

# 基于刺激-响应机制的改进鸡群算法

王英聪, 刘 驰, 王延峰<sup>†</sup>

(郑州轻工业大学 电气信息工程学院, 郑州 450002)

**摘要:** 针对鸡群算法收敛速度慢且容易陷入局部最优的不足, 从平衡勘探与开发的角度提出一种基于刺激-响应机制的改进鸡群算法. 首先, 考虑公鸡的引领作用为其设计勘探和开发两种搜索方程; 然后, 根据种群聚集度和平均改进度定义执行搜索方程的刺激和阈值; 最后, 在刺激-响应机制下, 公鸡充分发挥引领作用, 带领鸡群完成勘探与开发. 将改进算法用于基准函数优化和食管癌生存风险预测模型优化, 并与其他改进的鸡群算法进行对比, 表明了所提出算法的有效性.

**关键词:** 鸡群算法; 刺激-响应机制; 种群聚集度; 平均改进度; 生存风险预测

**中图分类号:** TP18      **文献标志码:** A

**DOI:** 10.13195/j.kzyjc.2021.1059

**引用格式:** 王英聪, 刘驰, 王延峰. 基于刺激-响应机制的改进鸡群算法[J]. 控制与决策, 2023, 38(1): 58-66.

## Chicken swarm optimization algorithm based on stimulus-response mechanism

WANG Ying-cong, LIU Chi, WANG Yan-feng<sup>†</sup>

(School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China)

**Abstract:** The original chicken swarm optimization algorithm converges slowly and easily falls into the local optimum. To overcome this issue, this paper proposes a chicken swarm optimization algorithm based on stimulus-response mechanism from the perspective of exploration-exploitation balance. Firstly, two search equations focusing on exploration and exploitation respectively are designed for roosters. Secondly, the stimulus and threshold of performing different search equations are designed according to population aggregation degree and average improvement degree. Finally, roosters play a leading role and guide the chickens to complete the exploration and exploitation under the stimulus-response mechanism. The experimental results on benchmark functions and survival risk prediction of esophageal cancer show the high efficiency of the proposed algorithm, compared with other improved chicken swarm optimization algorithms.

**Keywords:** chicken swarm optimization algorithm; stimulus-response mechanism; population aggregation degree; average improvement degree; survival risk prediction

## 0 引言

群体智能算法是一类模拟社会性昆虫或动物集群行为的随机优化算法, 算法中的个体仅通过简单的行为规则就能完成对复杂问题的求解. 与传统优化方法相比, 群体智能算法具有与初始值无关、不依赖梯度信息、对函数要求低、求解性能良好等特点. 目前, 比较常见的群体智能算法有蚁群算法<sup>[1]</sup>、粒子群算法<sup>[2]</sup>、人工蜂群算法<sup>[3]</sup>、人工鱼群算法<sup>[4]</sup>、狼群算法<sup>[5]</sup>和萤火虫算法<sup>[6]</sup>等. 这些算法一经提出便受到众多学者的关注, 已广泛应用于多种研究领域.

鸡群(chicken swarm optimization, CSO)算法是近

年来提出的一种较为新颖的群体智能算法, 它模拟了自然界中鸡群在等级制度下的觅食行为<sup>[7]</sup>. 整个鸡群由公鸡、母鸡和小鸡组成, 其中公鸡的搜索能力最强, 母鸡次之, 小鸡最弱. 等级制度是CSO算法区别于其他群体智能算法的一个显著特征, 表现为“公鸡带领母鸡搜索和小鸡跟随母鸡搜索”. CSO算法已被证实具有全局收敛性<sup>[8]</sup>, 并已成功应用于设施布局<sup>[9]</sup>、工程设计<sup>[10]</sup>、任务调度<sup>[11]</sup>、光伏系统功率控制<sup>[12]</sup>和天线阵列设计<sup>[13]</sup>等领域.

与其他群体智能算法一样, CSO算法也存在收敛速度慢、容易陷入局部最优等不足. 针对这些问题,

收稿日期: 2021-06-18; 录用日期: 2021-09-22.

基金项目: 国家自然科学基金项目(U1804262, 61702463); 中原千人计划项目(204200510003); 食管癌防治国家重点实验室开放基金项目(K2020-0010, K2020-0011).

责任编辑: 林崇.

<sup>†</sup>通讯作者. E-mail: yanfengwang@yeah.net.

学者们提出了许多改进策略. 杨菊靖等<sup>[14]</sup>在母鸡的搜索方程中加入最优解信息和权重因子; Wu等<sup>[15]</sup>利用子群公鸡信息和自学习因子改进小鸡的搜索方程; 黄霞等<sup>[16]</sup>设计自我、标榜、引领3种学习规则来引导公鸡、母鸡和小鸡的位置更新; 韩斐斐等<sup>[17]</sup>分别采用自适应、偏好随机游动、定向变异3种策略调整公鸡、母鸡和小鸡的搜索方程; 李宾等<sup>[18]</sup>在公鸡的搜索方程中加入全局和局部最优解信息, 利用吸引机制改进母鸡和小鸡的搜索方程; 童斌斌等<sup>[19]</sup>利用最优解信息改进公鸡和母鸡的搜索方程, 同时引入边界变异策略提升种群的多样性. 此外, 将CSO算法与其他智能算法融合也是一种常用的改进方法, 详见文献[20].

文献[20]进一步指出, 由于公鸡的引领作用, 当公鸡搜索陷入局部最优时, 母鸡和小鸡也会跟着陷入局部最优, 导致算法出现早熟收敛. 在群体智能算法中, 勘探(也称为全局探索)指搜索解空间未知区域发现优良解的过程, 开发(也称为局部开采)指利用已知优良解信息搜索优良解的过程<sup>[21]</sup>. 勘探和开发是一对互相矛盾的搜索过程, 平衡二者之间的关系是提高算法优化性能的一种有效措施. 本文充分发挥公鸡的引领优势, 从平衡勘探和开发的角度提出一种基于刺激-响应机制的鸡群算法(chicken swarm optimization based on stimulus-response mechanism, SRCSO). 新算法为公鸡设计勘探和开发两种搜索方程, 并根据种群聚集度和平均改进度定义公鸡执行不同搜索方程的环境刺激和响应阈值. 公鸡在刺激-响应机制下选择恰当的搜索方程, 从而带领鸡群完成勘探或者开发. 为验证所提出算法的性能, 将其用于基准函数优化和食管癌生存预测模型优化, 并与其他改进的鸡群算法进行对比.

## 1 基本CSO算法

CSO算法是由Meng等<sup>[7]</sup>于2014年提出的一种群体智能算法, 模拟了鸡群在等级制度下的觅食行为. CSO算法主要表现为“公鸡-母鸡-小鸡”等级制度的建立以及“母鸡围绕公鸡和小鸡围绕母鸡”觅食行为的模拟.

### 1.1 等级制度

在求解优化问题时, 鸡群中的每个个体都对应优化问题的一个解. 根据鸡群个体的适应度值确定鸡群的等级制度, 即适应度值排名靠前的个体设置为公鸡, 排在中间的个体设置为母鸡, 排名靠后的个体设置为小鸡. 根据公鸡个数将鸡群划分为若干子群, 每个子群都包含一只公鸡、若干母鸡和一些小鸡. 母鸡和小鸡随机加入子群, 并随机建立母子关系. 鸡群中

的等级制度、子群划分、母子关系等一旦建立就保持不变, 每隔一定迭代次数更新一次.

### 1.2 觅食行为

假设鸡群规模为 $N$ , 搜索空间维度为 $D$ , 个体位置 $x_{i,j}^t$  ( $i \in [1, 2, \dots, N], j \in [1, 2, \dots, D]$ )表示第 $i$ 个个体第 $j$ 维在第 $t$ 次迭代的值. 觅食过程中, 公鸡、母鸡和小鸡采用不同的搜索方程.

公鸡进行自主觅食, 采用的搜索方程为

$$x_{i,j}^{t+1} = x_{i,j}^t \times (1 + \text{randn}(0, \sigma^2)). \quad (1)$$

$$\begin{cases} \sigma^2 = \exp\left\{\frac{f_k - f_i}{|f_i| + \varepsilon}\right\}, & f_i \geq f_k; \\ \sigma^2 = 1, & f_i < f_k. \end{cases} \quad (2)$$

其中:  $\text{randn}(0, \sigma^2)$ 表示均值为0、方差为 $\sigma^2$ 的高斯分布随机数;  $\varepsilon$ 为一个很小的常数, 避免出现零除错误;  $f_i$ 和 $f_k$ 分别为第 $i$ 只公鸡和第 $k$ 只公鸡的适应度值, 满足 $i \neq k$ .

母鸡跟随公鸡觅食, 采用的搜索方程为

$$x_{i,j}^{t+1} = x_{i,j}^t + c_1 \times \text{rand}(x_{r_1,j}^t - x_{i,j}^t) + c_2 \times \text{rand}(x_{r_2,j}^t - x_{i,j}^t), \quad (3)$$

$$c_1 = \exp((f_i - f_{r_1}) / (\text{abs}(f_i) + \varepsilon)), \quad (4)$$

$$c_2 = \exp(f_{r_2} - f_i). \quad (5)$$

其中:  $c_1$ 、 $c_2$ 为学习因子;  $\text{rand}$ 为 $[-1, 1]$ 上均匀分布的随机数;  $r_1$ 为第 $i$ 只母鸡所在子群中的公鸡;  $r_2$ 为整个鸡群中随机选取的公鸡或母鸡, 满足 $r_1 \neq r_2 \neq i$ .

小鸡跟随母鸡觅食, 采用的搜索方程为

$$x_{i,j}^{t+1} = x_{i,j}^t + FL \times (x_{m,j}^t - x_{i,j}^t). \quad (6)$$

其中:  $m$ 为第 $i$ 只小鸡对应的母鸡;  $FL$ 为 $[0, 2]$ 上均匀分布的随机数.

## 2 SRCSO算法

### 2.1 算法改进思路

在基本CSO算法中, 公鸡自主觅食, 母鸡围绕公鸡觅食, 小鸡围绕母鸡觅食. 图1形象地描述了这一搜索过程<sup>[9]</sup>. 当公鸡搜索到局部最优后, 母鸡和小鸡也会跟着陷入局部最优. 对群体智能算法而言, 在搜索过程中平衡勘探和开发是提高算法性能的一种有效措施. 由图1可知, 鸡群的勘探和开发主要由公鸡带领完成. 因此, 分别为公鸡设计具有勘探能力和开发能力的搜索方程, 并通过恰当地选择机制协调二者之间的关系, 可以提高CSO算法的优化性能. 受此启发, 本文提出一种基于刺激-响应机制的鸡群算法SRCSO, 其核心思路如图2所示. SRCSO算法主要包含勘探-开发搜索和刺激-响应选择两个模块.

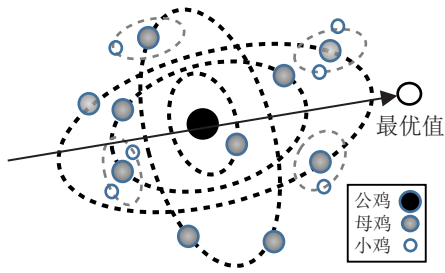


图1 鸡群寻优过程

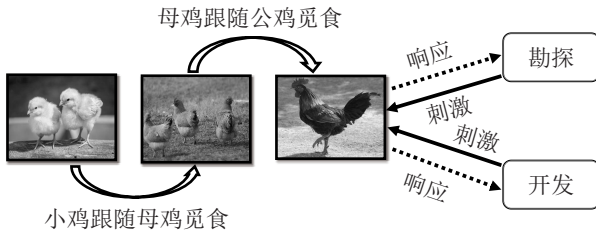


图2 SRCO算法核心思路

## 2.2 刺激-响应机制

刺激-响应机制是社会性昆虫中的一种任务选择机制,同时也是群体智能的一种表现形式.具体地,族群中每个任务都对应一个环境刺激,个体对于每个任务都存在一个响应阈值,个体以响应刺激的方式选择任务<sup>[22]</sup>.令 $S$ 表示任务的环境刺激, $\theta$ 表示个体的响应阈值,则个体选择任务的概率为

$$P = \frac{S^2}{S^2 + \theta^2}. \quad (7)$$

其中: $S$ 和 $\theta$ 分别反映了个体选择任务的外部驱动力和内部倾向性.当 $S > \theta$ 时,个体选择任务的概率高;反之,个体选择任务的概率低.在刺激-响应机制下,个体能够选择符合族群需求的恰当任务.目前,刺激-响应机制已被用于一些任务选择问题<sup>[23-24]</sup>.

## 2.3 公鸡搜索方式

### 2.3.1 搜索任务

由图2可知,公鸡需要执行勘探和开发两种不同的搜索任务.勘探指搜索解空间未知区域发现优良解的过程,开发指利用已知优良解信息搜索更优解的过程.下面将对公鸡的搜索方程进行改进,使其分别具有较强的勘探能力和开发能力.

在基本CSO算法中,公鸡的搜索本质上是一个高斯变异过程.与高斯变异相比,柯西变异会产生较大的搜索步长,具有较好的全局探索能力<sup>[25-26]</sup>.因此,将式(1)中的高斯分布 $\text{randn}(0, \sigma^2)$ 替换为柯西分布 $\text{cauchy}(0, \sigma^2)$ ,引导公鸡进行勘探搜索,有

$$x_{i,j}^{t+1} = x_{i,j}^t \times (1 + \text{cauchy}(0, \sigma^2)). \quad (8)$$

在群体智能算法中,将全局最优解信息引入搜索方程能有效提高算法的局部开采能力<sup>[27-28]</sup>.因此,将式(1)中的当前解 $x_{i,j}^t$ 替换为全局最优解 $x_{\text{best},j}^t$ ,引导

公鸡进行开发搜索,有

$$x_{i,j}^{t+1} = x_{\text{best},j}^t \times (1 + \text{randn}(0, \sigma^2)). \quad (9)$$

此外,惯性权重也是调节算法全局探索能力和局部开采能力的一种有效措施.惯性权重值较大时算法的勘探能力强,惯性权重值较小时算法的开发能力强<sup>[29-30]</sup>.在式(8)和(9)中分别引入较大和较小的惯性权重,进一步提高其勘探能力和开发能力.参考文献[31],惯性权重分别取值为0.9和0.4.

### 2.3.2 环境刺激

环境刺激描述了个体选择任务的外部驱动力,刺激越大任务越紧急.一般而言,群体智能算法的搜索状态与种群多样性有关<sup>[32]</sup>:当种群多样性差时,算法处于开发状态,需要通过勘探避免陷入局部最优;当种群多样性好时,算法处于勘探状态,需要通过开发提高搜索精度.因此,可基于种群多样性来设计勘探任务和开发任务的环境刺激,即多样性好时对开发的需求大,多样性差时对勘探的需求大.

方差反映了一组数据的离散度,可以在一定程度上度量种群多样性.令 $f_i$ 表示个体 $i$ 的适应度值, $\mu$ 表示鸡群的平均适应度值,则鸡群方差为 $v = \sum_{i=1}^N (f_i - \mu)^2 / N$ .为了降低种群规模、解空间、优化问题等对方差的影响,将多样性限定在 $[0,1]$ 内,定义如下种群聚集度:

$$a = \exp(-\sqrt{v}), \quad (10)$$

其中 $p$ 为缩放因子.由式(10)可知,方差小时聚集度大,对应的种群多样性差;方差大时聚集度小,对应的种群多样性好.

综上,勘探任务和开发任务的环境刺激分别为

$$S_e = a, \quad (11)$$

$$S_t = 1 - a. \quad (12)$$

### 2.3.3 响应阈值

响应阈值描述了个体选择任务的内部倾向性,阈值越小倾向性越大.在社会性昆虫中,个体阈值与其任务表现有关.如个体成功执行任务后对应的阈值会降低,再次执行任务的可能性会增加,从而保证族群效率.因此,可依据勘探和开发的搜索效率设计公鸡对勘探任务和开发任务的响应阈值,即勘探效率高时倾向于继续勘探,开发效率高时倾向于继续开发.

以最小化问题为例,定义解的改进度为

$$I = \begin{cases} f_{\text{old}} - f_{\text{new}}, & \text{solution updated successfully;} \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

其中:  $f_{old}$  为搜索前解的适应度值,  $f_{new}$  为搜索后解的适应度值. 令  $n$  表示搜索方程被选中的次数,  $I_q$  表示第  $q$  次时解的改进度, 则解的平均改进度为

$$A = \sum_{q=1}^n I_q/n. \quad (14)$$

令  $A_e$  和  $A_t$  分别表示公鸡进行勘探搜索和开发搜索时解的平均改进度, 其值越大表明搜索效率越高. 同理, 为避免式(7)中因刺激和阈值相差过大而导致选择失效, 将搜索效率限定在  $[0, 1]$  内.

综上, 公鸡对勘探任务和开发任务的响应阈值分别定义为

$$\theta_e = 1 - \frac{A_e}{A_e + A_t}, \quad (15)$$

$$\theta_t = 1 - \frac{A_t}{A_e + A_t}. \quad (16)$$

### 2.3.4 选择概率

在刺激-响应机制下, 公鸡选择勘探任务和开发任务的概率分别为

$$P_e = \frac{S_e^2}{S_e^2 + \theta_e^2}, \quad (17)$$

$$P_t = \frac{S_t^2}{S_t^2 + \theta_t^2}. \quad (18)$$

## 2.4 完整算法流程

基于上述描述, 本文提出的 SRCSO 算法流程详细步骤如下.

step 1: 初始化算法参数, 包括种群规模  $N$ 、公鸡数量  $N_R$ 、母鸡数量  $N_H$ 、小鸡数量  $N_C$ 、更新代数  $G$ 、函数最大评价次数 MAX\_FEs、搜索步长  $FL$  和缩放因子  $p$  等.

step 2: 初始化种群位置, 计算鸡群中每个个体的适应度值, 设置迭代次数  $t = 1$ .

step 3: 如果  $\text{mod}(t, G) = 1$ , 则对鸡群个体的适应度值进行排序, 进而确定鸡群的等级制度、子群划分和母子关系.

step 4: 根据式(11)和(12)计算勘探任务和开发任务的环境刺激.

step 5: 根据式(15)和(16)计算公鸡对应勘探任务和开发任务的响应阈值.

step 6: 根据式(17)和(18)计算公鸡执行勘探任务和开发任务的概率.

step 7: 公鸡根据概率大小选择搜索任务, 进行勘探搜索时按照式(8)觅食, 进行开发搜索时按照式(9)觅食, 并根据式(14)更新平均改进度.

step 8: 母鸡跟随公鸡按照式(3)觅食, 小鸡跟随母鸡按照式(6)觅食.

step 9: 记录全局最优值, 若满足终止条件, 则输

出最优解和最优值; 否则, 令  $t = t + 1$ , 返回 step 3.

## 2.5 时间复杂度分析

设  $N$  为种群规模,  $D$  为解空间维度,  $M$  为最大迭代次数. 在整个迭代过程中, 基本 CSO 算法的时间复杂度大约为

$$O(ND + ND + MN + MN(N-1)/(2G) + MND + MND) = O(2ND + MN(1 + (N-1)/(2G)) + 2MND).$$

与基本 CSO 算法相比, SRCSO 算法在迭代过程中主要引入了环境刺激、响应阈值和执行概率, 并将确定鸡群等级制度的更新代数  $G$  由 10 调整为 100. 在整个迭代过程中, SRCSO 算法的时间复杂度大约为

$$O(ND + ND + MN + MN(N-1)/20G + MN + 0.2MN + 0.2MN + MND + MND) = O(2ND + MN(2.4 + (N-1)/(20G)) + 2MND).$$

因此, SRCSO 算法与基本 CSO 算法的时间复杂度在一个数量级上, 二者相差较小.

## 3 基准函数优化

### 3.1 测试函数

为了测试 SRCSO 算法的性能, 选取 15 个基准函数进行仿真实验. 表 1 给出了测试函数的信息, 包括函数名称、搜索范围、最优值和函数类型. 其中:  $U$  表示单峰函数,  $M$  表示多峰函数,  $S$  表示可分离函数,  $N$  表示不可分离函数. 实验测试分别对函数在 30 维和 100 维下进行仿真, 仿真环境为 Windows 10 操作系统和 Matlab R2017b 编程语言.

表 1 实验中使用的 15 个测试函数

编号	函数名称	搜索范围	最优值	类型
$f_1$	Sphere	$[-100, 100]^D$	0	US
$f_2$	Elliptic	$[-100, 100]^D$	0	UN
$f_3$	SumSquare	$[-10, 10]^D$	0	US
$f_4$	SumPower	$[-1, 1]^D$	0	US
$f_5$	Schwefel 2.22	$[-10, 10]^D$	0	UN
$f_6$	Schwefel 2.21	$[-100, 100]^D$	0	UN
$f_7$	Step	$[-100, 100]^D$	0	US
$f_8$	Rosenbrock	$[-5, 10]^D$	0	UN
$f_9$	Quartic	$[-1.28, 1.28]^D$	0	US
$f_{10}$	Penalized1	$[-100, 100]^D$	0	MN
$f_{11}$	Rastrigin	$[-5.12, 5.12]^D$	0	MS
$f_{12}$	Penalize2	$[-50, 50]^D$	0	MN
$f_{13}$	Ackley	$[-50, 50]^D$	0	MN
$f_{14}$	Alpine	$[-10, 10]^D$	0	MS
$f_{15}$	Levy	$[-10, 10]^D$	0	MN

### 3.2 与基本 CSO 及其改进算法的比较

将 SRCSO 算法与基本 CSO<sup>[7]</sup> 及其改进算法 ICSO-1<sup>[15]</sup>、HCSO<sup>[16]</sup>、ECSO<sup>[17]</sup>、ICSO-2<sup>[18]</sup>、ISCSO<sup>[19]</sup>

进行比较,对比算法参数按照原文献设置. SRCSO算法的参数设置为:  $N = 100, G = 100, N_R = 0.2N, N_H = 0.6N, N_C = N - N_R - N_H, FL = \text{rand}(0.4, 0.9), w_{\max} = 0.9, w_{\min} = 0.4, p = 80$ , 其中  $G$  和  $p$  通过调整参数的对比实验确定. 为了保证比较的公平性, 每个算法在所有函数上独立运行 30 次, 终止条件设置为函数评价次数达到  $1000 \times D$ .

表 2 给出了函数在 30 维下的测试结果, 包括基于平均值 (Mean) 和标准差 (Std) 的解质量对比和基于非参数检验的综合对比. 在解质量对比中, “+” “-” “=” 分别表示 SRCSO 算法的优结果优于、劣于、相当于对比算法. 由表 2 可见, 与 CSO、ICSO-

1、ECSO 和 ICSO 相比, SRCSO 在 14 个函数上取得了更好的解, 在剩余 1 个函数上取得了相同的解. SRCSO 在 13 个函数上优于 HCSO, 在剩余 2 个函数上等于 HCSO. SRCSO 和 ICSO-2 在 3 个函数上找到了相同的解, 在剩余 12 个函数上 SRCSO 的优化效果更好. 综合对比选取常用的 Friedman 检验和 Wilcoxon 检验. Friedman 检验时所有算法的排序结果为  $\text{SRCSO} < \text{HCSO} < \text{ECSO} < \text{ICSO-2} < \text{ICSO} < \text{ICSO-1} < \text{CSO}$ , 排序值越小表明算法的整体优化性能越好. Wilcoxon 检验时显著性水平设置为 0.05, 6 个对比算法的  $p$  值均小于 0.05, 表明 SRCSO 与这些算法之间存在显著性差异, 即 SRCSO 优于这些算法.

表 2 SRLDABC 与各个算法的比较 ( $D = 30$ )

函数	指标	CSO	ICSO-1	HCSO	ECSO	ICSO-2	ISCSO	SRCSO
$f_1$	Mean	1.97e-12	1.62e-15	4.99e-182	3.29e-16	2.48e-121	2.48e-15	3.74e-308
	Std	4.20e-12	3.26e-15	0.00e+00	7.48e-16	1.13e-120	5.54e-15	0.00e+00
$f_2$	Mean	3.91e-10	3.24e-11	4.75e-176	4.83e-14	4.93e-55	9.23e-13	3.27e-297
	Std	8.82e-10	4.09e-11	0.00e+00	1.03e-13	2.70e-54	1.60e-12	0.00e+00
$f_3$	Mean	9.94e-13	7.53e-16	6.89e-182	4.68e-17	2.39e-124	9.12e-16	1.45e-305
	Std	3.11e-12	6.15e-16	0.00e+00	7.69e-17	1.31e-123	2.17e-15	0.00e+00
$f_4$	Mean	1.53e-26	2.36e-40	9.74e-198	1.85e-49	1.33e-157	4.75e-56	0.00e+00
	Std	8.38e-26	8.50e-40	0.00e+00	1.01e-48	7.29e-157	1.72e-55	0.00e+00
$f_5$	Mean	3.02e-11	2.79e-11	1.32e-92	7.72e-13	7.55e+00	1.79e-10	1.01e-160
	Std	2.92e-11	3.31e-11	1.09e-92	1.14e-12	1.53e+01	2.70e-10	3.47e-160
$f_6$	Mean	1.33e+01	2.39e-05	1.77e-92	4.13e-07	5.04e-03	7.60e-01	2.35e-146
	Std	5.48e+01	1.17e-05	4.28e-93	3.86e-07	2.76e-02	9.01e-01	1.18e-145
$f_7$	Mean	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
	Std	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
$f_8$	Mean	3.45e+01	2.75e+01	2.87e+01	2.76e+01	2.89e+01	2.72e+01	5.21e+00
	Std	3.66e+01	4.62e-01	8.69e-02	3.84e-01	5.35e-02	3.44e-01	7.76e+01
$f_9$	Mean	1.65e-02	1.54e-03	3.75e-04	6.85e-04	4.76e-03	3.48e-03	1.23e-04
	Std	1.22e-02	5.19e-04	3.79e-04	3.37e-04	5.75e-03	1.94e-03	7.99e-05
$f_{10}$	Mean	1.37e+00	6.92e-02	5.96e-01	1.18e-01	1.34e+00	2.75e-01	5.00e-03
	Std	3.91e+00	2.22e-02	5.96e-02	2.96e-02	2.12e-01	9.29e-01	9.69e-03
$f_{11}$	Mean	3.72e-13	1.28e-13	0.00e+00	7.10e-16	0.00e+00	2.31e-13	0.00e+00
	Std	6.76e-13	7.13e-13	0.00e+00	2.94e-15	0.00e+00	6.21e-13	0.00e+00
$f_{12}$	Mean	1.81e+00	1.25e+00	3.07e+00	1.63e+00	4.75e+00	8.59e-01	5.83e-02
	Std	8.16e-01	2.02e-01	1.95e-01	3.47e-01	1.35e+00	2.89e-01	1.01e-01
$f_{13}$	Mean	2.74e-07	9.74e-09	4.20e-15	9.13e-10	8.88e-16	3.55e-08	8.88e-16
	Std	1.93e-07	4.34e-09	9.01e-16	1.26e-09	0.00e+00	4.62e-08	0.00e+00
$f_{14}$	Mean	2.88e-04	1.11e-10	6.97e-93	3.66e-13	9.69e-69	4.12e-02	2.69e-131
	Std	1.58e-03	1.62e-10	1.59e-92	4.55e-13	4.98e-68	1.49e-01	1.45e-130
$f_{15}$	Mean	1.29e+01	4.06e+01	2.39e-01	1.45e+01	4.07e+01	5.21e+00	7.64e-02
	Std	2.17e+01	3.67e+01	2.01e+00	3.21e+01	3.67e+01	1.99e+00	1.45e-01
+ / = / -		14 / 1 / 0	14 / 1 / 0	13 / 2 / 0	14 / 1 / 0	12 / 3 / 0	14 / 1 / 0	-
Friedman mean ranking		5.74	4.61	3.38	4.09	4.29	4.53	1.35
Wilcoxon $p_{\text{value}}$		2.06e-06	4.29e-06	6.91e-05	4.13e-06	1.16e-05	7.93e-06	-

表 3 给出了函数在 100 维下的测试结果. 由表 3 可见, 函数维度由 30 增到 100 后, 对 SRCSO 与其他算法的性能对比影响不大. 在解质量对比方面, SRCSO 在 15 个函数上均优于 CSO 和 ISCSO; 与 ICSO-1 和 ECSO 相比, SRCSO 在 13 个函数上找到了更好的解, 在剩余 2 个函数上找到了相同的解; SRCSO 分别在 8、6、1 个函数上优于、等于、劣于 HCSO; SRCSO

在 8 个函数上优于 ICSO-2, ICSO-2 仅在函数  $f_{14}$  上优于 SRCSO, 在剩余 6 个函数上二者优化结果相同. 在综合对比方面, Friedman 检验时所有算法的排序结果为  $\text{SRCSO} < \text{HCSO} < \text{ICSO-2} < \text{ECSO} < \text{ICSO-1} < \text{ISCSO} < \text{CSO}$ , Wilcoxon 检验时 6 个对比算法的  $p$  值均小于 0.05, 表明 SRCSO 具有较好的优化性能.

表3 SRCSO与各个算法的比较( $D = 100$ )

函数	指标	CSO	ICSO-1	HCSO	ECSO	ICSO-2	ISCSO	SRCSO
$f_1$	Mean	1.24e+03	1.89e-37	0.00e+00	4.86e-37	0.00e+00	3.11e-01	0.00e+00
	Std	3.97e+03	1.64e-37	0.00e+00	6.34e-37	0.00e+00	1.70e+00	0.00e+00
$f_2$	Mean	9.62e-02	2.59e-34	0.00e+00	1.16e-34	0.00e+00	9.29e-03	0.00e+00
	Std	2.88e+03	2.68e-34	0.00e+00	3.44e-34	0.00e+00	3.66e-02	0.00e+00
$f_3$	Mean	1.16e+01	3.38e-40	0.00e+00	9.66e-38	0.00e+00	6.91e-03	0.00e+00
	Std	2.08e+01	4.40e-40	0.00e+00	1.98e-37	0.00e+00	3.77e-02	0.00e+00
$f_4$	Mean	3.69e-13	4.75e-92	0.00e+00	1.23e-189	0.00e+00	9.18e-19	0.00e+00
	Std	1.16e-12	1.37e-91	0.00e+00	0.00e+00	0.00e+00	3.45e-18	0.00e+00
$f_5$	Mean	6.81e-16	2.21e-21	9.18e-312	1.40e-27	3.34e+01	2.79e-13	0.00e+00
	Std	7.11e-16	1.32e-21	3.15e-311	5.89e-21	6.80e+01	2.99e-13	0.00e+00
$f_6$	Mean	3.44e+01	1.77e-18	7.47e-312	2.13e-18	1.94e+00	8.29e-01	0.00e+00
	Std	3.77e+00	8.54e-19	0.00e+00	1.71e-18	1.06e+01	7.57e-01	0.00e+00
$f_7$	Mean	1.70e+02	0.00e+00	0.00e+00	0.00e+00	9.12e+03	4.23e+01	0.00e+00
	Std	2.64e+02	0.00e+00	0.00e+00	0.00e+00	2.08e+03	7.31e+01	0.00e+00
$f_8$	Mean	1.19e+05	9.84e+01	9.87e+01	9.81e+01	9.84e+01	1.26e+03	4.21e+00
	Std	6.24e+04	4.75e-01	1.10e-01	5.76e-01	6.41e-02	6.66e+02	5.92e+00
$f_9$	Mean	1.47e+01	5.85e-04	8.95e-05	3.86e-04	1.53e-03	4.59e-01	3.43e-05
	Std	8.73e+01	1.81e-04	6.11e-05	2.03e-04	1.77e-03	1.59e-01	3.24e-05
$f_{10}$	Mean	8.59e+05	5.37e-01	8.79e-01	4.04e-01	1.15e+00	5.84e+00	4.17e-04
	Std	1.02e+06	5.70e-02	3.17e-02	3.84e-02	7.14e-02	3.45e+00	5.79e-04
$f_{11}$	Mean	1.81e-04	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.03e+00	0.00e+00
	Std	8.14e-04	0.00e+00	0.00e+00	0.00e+00	0.00e+00	5.69e+00	0.00e+00
$f_{12}$	Mean	1.62e+07	9.33e+00	1.11e+01	9.78e+00	1.81e+01	2.67e+02	2.62e-02
	Std	1.62e+07	3.98e-01	5.87e-01	4.64e-01	3.74e+00	5.54e+02	6.61e-02
$f_{13}$	Mean	4.13e+00	4.44e-15	4.44e-15	4.44e-15	8.88e-16	1.95e+00	8.88e-16
	Std	3.54e+00	0.00e+00	0.00e+00	6.48e-16	0.00e+00	2.16e+00	0.00e+00
$f_{14}$	Mean	9.23e-02	2.79e-21	6.99e-312	3.86e-30	1.03e-273	1.23e-02	7.39e-177
	Std	7.43e-02	8.05e-21	0.00e+00	6.71e-30	0.00e+00	3.71e-02	0.00e+00
$f_{15}$	Mean	8.68e+01	7.74e+01	9.78e+01	9.43e+01	1.24e+02	6.79e+01	1.11e-01
	Std	7.12e+00	3.81e+01	1.01e+00	2.73e+01	1.43e+02	1.20e+01	2.31e-01
+ / = / -		15 / 0 / 0	13 / 2 / 0	8 / 6 / 1	13 / 2 / 0	8 / 6 / 1	15 / 0 / 0	-
Friedman mean ranking		6.66	4.08	3.10	3.65	3.32	5.65	1.54
Wilcoxon $p_{value}$		2.27e-06	4.36e-06	1.29e-04	7.98e-06	8.81e-05	5.00e-06	-

为了更直观地显示本文算法的收敛速度,图3给出了各算法在 $f_1$ 、 $f_6$ 、 $f_{10}$ 和 $f_{14}$ 四个函数上的收敛曲线,其中横轴为函数评价次数,纵轴为最优适应值。

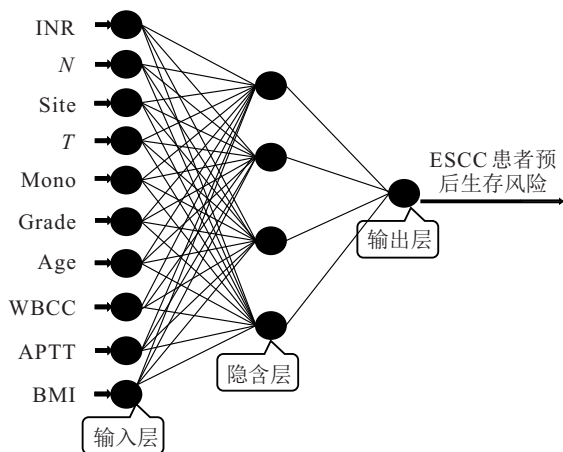


图3 BP神经网络生存风险预测模型

由图3可见,对于不同类型的测试函数, SRCSO均具有较好的收敛速度和收敛精度. 具体而言,对于函数 $f_1(D = 100)$ , HCSO、ICSO-2和SRCSO均找到了理论最优值,但SRCSO的收敛速度最快. 对于函数 $f_{14}(D = 100)$ , SRCSO在前期收敛速度最快,在后期

陷入局部最优,收敛精度仅次于HCSO和ICSO-2. 对于图3中其他6种情形, SRCSO均表现出更快的收敛速度和更高的收敛精度。

### 4 食管癌生存风险预测模型优化

为了检验 SRCSO 算法在实际问题中的优化效果,将其用于食管癌生存风险预测模型优化. 首先基于BP神经网络建立食管癌生存风险预测模型,然后利用 SRCSO 算法优化神经网络的初始权值和阈值,并与其他改进CSO算法进行对比。

#### 4.1 BP神经网络生存风险预测模型

BP神经网络是一种高效的机器学习算法,目前已广泛应用于癌症生存风险预测<sup>[33-34]</sup>. 本节以河南省食管癌重点开放实验室收集的2007年~2008年261例食管癌患者的病例资料为基础,利用BP神经网络建立食管癌生存风险预测模型. 在所有患者中,男性162例,女性99例,年龄区间38~82岁,平均年龄60岁. 通过FCM聚类从50个特征变量中筛选10个作为输入,采用“试错法”确定隐含层神经元个数,选取预后生存风险为输出,建立如图4所示的BP神经网络生存风险预测模型。

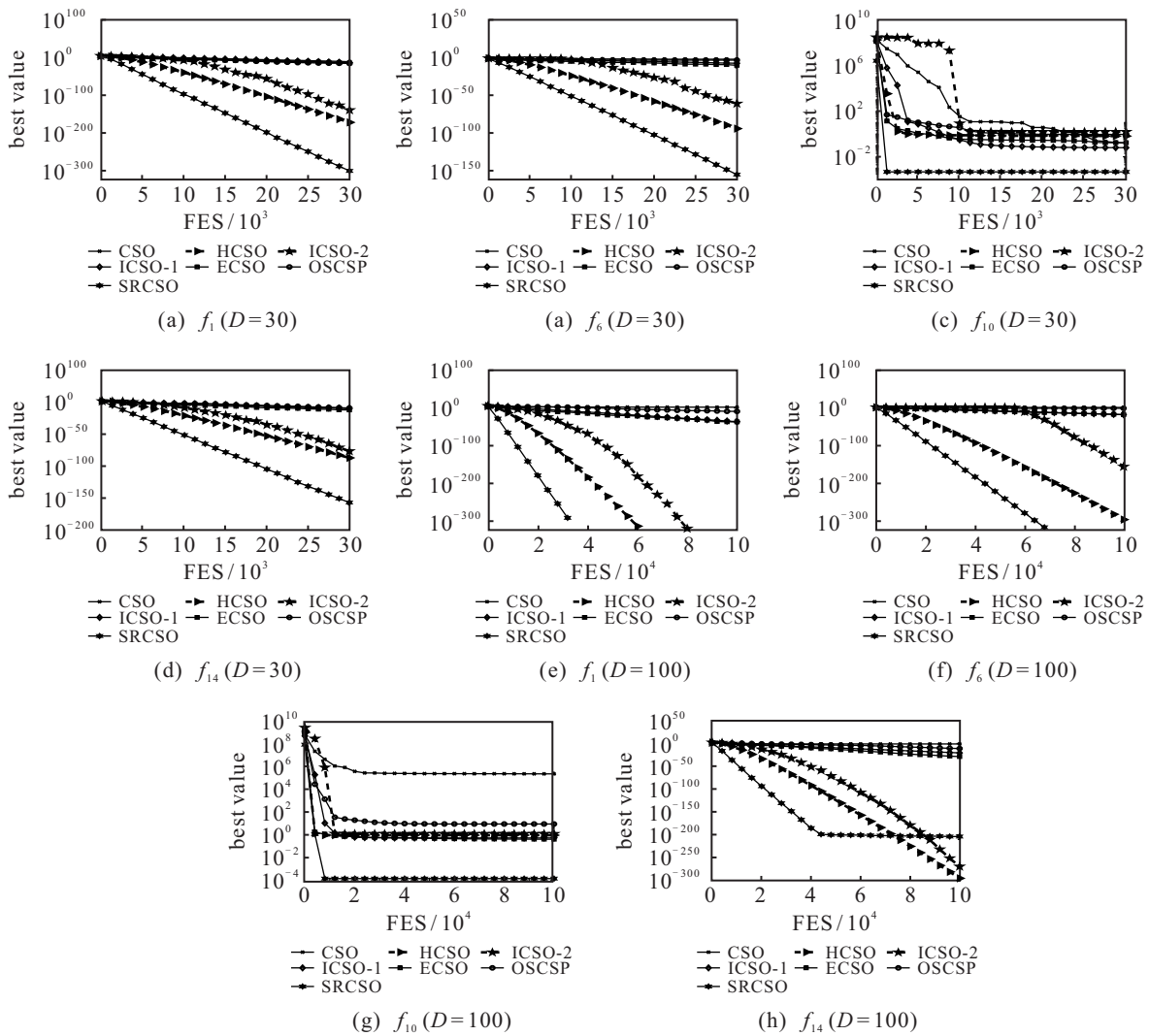


图4 函数收敛曲线

上述模型待优化的参数为权值和阈值,其中权值是指神经元之间的连接权值,阈值是指隐含层和输出层神经元的节点阈值. 本文所建立BP神经网络生存风险预测模型的结构是10-4-1,待优化参数维度为

$$\text{Dim} = \text{input} \cdot \text{hid} + \text{hid} \cdot \text{output} + \text{hid} + \text{output}, \quad (19)$$

其中input、hid、output分别为BP神经网络输入层、隐含层和输出层的神经元个数,即本实验有 $10 \times 4 + 4 \times 1 = 44$ 个权值和 $4 + 1 = 5$ 个阈值需要优化. 以食管癌患者预后生存风险分类正确率为目标函数,优化问题定义如下:

$$\max f(w, v) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}. \quad (20)$$

其中:  $w$ 、 $v$ 分别为权值和阈值,TP表示实际为低风险且预测也是低风险的样本个数,FP表示实际为高风险而预测是低风险的样本个数,FN表示实际为低风险而预测是高风险的样本个数,TN表示实际为高风险且预测也是高风险的样本个数.

## 4.2 结果分析

对数据进行归一化处理,随机选取207例样本作为训练集,54例样本作为测试集. 将SRCSO算法与6种CSO算法进行比较,设置所有算法的种群规模 $N = 40$ ,函数评价次数 $\text{MAX\_FEs} = 1000 \times 49$ ,各算法参数与第4.1节一致. 在网络训练过程中,学习速率和训练最小误差均设置为0.01,最大训练次数设置为1000.

图5给出了7种CSO算法独立优化预测模型10次的结果分布. 由图5可知,SRCSO算法的预测正确率从最大值、最小值、中位数及上下四分位数5个方面均优于其他CSO算法. 除了正确率,敏感性和特异性也经常用来评价预测效果. 敏感性计算公式为 $\text{Sensitivity} = \text{TP}/(\text{TP} + \text{FN})$ ,代表从低风险样本中准确分类出低风险患者的能力. 特异性计算公式为 $\text{Specificity} = \text{TN}/(\text{TN} + \text{FP})$ ,代表从高风险样本中准确分类出高风险患者的能力. 表4进一步从正

准确率、敏感性和特异性3个方面给出了10次预测结果的平均值,图6绘制了相应的对比情况. 从上述对比结果可以看出,SRCSO算法的预测效果最好,表明SRCSO算法的优化性能优于其他算法.

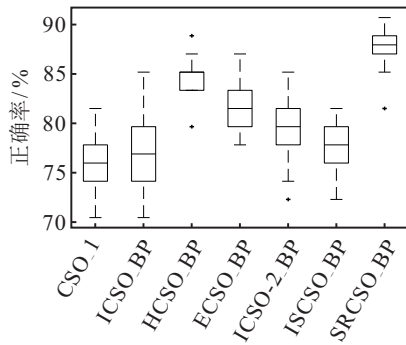


图5 7种CSO\_BP预测模型结果分布

表4 7种CSO\_BP预测模型平均结果对比 %

模型	正确率 (Accuracy)	敏感性 (Sensitivity)	特异性 (Specificity)
CSO_BP	75.74	76.94	73.33
ICSO-1_BP	77.22	80.28	71.11
HCSO_BP	84.63	86.67	80.56
ECSO_BP	81.85	84.44	76.67
ICSO-2_BP	79.26	85.28	67.78
ISCSO_BP	77.59	81.94	81.94
SRCSO_BP	87.59	90.28	82.22

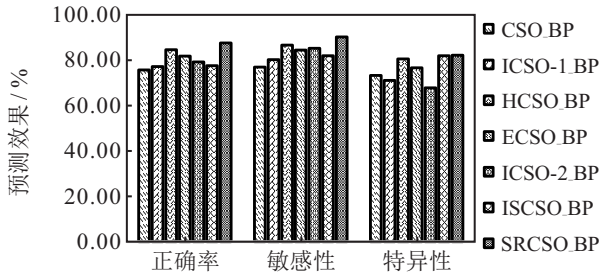


图6 7种CSO\_BP预测模型平均结果对比

## 5 结论

作为一种新型的群体智能算法,鸡群算法具有独特的等级制度. 勘探与开发是一对互相矛盾的搜索过程,平衡二者之间的关系是提高群体智能算法优化性能的一种有效措施. 本文将二者相结合,提出一种新的改进思路:充分发挥公鸡的引领作用,带领鸡群完成勘探与开发. 将勘探和开发看成两种不同的搜索任务,采用刺激-响应任务选择机制指导公鸡进行搜索,提出一种基于刺激-响应机制的鸡群算法. 在测试函数上的对比实验表明,所提出算法优于其他改进鸡群算法. 最后,在食管癌生存风险预测问题上进一步验证了所提出算法的有效性.

在发挥公鸡引领作用方面,本文未考虑改进母鸡和小鸡的搜索方程,后续研究可从以下3个方面展开: 1) 加强母鸡和小鸡的跟随搜索; 2) 研究激发-抑制

任务选择机制与鸡群算法的融合; 3) 研究改进鸡群算法在更多实际问题中的应用.

## 参考文献(References)

- [1] Dorigo M, Blum C. Ant colony optimization theory: A survey[J]. Theoretical Computer Science, 2005, 344(2/3): 243-278.
- [2] Poli R, Kennedy J, Blackwell T. Particle swarm optimization[J]. Swarm Intelligence, 2007, 1(1): 33-57.
- [3] Karaboga D. An idea based on honey bee swarm for numerical optimization[D]. Kayseri: Erciyes University, 2005.
- [4] Li X, Shao Z, Qian J. An optimizing method based on autonomous animats: Fish-swarm algorithm[J]. Systems Engineering Theory and Practice, 2002, 22(11): 32-38.
- [5] 吴虎胜, 张凤鸣, 吴庐山. 一种新的群体智能算法——狼群算法[J]. 系统工程与电子技术, 2013, 35(11): 2430-2438.  
(Wu H S, Zhang F M, Wu L S. New swarm intelligence algorithm—Wolf pack algorithm[J]. Systems Engineering and Electronics, 2013, 35(11): 2430-2438.)
- [6] Yang X S. Nature-inspired metaheuristic algorithms[M]. London: Luniver Press, 2008: 81-89.
- [7] Meng X B, Liu Y, Gao X Z, et al. A new bio-inspired algorithm: Chicken swarm optimization[C]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014: 86-94.
- [8] 吴定会, 孔飞, 纪志成. 鸡群算法的收敛性分析[J]. 中南大学学报: 自然科学版, 2017, 48(8): 2105-2112.  
(Wu D H, Kong F, Ji Z C. Convergence analysis of chicken swarm optimization algorithm[J]. Journal of Central South University: Science and Technology, 2017, 48(8): 2105-2112.)
- [9] 刘俊琦, 张则强, 王沙沙, 等. 考虑不规则物流交互点的过道布置问题建模与优化[J]. 计算机集成制造系统, 2021, 27(4): 1155-1166.  
(Liu J Q, Zhang Z Q, Wang S S, et al. Modeling and optimization of corridor allocation problem considering irregular logistics interaction points[J]. Computer Integrated Manufacturing Systems, 2021, 27(4): 1155-1166.)
- [10] Wang J Q, Cheng Z W, Ersoy O K, et al. Improvement and application of chicken swarm optimization for constrained optimization[J]. IEEE Access, 2019, 7: 58053-58072.
- [11] Torabi S, Safi-Esfahani F. A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing[J]. The Journal of Supercomputing, 2018, 74(6): 2581-2626.
- [12] 吴忠强, 于丹琦, 康晓华. 改进鸡群算法在光伏系统MPPT中的应用[J]. 太阳能学报, 2019, 40(6): 1589-1598.  
(Wu Z Q, Yu D Q, Kang X H. Application of improved chicken swarm optimization for MPPT in photovoltaic

- system[J]. *Acta Energiæ Solaris Sinica*, 2019, 40(6): 1589-1598.)
- [13] Liang S, Feng T, Sun G. Sidelobe-level suppression for linear and circular antenna arrays via the cuckoo search-chicken swarm optimisation algorithm[J]. *IET Microwaves, Antennas & Propagation*, 2017, 11(2): 209-218.
- [14] 杨菊蜻, 张达敏, 何锐亮, 等. 基于Powell搜索的混沌鸡群优化算法[J]. *微电子学与计算机*, 2018, 35(7): 78-82.  
(Yang J Q, Zhang D M, He R L, et al. A chaotic chicken optimization algorithm based on Powell search[J]. *Microelectronics & Computer*, 2018, 35(7): 78-82.)
- [15] Wu D H, Xu S P, Kong F. Convergence analysis and improvement of the chicken swarm optimization algorithm[J]. *IEEE Access*, 2016, 4: 9400-9412.
- [16] 黄霞, 叶春明, 郑军. 混合改进搜索策略的鸡群优化算法[J]. *计算机工程与应用*, 2018, 54(7): 176-181.  
(Huang X, Ye C M, Zheng J. Chicken swarm optimization algorithm of hybrid evolutionary searching strategy[J]. *Computer Engineering and Applications*, 2018, 54(7): 176-181.)
- [17] 韩斐斐, 赵齐辉, 杜兆宏, 等. 全局优化的改进鸡群算法[J]. *计算机应用研究*, 2019, 36(8): 2317-2319.  
(Han F F, Zhao Q H, Du Z H, et al. Enhanced chicken swarm algorithm for global optimization[J]. *Application Research of Computers*, 2019, 36(8): 2317-2319.)
- [18] 李宾, 申国君, 孙庚, 等. 改进的鸡群优化算法[J]. *吉林大学学报: 工学版*, 2019, 49(4): 1339-1344.  
(Li B, Shen G J, Sun G, et al. Improved chicken swarm optimization algorithm[J]. *Journal of Jilin University: Engineering and Technology Edition*, 2019, 49(4): 1339-1344.)
- [19] 童斌斌, 周晓南, 何庆. 基于信息交互的改进鸡群优化算法[J]. *贵州大学学报: 自然科学版*, 2021, 38(1): 58-64.  
(Tong B B, Zhou X N, He Q. Improved chicken swarm optimization based on information sharing[J]. *Journal of Guizhou University: Natural Sciences*, 2021, 38(1): 58-64.)
- [20] Deb S, Gao X Z, Tammi K, et al. Recent studies on chicken swarm optimization algorithm: A review[J]. *Artificial Intelligence Review*, 2020, 53(3): 1737-1765.
- [21] Singh A, Deep K. Exploration-exploitation balance in artificial bee colony algorithm: A critical analysis[J]. *Soft Computing*, 2019, 23(19): 9525-9536.
- [22] Bonabeau E, Theraulaz G, Deneubourg J. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies[J]. *Proceedings of the Royal Society of London—Series B: Biological Sciences*, 1996, 263(1376): 1565-1569.
- [23] Kim M H, Baik H, Lee S. Response threshold model based UAV search planning and task allocation[J]. *Journal of Intelligent & Robotic Systems*, 2014, 75(3/4): 625-640.
- [24] Wu H S, Li H, Xiao R B, et al. Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm[J]. *Physica A: Statistical Mechanics and Its Applications*, 2018, 491: 127-141.
- [25] 郭一楠, 刘丹丹, 程健, 等. 自适应混合变异文化算法[J]. *电子学报*, 2011, 39(8): 1913-1918.  
(Guo Y N, Liu D D, Cheng J, et al. Adaptive cultural algorithm adopting mixed mutation[J]. *Acta Electronica Sinica*, 2011, 39(8): 1913-1918.)
- [26] 罗平, 姚立海, 杨仕友, 等. 一种改进的粒子群优化算法[J]. *江南大学学报: 自然科学版*, 2007(5): 505-509.  
(Luo P, Yao L H, Yang S Y, et al. An improved particle swarm optimization algorithm[J]. *Journal of Jiangnan University: Natural Science Edition*, 2007(5): 505-509.)
- [27] Zhu G P, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. *Applied Mathematics and Computation*, 2010, 217(7): 3166-3173.
- [28] Cao Y C, Lu Y, Pan X Q, et al. An improved global best guided artificial bee colony algorithm for continuous optimization problems[J]. *Cluster Computing*, 2019, 22(2): 3011-3019.
- [29] Zhou H, Li Y. Cuckoo search algorithm with dynamic inertia weight[J]. *CAAI Transactions on Intelligent Systems*, 2015, 10(4): 645-651.
- [30] Eberhart R C, Shi Y H. Tracking and optimizing dynamic systems with particle swarms[C]. *Proceedings of the 2001 Congress on Evolutionary Computation*. Seoul, 2001: 94-100.
- [31] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[C]. *Proceedings of the 1999 Congress on Evolutionary Computation*. Washington DC, 1999: 1945-1950.
- [32] Li H J, Zhang P W, Guo H D. Adaptive multi-objective particle swarm optimization algorithm based on population Manhattan distance[J]. *Computer Integrated Manufacturing Systems*, 2020, 26(4): 1019-1032.
- [33] Zhang B, Liang X L, Gao H Y, et al. Models of logistic regression analysis, support vector machine, and back-propagation neural network based on serum tumor markers in colorectal cancer diagnosis[J]. *Genetics and Molecular Research: GMR*, 2016, 15(2): 10.4238.
- [34] Senthil S, Ayshwarya B. Lung cancer prediction using feed forward back propagation neural networks with optimal features[J]. *International Journal of Applied Engineering Research*, 2018, 13(1): 318-325.

### 作者简介

王英聪(1987—), 男, 讲师, 博士, 从事群智能、布局优化等研究, E-mail: ying\_cong\_wang@163.com;

刘驰(1995—), 男, 硕士生, 从事群智能的研究, E-mail: 15690848185@163.com;

王延峰(1973—), 男, 教授, 博士, 从事生物启发计算、生物信息处理、复杂系统建模等研究, E-mail: yanfengwang@yeah.net.

(责任编辑: 郑晓蕾)