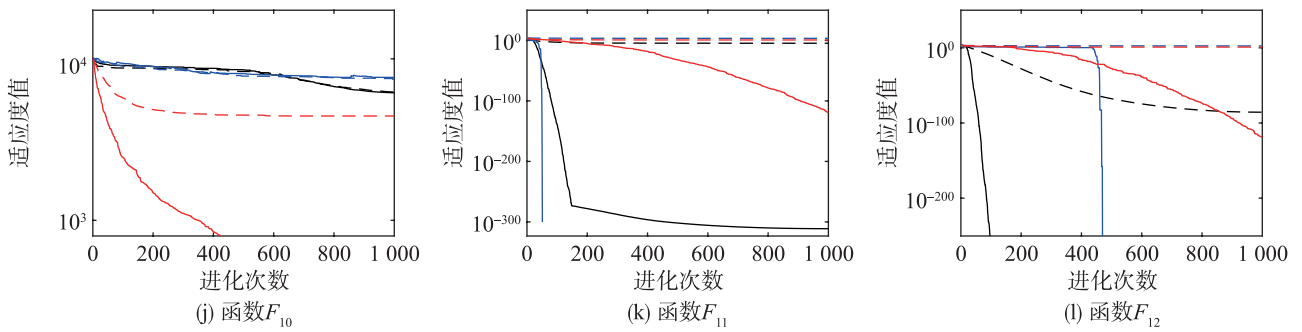


图 6 经典测试函数优化的收敛曲线

Fig.6 Convergence curves for optimization of classical test functions

4.2 CSOGWO 与其他应用 OBL 策略的算法实验结果分析

COPSO<sup>[26]</sup>是应用了重心反向策略的粒子群算法, DCOPSO<sup>[29]</sup>是对最优粒子逐维重心反向的粒子群算法, OGA-CM<sup>[19]</sup>是应用了反向学习和柯西变异的遗传算法, ODE<sup>[17]</sup>是基于反向学习的差分进化算法, OLGWO<sup>[41]</sup>是 Long 等提出的反向学习灰狼算法。将 CSOGWO 与上述文献中的进化算法进行对比。为了与文献中的结果保持一致, 各算法的种群大小设置为 60, 最大迭代次数设置为 500, 每个函数的维数都设置为 30, 其他参数均与原论文相同。

结果如表 3 所示, 其中加黑部分表示结果更接近最优解。从表 3 中可以直观地看到, CSOGWO 在 5 个测试函数中找到了最优解, 在 6 个测试函数中的寻优结果优于其他 5 个算法。基于余弦相似度的反向是方向性的, 能够在快速收敛的同时保证种群多样性, 提高计算精度。算法加入柯西扰动, 能够增强全局搜索能力, 进一步提高种群多样性, 很大程度上避免种群陷入局部极值。但是 CSOGWO 在测试函数  $F_2$ 、 $F_9$  中取得的成果不理想, 这是因为在多维的情况下, 函数相关系数太多, 无法精确地进行余弦反向。

表 3 各算法的实验对比结果

Tab.3 The experimental comparison results of various algorithms

测试函数	COPSO	DCOPSO	OGA-CM	ODE	OLGWO	CSOGWO
$F_1$	$1.28 \times 10^{-12}$	$5.67 \times 10^{-27}$	$1.84 \times 10^3$	$5.57 \times 10^{-30}$	$2.06 \times 10^{-54}$	<b>0</b>
$F_2$	$3.87 \times 10^{-2}$	$4.91 \times 10^0$	$4.51 \times 10^3$	$5.29 \times 10^{-17}$	<b><math>1.97 \times 10^{-28}</math></b>	$2.70 \times 10^1$
$F_5$	$7.86 \times 10^{-8}$	$3.99 \times 10^{-14}$	$1.18 \times 10^1$	$2.81 \times 10^1$	$2.88 \times 10^1$	<b><math>8.88 \times 10^{-16}</math></b>
$F_6$	$1.24 \times 10^{-12}$	<b>0</b>	$1.42 \times 10^2$	<b>0</b>	<b>0</b>	<b>0</b>
$F_7$	$8.86 \times 10^{-13}$	<b>0</b>	$2.24 \times 10^1$	$3.96 \times 10^{-3}$	$1.57 \times 10^{-6}$	<b>0</b>
$F_9$	$4.65 \times 10^{-2}$	$1.00 \times 10^{-4}$	$2.38 \times 10^1$	$2.97 \times 10^{-31}$	<b><math>1.05 \times 10^{-51}</math></b>	$4.46 \times 10^{-1}$
$F_{11}$	$7.96 \times 10^{-9}$	$6.12 \times 10^{-9}$	$1.14 \times 10^1$	$2.07 \times 10^{-26}$	$9.05 \times 10^{-48}$	<b>0</b>
$F_{12}$	$2.36 \times 10^{-12}$	$3.05 \times 10^{-27}$	$2.56 \times 10^2$	$5.10 \times 10^1$	<b>0</b>	<b>0</b>

4.3 非参数检验

为了对实验结果有一个更加全面的了解, 对 CSOGWO 和其他算法的结果逐一进行非参数检验<sup>[42]</sup>。选用 Mann-Whitney 检验方法, 显著水平  $p = 0.05$ 。由于文[36]不包含 ODE 和 OLGWO 的检验数据, 故在此省略。表 4 列出了其余 3 种算法的检验数据。以“COPSO”列与“ $F_1$ ”行交叉的数据“0.013  $\uparrow$ ”为例, 其中, “0.013”为 Mann-Whitney 检验的渐近显著性值, “ $\uparrow$ ”表示 COPSO 在  $F_1$  上运行数据的秩均值大于 CSOGWO 在  $F_1$  上运行数据的秩

均值。

表 4 Mann-Whitney 检验结果

Tab.4 The result of Mann-Whitney test

测试函数	COPSO	DCOPSO	OGA-CM
$F_1$	0.013 $\uparrow$	1.000 $\uparrow$	0.000 $\uparrow$
$F_2$	1.000 $\downarrow$	0.728 $\uparrow$	0.000 $\uparrow$
$F_5$	0.001 $\uparrow$	0.341 $\uparrow$	0.000 $\uparrow$
$F_6$	0.068 $\downarrow$	1.000 $\downarrow$	0.000 $\uparrow$
$F_7$	0.061 $\uparrow$	0.683 $\downarrow$	0.000 $\uparrow$

表4 Mann-Whitney 检验结果(续)  
Tab.4 The result of Mann-Whitney test(continued)

测试函数	COPSO	DCOPSO	OGA-CM
$F_9$	1.000 ↓	0.007 ↓	0.077 ↓
$F_{11}$	0.023 ↑	0.023 ↑	0.000 ↑
$F_{12}$	0.013 ↓	1.000 ↓	0.000 ↑
劣于对比算法个数	1	1	0
与对比算法无明显差异个数	4	5	1
优于对比算法个数	3	2	7

从表4可以看出 CSOGWO 劣于对比函数的次数明显较少, 检验结果表明, 在统计意义上应用了本文所提策略的算法明显占优。同时结合表3及以上分析, 可以说明余弦相似度反向策略有效提高了算法的收敛速度与精度。

## 5 结论

本文提出一种基于余弦相似度反向的快速收敛策略提高种群的全局搜索能力, 通过大量的理论分析和实验结果表明, 本策略具有快速收敛能力, 且能够保证种群多样性, 能够带领种群探索更大的空间范围, 找到精度更高的解。将其与三种自然计算方法结合, 与标准算法和基于反向学习策略的改进算法进行比较, 选取12个测试函数验证其普适性和有效性。实验结果表明, 与其他先进的反向学习算法相比, 应用所提策略的算法具有更好的竞争力, 表现出了很好的结果。但是, 对于某些并非对单一个体进行变异的算法, 本文策略具有一定的局限性。因此在后续的研究工作中, 考虑将余弦相似度与欧氏距离结合, 改进算法性能。

## 参考文献

- [1] Kari L, Rozenberg G. The many facets of natural computing[J]. *Communications of the ACM*, 2008, 51(10): 72–83.
- [2] 康琦, 安静, 汪镭, 等. 自然计算的研究综述[J]. *电子学报*, 2012, 40(3): 548–558.  
Kang Q, An J, Wang L, et al. Nature-inspired computation; A survey[J]. *Acta Electronica Sinica*, 2012, 40(3): 548–558.
- [3] 莫宏伟, 徐立芳. 自然计算与计算主义[C]//第十一届中国人工智能学术年会. 北京: 北京邮电大学出版社, 2005: 210–215.  
Mo H W, Xu L F. Natural computing and computationalism[C]//The 11th China Annual Conference on Artificial Intelligence. Beijing: Beijing University of Posts and Telecommunications Press, 2005: 210–215.
- [4] 汪镭, 张永韡, 郭为安, 等. 自然计算发展趋势研究[J]. *微型电脑应用*, 2010, 26(7): 1–5, 10.  
Wang L, Zhang Y W, Guo W A, et al. On development trends of nature inspired computing[J]. *Microcomputer Applications*, 2010, 26(7): 1–5, 10.
- [5] Khan A, Jaffar M A. Genetic algorithm and self organizing map based fuzzy hybrid intelligent method for color image segmentation[J]. *Applied Soft Computing*, 2015, 32: 300–310.
- [6] Bharti K K, Singh P K. Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering[J]. *Applied Soft Computing*, 2016, 43: 20–34.
- [7] Yang Y F, Cui Z M, Wu J, et al. Fuzzy c-means clustering and opposition-based reinforcement learning for traffic congestion identification[J]. *Journal of Information and Computational Science*, 2012, 9(9): 2441–2450.
- [8] Huang W, Oh S K. Identification of fuzzy inference systems by means of a multiobjective opposition-based space search algorithm[J/OL]. *Mathematical Problems in Engineering*, 2013 [2021–06–21]. <https://www.hindawi.com/journals/mpe/2013/725017/>. DOI: 10.1155/2013/725017.
- [9] da Silveira L A, Soncoo-Álvarez J L, de Lima T A, et al. Memetic and opposition-based learning genetic algorithms for sorting unsigned genomes by translocations[M]//Advances in Intelligent Systems and Computing, vol. 419. Berlin, Germany: Springer, 2015: 73–85.
- [10] 张凯, 陈彬, 许志伟. 基于多目标进化策略算法的DNA核酸编码设计[J]. *电子与信息学报*, 2020, 42(6): 1365–1373.  
Zhang K, Chen B, Xu Z W. A multiobjective evolution strategy algorithm for DNA sequence design[J]. *Journal of Electronics & Information Technology*, 2020, 42(6): 1365–1373.
- [11] 傅蔚阳, 刘以安, 薛松. 基于灰狼算法与小波神经网络的目标威胁评估[J]. *浙江大学学报(工学版)*, 2018, 52(4): 680–686.  
Fu W Y, Liu Y A, Xue S. Target threat assessment using grey wolf optimization and wavelet neural network[J]. *Journal of Zhejiang University (Engineering Science)*, 2018, 52(4): 680–686.
- [12] 金泽芬芬, 侯志强, 余旺盛, 等. 基于量子遗传算法的视觉目标跟踪[J]. *电子学报*, 2020, 48(8): 1493–1501.  
Jin Z F F, Hou Z Q, Yu W S, et al. An object tracking approach based on quantum genetic algorithm[J]. *Acta Electronica Sinica*, 2020, 48(8): 1493–1501.
- [13] 李泽, 田增山, 王中春, 等. 基于粒子群优化的多径辅助室内定位算法[J]. *电子学报*, 2020, 48(10): 1952–1960.

- Li Z, Tian Z S, Wang Z C, et al. Multipath-assisted indoor localization algorithm based on particle swarm optimization[J]. *Acta Electronica Sinica*, 2020, 48(10): 1952 – 1960.
- [14] Draa A, Benayad Z, Djenna F Z. An opposition-based firefly algorithm for medical image contrast enhancement[J]. *International Journal of Information and Communication Technology*, 2015, 7(4/5): 385 – 405.
- [15] 王蓉芳, 焦李成, 刘芳, 等. 自适应动态控制种群规模的自然计算方法[J]. *软件学报*, 2012, 23(7): 1760 – 1772.  
Wang R F, Jiao L C, Liu F, et al. Nature computation with self-adaptive dynamic control strategy of population size[J]. *Journal of Software*, 2012, 23(7): 1760 – 1772.
- [16] Tizhoosh H R. Opposition-based learning: A new scheme for machine intelligence[C]//International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. Piscataway, USA: IEEE, 2005: 695 – 701.
- [17] Rahnamayan S, Tizhoosh H R, Salama M M A. Opposition-based differential evolution algorithms[C]//IEEE International Conference on Evolutionary Computation. Piscataway, USA: IEEE, 2006: 2010 – 2017.
- [18] 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.
- [19] Iqbal M A, Khan N K, Jaffar M A, et al. Opposition based genetic algorithm with Cauchy mutation for function optimization[C/OL]//International Conference on Information Science and Applications. Piscataway, USA: IEEE [2021 – 05 – 15]. <https://ieeexplore.ieee.org/document/5480382/>. DOI: 10.1109/ICISA.2010.5480382.
- [20] Ewees A A, Elaziz M A, Oiego D. A new multi-objective optimization algorithm combined with opposition-based learning[J/OL]. *Expert Systems with Applications*, 2021, 165 [2021 – 05 – 20]. <https://www.sciencedirect.com/science/article/pii/S0957417420306539>. DOI: 10.1016/j.eswa.2020.113844.
- [21] Chen K, Zhou F Y, Yin L, et al. A hybrid particle swarm optimizer with sine cosine acceleration coefficients[J]. *Information Sciences*, 2018, 422: 218 – 241.
- [22] Feng X Y, Liu A, Sun W L, et al. A dynamic generalized opposition-based learning fruit fly algorithm for function optimization [C/OL]//IEEE Congress on Evolutionary Computation. Piscataway, USA: IEEE [2021 – 02 – 21]. <https://ieeexplore.ieee.org/document/8477794/>. DOI: 10.1109/CEC.2018.8477794.
- [23] Dhargupta S, Ghosh M, Mirjalili S, et al. Selective opposition based grey wolf optimization[J/OL]. *Expert Systems with Applications*, 2020, 151 [2021 – 04 – 14]. <https://www.sciencedirect.com/science/article/pii/S095741742030213X?via%3Dihub>. DOI: 10.1016/j.eswa.2020.113389.
- [24] Dhargupta S, Ghosh M, Mirjalili S, et al. Selective opposition based grey wolf optimization[J/OL]. *Expert Systems with Applications*, 2020, 151 [2021 – 05 – 01]. <https://www.mdpi.com/1999-4893/11/4/47>. DOI: 10.1016/j.eswa.2020.113389.
- [25] 孙辉, 邓志诚, 赵嘉, 等. 混合均值中心反向学习粒子群优化算法[J]. *电子学报*, 2019, 47(9): 1809 – 1818.  
Sun H, Deng Z C, Zhao J, et al. Hybrid mean center opposition-based learning particle swarm optimization[J]. *Acta Electronica Sinica*, 2019, 47(9): 1809 – 1818.
- [26] Rahnamayan S, Jesuthasan J, Bourennani F, et al. Computing opposition by involving entire population[C]//IEEE Congress on Evolutionary Computation. Piscataway, USA: IEEE, 2014: 1800 – 1807.
- [27] 周凌云, 丁立新, 彭虎, 等. 一种邻域重心反向学习的粒子群优化算法[J]. *电子学报*, 2017, 45(11): 2815 – 2824.  
Zhou L Y, Ding L X, Peng H, et al. Neighborhood centroid opposition-based particle swarm optimization[J]. *Acta Electronica Sinica*, 2017, 45(11): 2815 – 2824.
- [28] 邵鹏, 吴志健, 周炫余, 等. 基于折射原理反向学习模型的改进粒子群算法[J]. *电子学报*, 2015, 43(11): 2137 – 2144.  
Shao P, Wu Z J, Zhou X Y, et al. Improved particle swarm optimization algorithm based on opposite learning of refraction[J]. *Acta Electronica Sinica*, 2015, 43(11): 2137 – 2144.
- [29] 罗强, 季伟东, 徐浩天, 等. 一种最优粒子逐维变异的粒子群优化算法[J]. *小型微型计算机系统*, 2020, 41(2): 259 – 263.  
Luo Q, Ji W D, Xu H T, et al. Particle swarm optimization with global best particle dimension-by-dimension mutation[J]. *Journal of Chinese Computer Systems*, 2020, 41(2): 259 – 263.
- [30] Koufakou A, Georgiopoulos M. A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes [J]. *Data Mining and Knowledge Discovery*, 2010, 20(2): 259 – 289.
- [31] Han S, Zhao C, Meng W X, et al. Cosine similarity based fingerprinting algorithm in WLAN indoor positioning against device diversity[C]//IEEE International Conference on Communications. Piscataway, USA: IEEE, 2015: 2710 – 2714.
- [32] 刘爱琴, 张继福, 荀亚玲. 基于大熵值变化区域和余弦相似度的离群迭代算法[J]. *小型微型计算机系统*, 2013, 34(7): 1518 – 1521.  
Liu A Q, Zhang J F, Xun Y L. Outlier iteration algorithm based on large entropy vary and cosine similarity[J]. *Journal of Chinese Computer Systems*, 2013, 34(7): 1518 – 1521.
- [33] Wang H, Li C H, Liu Y, et al. A hybrid particle swarm algorithm with Cauchy mutation[C]//IEEE Swarm Intelligence Sympo-

- sium. Piscataway, USA: IEEE, 2007: 356 – 360.
- [34] 何庆, 林杰, 徐航. 混合柯西变异和均匀分布的蝗虫优化算法[J]. 控制与决策, 2021, 36(7): 1558 – 1568.  
He Q, Lin J, Xu H. Hybrid Cauchy mutation and uniform distribution of grasshopper optimization algorithm[J]. Control and Decision, 2021, 36(7): 1558 – 1568.
- [35] 张孟健, 龙道银, 王霄, 等. 基于马尔科夫链的灰狼优化算法收敛性研究[J]. 电子学报, 2020, 48(8): 1587 – 1595.  
Zhang M J, Long D Y, Wang X, et al. Research on convergence of grey wolf optimization algorithm based on Markov chain[J]. Acta Electronica Sinica, 2020, 48(8): 1587 – 1595.
- [36] Solis F J, Wets R J B. Minimization by random search techniques[J]. Mathematics of Operations Research, 1981, 6(1): 19 – 30.
- [37] 孙小晴, 程昊, 张璐瑶, 等. 一种基于空间分割搜索策略的自然计算方法[J]. 系统仿真学报, 2021, 33(11): 2589 – 2605.  
Sun X Q, Cheng H, Zhang L Y, et al. A natural computing method based on spatial division search strategy[J]. Journal of System Simulation, 2021, 33(11): 2589 – 2605.
- [38] Kennedy J, Eberhart R. Particle swarm optimization[C]//International Conference on Neural Networks. Piscataway, USA: IEEE, 1995: 1942 – 1948.
- [39] 孟凡超, 初佃辉, 李克秋, 等. 基于混合遗传模拟退火算法的 SaaS 构件优化放置[J]. 软件学报, 2016, 27(4): 916 – 932.  
Meng F C, Chu D H, Li K Q, et al. Solving SaaS components optimization placement problem with hybrid genetic and simulated annealing algorithm[J]. Journal of Software, 2016, 27(4): 916 – 932.
- [40] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46 – 61.
- [41] Long W, Jiao J J, Liang X M, et al. A random opposition-based learning grey wolf optimizer[J]. IEEE Access, 2019, 7: 113810 – 113825.
- [42] Derrac J, García S, Molina D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. Swarm and Evolutionary Computation, 2011, 1(1): 3 – 18.

## 作者简介

孙小琳(1997 –), 女, 硕士生。研究领域为群体智能。

季伟东(1978 –), 男, 博士, 教授。研究领域为大数据, 群体智能。

王旭(1997 –), 男, 硕士生。研究领域为群体智能, 知识图谱。

(上接第 707 页)

- [15] Lu S Y, Wang S H, Zhang Y D. A classification method for brain MRI via MobileNet and feedforward network with random weights[J]. Pattern Recognition Letters, 2020, 140: 252 – 260.
- [16] Karthik R, Menaka R, Hariharan M. Learning distinctive filters for COVID-19 detection from chest X-ray using Shuffled residual CNN[J]. Applied Soft Computing, 2021. DOI: 10.1016/j.asoc.2020.106744.
- [17] Lu Z Y, Bai Y Z, Chen Y, et al. The classification of gliomas based on a pyramid dilated convolution resnet model[J]. Pattern Recognition Letters, 2020, 133: 173 – 179.
- [18] Xiao Z Y, He K H, Liu J J, et al. Multi-view hierarchical split network for brain tumor segmentation[J]. Biomedical Signal Processing and Control, 2021. DOI: 10.1016/j.bspc.2021.102897.
- [19] Bakas S, Akbari H, Sotiras A, et al. Data descriptor: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features[J]. Nature Scientific Data, 2017. DOI:10.1038/sdata.2017.117.
- [20] Nuechterlein N, Mehta S. 3D-ESPNet with pyramidal refinement for volumetric brain tumor image segmentation[M]//Lecture Notes in Computer Science, vol. 11384. Berlin, Germany: Springer, 2018: 245 – 253.
- [21] Mckinley R, Rebsamen M, Meier R, et al. Triplanar ensemble of 3D-to-2D cnns with label-uncertainty for brain tumor segmentation[M]//Lecture Notes in Computer Science, vol. 10379. Berlin, Germany: Springer, 2020: 379 – 387.
- [22] Milan D, Stijn B, Karel D, et al. Automated MRI based pipeline for segmentation and prediction of grade, IDH mutation and 1p19q co-deletion in glioma[J]. Computerized Medical Imaging and Graphics, 2021, DOI: 10.1016/j.compmedimag.2020.101831.

## 作者简介

霍国强(1996 –), 男, 硕士生。研究领域为医学图像处理。

李镛(1974 –), 男, 博士, 教授。研究领域为智能信息处理, 滤波器设计, 数字系统和微系统设计。

关欣(1978 –), 女, 博士, 副教授。研究领域为音乐信号处理。