

# 改进的鸡群优化算法

李 宾<sup>1</sup>, 申国君<sup>2</sup>, 孙 庚<sup>2,3</sup>, 郑婷婷<sup>2</sup>

(1. 吉林大学 数学学院, 长春 130012; 2. 吉林大学 计算机科学与技术学院, 长春 130012; 3. 吉林大学 通信工程学院, 长春 130012)

**摘 要:** 针对鸡群优化算法中解的更新效率较低且缺乏探索性等问题, 提出了一种改进的鸡群优化算法。该算法基于标准鸡群优化算法的种群分组更新机制, 并借鉴狼群优化算法和粒子群优化算法的思想, 引入改进因子和去重操作算子分别用以增强算法的寻优能力和提高种群的多样性。通过与其他4种算法在CEC 2014测试函数集上进行比较, 结果表明本文算法在绝大多数测试函数上均表现出了良好的优化效果, 在求解精度及收敛速度方面也优于其他算法。

**关键词:** 计算机应用; 鸡群优化算法; 收敛速度; 功能优化

**中图分类号:** TP301 **文献标志码:** A **文章编号:** 1671-5497(2019)04-1339-06

**DOI:** 10.13229/j.cnki.jdxbgxb20181268

## Improved chicken swarm optimization algorithm

LI Bin<sup>1</sup>, SHEN Guo-jun<sup>2</sup>, SUN Geng<sup>2,3</sup>, ZHENG Ting-ting<sup>2</sup>

(1. College of Mathematics, Jilin University, Changchun 130012, China; 2. College of Computer Science and Technology, Jilin University, Changchun 130012, China; 3. College of Communication Engineering, Jilin University, Changchun 130012, China)

**Abstract:** Based on the hierarchy mechanism of the conventional chicken swarm optimization (CSO) algorithm, an improved chicken swarm optimization (ICSO) algorithm is proposed to enhance the solution accuracy and the convergence rate of the conventional CSO algorithm. The ICSO algorithm introduces several improved factors that learned from the grey wolf optimizer (GWO) and the particle swarm optimization (PSO), to extend the searching ability of the algorithm. Moreover, a duplicate remove operator is also introduced to improve the diversity of the population. Experimental results show that the accuracy of the solution and the convergence rate of the proposed algorithm are better than other benchmark algorithms.

**Key words:** computer applications; chicken swarm optimization (CSO) algorithm; convergence rate; function optimization

收稿日期: 2018-08-12.

基金项目: 国家自然科学基金项目(61872158); 博士后创新人才支持计划项目; 中国博士后科学基金项目(2018M640283).

作者简介: 李宾(1960-), 女, 副教授. 研究方向: 群智能优化算法分析与设计. E-mail: lb@jlu.edu.cn

通信作者: 孙庚(1988-), 男, 讲师, 博士. 研究方向: 协作通信与智能计算. E-mail: sungeng@jlu.edu.cn

## 0 引言

群智能优化算法是基于自然界优胜劣汰法则和各种生物群体的特殊行为而提出的一种新兴的仿生计算方法。在2014年提出的鸡群优化(Chicken swarm optimization, CSO)算法<sup>[1]</sup>是群智能优化算法的一种,该算法从鸡群的觅食行为中得到启发,通过对种群进行分组更新的方式取得了良好的优化效果,并已经在一些领域有所应用。Banerjee等<sup>[2]</sup>利用CSO算法对串行级联卷积Turbo码进行了改进。Hafez等<sup>[3]</sup>将CSO算法作为评估函数的一部分,提出了一种特征选择系统并将其应用于数据集的特征选择。Chen等<sup>[4]</sup>基于CSO算法提出了一种无线传感器网络节点定位算法用以提高统计精度。Mu等<sup>[5]</sup>利用CSO算法对机器人的移动轨迹进行优化,取得了较好的效果。Roslina等<sup>[6]</sup>基于CSO算法对自适应神经模糊推理系统的参数进行训练,成功地提高了该系统的性能。Yi等<sup>[7]</sup>利用CSO算法对通信系统中载波的初始相位进行了优化,进而降低了峰均功率比,提高了误码性能。

然而,在一些实际优化问题中,由于CSO算法的解更新方式较为简单,导致求解过程易陷入局部最优,进而影响算法的求解精度和收敛速度。因此,如何对CSO算法进行改进使其具有更好的优化性能成为了该领域的研究热点。Wu等<sup>[8]</sup>引入交叉算子对CSO算法进行改进,用以解决该算法易陷入局部最优的问题,并将其应用于再入弹道的优化。Wu等<sup>[9]</sup>将混沌序列和自适应惯性权重因子引入标准CSO算法,提高了算法的寻优能力并将其应用于光伏系统的最大功率点跟踪控制问题。Han等<sup>[10]</sup>结合贪婪策略和变异策略提出了一种改进的离散CSO算法,用以求解0-1背包问题。

本文提出了一种改进的鸡群优化(Improved chicken swarm optimization, ICSO)算法,该算法结合萤火虫算法(Firefly algorithm, FA)、灰狼优化(Grey wolf optimizer, GWO)算法和粒子群优化(Particle swarm optimization, PSO)算法的优点,分别对原始CSO算法中每一种解的更新方式进行改进,提高算法的求解精度和收敛速度。另外,通过引入去重算子使得ICSO算法在每次迭代后可以将种群中的重复解进行替换,进而提高种群的多样性。为验证本文算法的有效性,将其

与其他算法在CEC 2014测试函数集上进行对比实验,结果表明,本文算法具有较明显优势。

## 1 鸡群优化算法

CSO算法是基于鸡群内部的等级秩序和鸡群中个体的觅食行为而提出的一种群智能优化算法<sup>[1]</sup>。鸡群内部有着特殊的等级秩序,主要为:公鸡的地位最高,而母鸡和小鸡属于弱势群体。因此,公鸡通常为决策者,母鸡们会跟着从属的公鸡进行觅食,而小鸡们则会跟着它们的母亲进行觅食。CSO算法将鸡群中的这种等级秩序和觅食行为抽象化,用群体中每个个体代表优化问题的解,并针对处于不同等级的解采用不同的迭代更新方法。在算法中,具有最优目标函数值的 $N_R$ 个解被定义为公鸡,若用 $i$ 表示第 $i$ 只鸡, $d$ 表示解的维度, $j \in [1, d]$ 表示解中的第 $j$ 维, $t$ 表示当前迭代轮数,则 $x_{i,j}^t$ 表示在第 $t$ 轮迭代时,第 $i$ 只鸡在第 $j$ 维上的位置。公鸡的位置更新方法为:

$$x_{i,j}^{t+1} = x_{i,j}^t \times [1 + N(0, \sigma^2)] \quad (1)$$

$$\sigma^2 = \begin{cases} 1, & \text{if } f_i \leq f_k \\ \exp\left(\frac{f_k - f_i}{|f_i| + \epsilon}\right), & \text{otherwise} \end{cases} \quad (2)$$

$$k \in [1, N_R], k \neq i$$

式中: $N(0, \sigma^2)$ 表示均值为0、方差为 $\sigma^2$ 的正态分布; $k$ 代表公鸡群体中不同于公鸡 $i$ 的另一只公鸡, $k \in [1, N_R]$ 且 $k \neq i$ ;  $f$ 为目标函数值; $\epsilon$ 为计算机中最小的常数,其作用是避免公式中的分母为0。式(1)和式(2)分别模拟了公鸡的随机移动觅食行为和不同组公鸡之间的竞争行为。

CSO算法中目标函数值比公鸡较差的 $N_H$ 个解被定义为母鸡,母鸡的觅食通常跟随组内的公鸡进行移动,并受到其约束,但同时还与其他鸡存在竞争关系。因此,母鸡的位置更新方法为:

$$x_{i,j}^{t+1} = x_{i,j}^t + S_1 R(x_{r_1,j}^t - x_{i,j}^t) + S_2 R(x_{r_2,j}^t - x_{i,j}^t) \quad (3)$$

$$S_1 = \exp((f_i - f_{r_1}) / [abs(f_i) + \epsilon]) \quad (4)$$

$$S_2 = \exp(f_{r_2} - f_i) \quad (5)$$

式中: $R$ 表示区间 $[0, 1]$ 上的随机数; $abs(\bullet)$ 表示取绝对值运算; $r_1$ 为母鸡 $i$ 所在组内的公鸡; $r_2$ 为鸡群中除了母鸡 $i$ 和与其同组的公鸡 $r_1$ 之外的另一只公鸡或母鸡,且 $r_2$ 的觅食能力要强于母鸡 $i$ 。由式(3)可知,母鸡的位置更新分为两部分,一部分

模拟了母鸡跟随公鸡的觅食行为;另一部分模拟了母鸡与其他鸡之间的信息交互。

小鸡是种群中目标函数值最差的  $N_c$  个解,其只能跟随它们的母亲进行觅食,因此小鸡的位置更新方法为:

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

式中:  $m$  表示小鸡  $i$  的母亲;  $F$  表示小鸡跟随母鸡进行觅食的调节参数,由于小鸡群体中的每个个体也存在差异性,因此  $F$  为区间  $[0, 2]$  上的随机数。

CSO算法的基本步骤如下<sup>[1]</sup>:

Step1 初始化参数,主要包括:种群规模  $pop$ 、解的维度  $d$ 、算法的最大迭代轮数  $G_{Max}$  和当前迭代轮数  $t$  等。

Step2 计算初始种群目标函数值。

Step3 种群内等级秩序的定义,当  $t \bmod G = 1$ ,即每隔  $G$  轮迭代,需重新定义种群内部的等级秩序。这一步主要包括两部分,分别为种群内3种角色的划分和代表不同角色的候选解之间关系的定义。

Step4 根据候选解的角色分别选择式(1)~(6)中的位置更新方法对其进行更新。

Step5 判断算法终止条件。若算法的迭代轮数  $t$  已经达到最大迭代轮数,或者已经得到一个目标函数值满足条件的候选解,则算法终止;否则,  $t = t + 1$ ,且返回 Step3 继续执行算法循环。

## 2 本文算法

标准CSO算法虽然可以通过等级机制提高种群利用率,但其位置更新方法的有效性较低,导致算法整体的搜索能力下降。因此,本文提出了一种改进的ICSO算法,该算法基于标准CSO算法的分组思想,分别对公鸡、母鸡和小鸡的位置更新方法进行改进,进而平衡算法的搜索深度和广度。

### 2.1 改进的母鸡和小鸡位置更新方法

母鸡和小鸡在种群中处于弱势地位,因此其对应解的随机性较弱。本文将FA算法<sup>[11,12]</sup>的吸引机制引入到母鸡和小鸡的位置更新方法中,用以提升标准CSO算法的搜索深度。改进后的母鸡位置更新方法为:

$$x_{i,j}^{t+1} = x_{i,j}^t + \beta(x_{r1,j}^t - x_{i,j}^t) + \alpha\xi + S_2 R(x_{r2,j}^t - x_{i,j}^t) \quad (7)$$

$$\beta = e^{-\gamma r^2} \quad (8)$$

$$r = \|x_{r1} - x_i\| = \sqrt{\sum_{j=1}^d (x_{r1,j} - x_{i,j})^2} \quad (9)$$

式中:  $\beta$  表示吸引力;  $\gamma$  是区间  $[0.1, 10]$  上的随机数,代表光吸收系数;  $r$  是  $r_1$  与  $i$  的笛卡尔距离;  $\alpha$  是随机步长;  $\xi$  是符合正态分布的随机数。

改进后的小鸡位置更新方法如下:

$$x_{i,j}^{t+1} = x_{i,j}^t + \beta(x_{m,j}^t - x_{i,j}^t) + \alpha\xi \quad (10)$$

### 2.2 改进的公鸡位置更新方法

公鸡是种群内部的优势群体,它们更加接近最优解所在的位置。然而,在标准CSO算法中,公鸡采用基于正态分布的位置更新方法,具有一定的盲目性,导致算法的收敛速度降低。为解决该问题,本文引入GWO算法<sup>[13,14]</sup>和PSO算法的位置更新方法,用以提高CSO算法中公鸡的搜索广度。改进后的公鸡位置更新方法为:

$$x_{i,j}^{t+1} = g_{best} - SD + X_v \quad (11)$$

$$D = |Cg_{best} - x_{i,j}^t|, C = 2r_3 \quad (12)$$

$$S = 2Ar_4 - A \quad (13)$$

$$X_v = \omega v_{i,j}^t + r_5 R(p_{best} - x_{i,j}^t) + r_6 R(g_{best} - x_{i,j}^t) \quad (14)$$

式中:  $S$  为收敛因子;  $D$  为当前解和全局最优解之间的距离;  $r_3, r_4$  均为区间  $[0, 1]$  上的随机数;  $C$  为摆动因子;  $A$  将随着迭代轮数的增加,线性递减到0;  $v_{i,j}^t$  为第  $i$  只公鸡在第  $t$  代的速度;  $p_{best}, g_{best}$  分别代表算法迭代过程中的局部最优解和全局最优解;  $r_5, r_6$  为学习因子;  $\omega = (0.5 + R)/2$  为惯性权重。

### 2.3 去重因子

在算法的迭代过程中可能在种群中出现两个完全相同解的情况,进而导致种群多样性降低。在ICSO算法中引入了去重操作因子,具体操作过程如下所示:

```
for i = 1 to pop do
  将  $x_i$  各维度上的值按降序排序,得到一个新解  $x_i'$ ;
  for j = i + 1 to pop do
    将  $x_j$  各维度上的值按降序排序,得到一个新解  $x_j'$ ;
    if  $x_i' == x_j'$  then
      随机选择  $x_j'$  的一个维度并改变其值;
    end
  end
end
```

### 2.3 ICSO算法流程

引入改进解更新方法及去重操作因子的IC-

SO算法的主要步骤如图1所示。

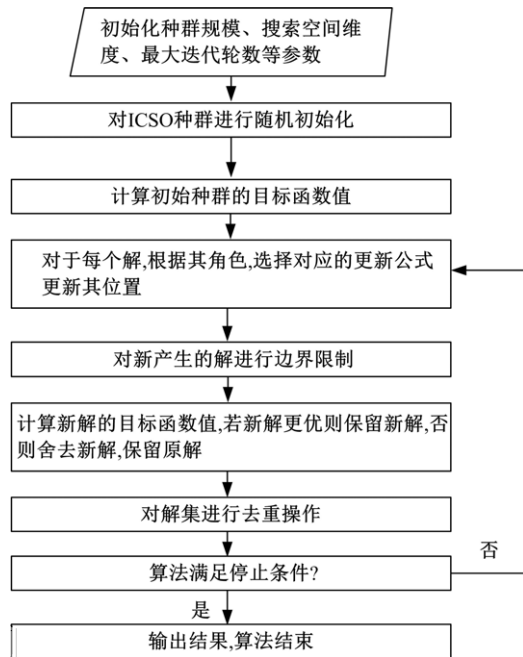


图1 ICSO算法流程图

Fig. 1 Flow chart of proposed ICSO algorithm

### 3 仿真及分析

#### 3.1 仿真设置

在CEC 2014测试函数集<sup>[15]</sup>上对ICSO算法的性能进行测试,并将其与FA算法、CSO算法、PSO算法和生物地理学优化算法(Biogeography-based optimization, BBO)的测试结果进行对比。CEC 2014测试集提供了30个最小化问题的测试函数,其中 $f_1 \sim f_3$ 为单峰函数, $f_4 \sim f_{16}$ 为简单多峰函数, $f_{17} \sim f_{22}$ 为混合函数, $f_{23} \sim f_{30}$ 为复合函数。在实验中,所有测试函数的解维度为10,解的搜索区间为 $[-100, 100]$ ,每种算法的种群规模为100。所有算法分别在每个测试函数上独立运行51次,每次运行的最大迭代轮数为10 000。

#### 3.2 求解精度分析

表1给出了各种算法在51次独立重复测试中的数值统计结果,可以直观地看出,ICSO算法在21个测试函数上都取得了最优的结果,且该算法在10个测试函数上都找到了对应函数的理论最优解。另外,通过对实验结果的进一步分析可知,ICSO算法在对简单多峰函数和复合函数的求解上拥有更高的寻优能力。

#### 3.3 收敛性能分析

鉴于篇幅原因,本文只给出每个算法在 $f_5$ 、 $f_{16}$ 和 $f_{27}$ 三个函数上的收敛趋势图,如图2~图4

所示。

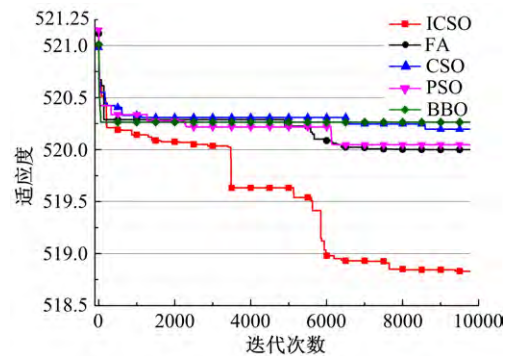


图2 各种算法在函数 $f_5$ 上的收敛曲线

Fig. 2 Convergence rate of each algorithm in function  $f_5$

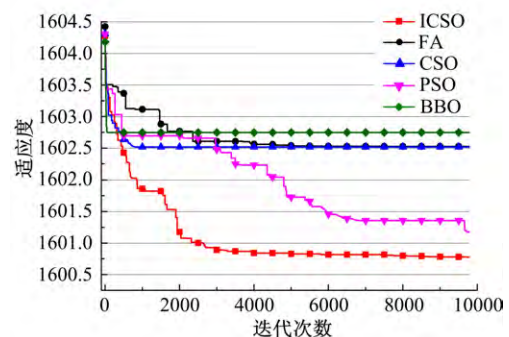


图3 各种算法在函数 $f_{16}$ 上的收敛曲线

Fig. 3 Convergence rate of each algorithm in function  $f_{16}$

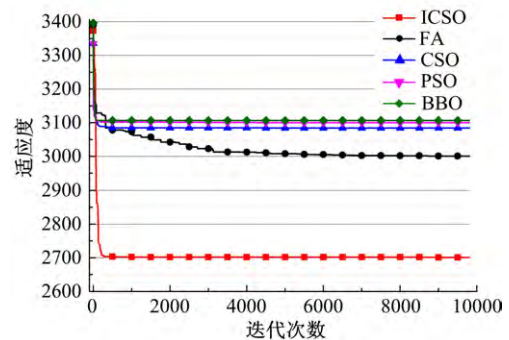


图4 各种算法在函数 $f_{27}$ 上的收敛曲线

Fig. 4 Convergence rate of each algorithm in function  $f_{27}$

从图2中可以看出,在求解函数 $f_5$ 的过程中,BBO算法和CSO算法过早地陷入了局部最优,而FA算法和PSO算法虽然在后期跳出了局部最优,但却陷入了另一个局部最优中。ICSO算法在前期收敛速度不明显,仅找到了与BBO算法和PSO算法相似的局部最优值,但在第3500轮和6000轮时,该算法又分别跳出了局部最优,得到了较好的结果。

从图3中可以看出,ICSO算法在求解函数 $f_{16}$ 的初期即表现出较快的收敛趋势,并在2000轮左



右时跳出局部最优,最终趋近理论最优值。BBO算法在前期收敛速度较高,但后期陷入了局部最优。FA算法在求解过程中收敛较为平稳,虽然结果优于BBO算法,但在运行至3000轮左右时依然陷入了局部最优。CSO算法在前期同样具有较高的收敛速度,但其与FA算法类似,在算法运行至1000轮左右时,陷入了局部最优。PSO算法虽然优于其他对比算法,但其在求解精度和收

敛速度上均不及本文ICSO算法。

从图4中可以看出,在求解函数 $f_{27}$ 的过程中,ICSO算法仅通过300轮左右的迭代即找到了该函数的理论最优值,表现出良好的性能。反观其他算法,CSO算法、PSO算法和BBO算法虽收敛速度较快,但随后均陷入了局部最优。FA算法虽然跳出了前期的局部最优,但后续收敛缓慢,并最终陷入另一个局部最优值之中。

表1 基准函数测试结果

Table 1 Results of testing benchmark functions

函数	ICSO算法	FA算法	CSO算法	PSO算法	BBO算法
$f_1$	5.55e+04 ± 4.75e+04	1.78e+03 ± 1.12e+04	2.91e+06 ± 2.44e+06	2.18e+03 ± 1.93e+04	2.60e+05 ± 2.26e+06
$f_2$	1.08e+03 ± 2.71e+02	2.96e+02 ± 2.23e+03	1.25e+07 ± 7.15e+07	1.26e+03 ± 4.20e+02	5.02E+03 ± 1.09e+05
$f_3$	7.71e+02 ± 2.86e+02	3.00e+02 ± 3.21e-04	1.45e+03 ± 6.71e+02	3.00e+02 ± 5.75e-03	1.01e+04 ± 3.39e+03
$f_4$	4.00e+02 ± 1.88e-01	4.35e+02 ± 8.24e+00	4.41e+02 ± 8.99e+00	4.00e+02 ± 1.70e+01	4.35e+02 ± 1.16e+01
$f_5$	5.19e+02 ± 5.97e+00	5.20e+02 ± 2.80e+00	5.20e+02 ± 3.47e+00	5.20e+02 ± 5.48e+00	5.20e+02 ± 7.66e-02
$f_6$	6.00e+02 ± 1.24e+00	6.00e+02 ± 2.23e-02	6.05e+02 ± 8.97e-01	6.00e+02 ± 9.31e-01	6.02e+02 ± 1.19e+00
$f_7$	7.00e+02 ± 1.00e-02	7.00e+02 ± 4.89e-02	7.07e+02 ± 2.21e+00	7.00e+02 ± 3.53e-02	7.00e+02 ± 1.14e-01
$f_8$	8.03e+02 ± 3.77e+00	8.11e+02 ± 3.07e+00	8.12e+02 ± 3.27e+00	8.00e+02 ± 1.61e-14	8.01e+02 ± 1.20e+00
$f_9$	9.06e+02 ± 4.06e+00	9.07e+02 ± 3.23e+00	9.12e+02 ± 3.43e+00	9.07e+02 ± 1.71e+00	9.14e+02 ± 4.69e+00
$f_{10}$	1.06e+03 ± 2.36e+02	1.24e+03 ± 1.68e+02	1.26e+03 ± 8.50e+01	1.13e+03 ± 6.20e+01	1.03e+03 ± 1.18e+01
$f_{11}$	1.46e+03 ± 2.01e+02	1.69e+03 ± 1.68e+02	1.66e+03 ± 1.72e+02	1.48e+03 ± 1.29e+02	2.02e+03 ± 2.10e+02
$f_{12}$	1.20e+03 ± 3.14e-02	1.20e+03 ± 6.00e-03	1.20e+03 ± 1.17e-01	1.20e+03 ± 5.62e-02	1.20e+03 ± 2.40e-01
$f_{13}$	1.30e+03 ± 6.97e-02	1.30e+03 ± 1.91e-02	1.30e+03 ± 8.45e-02	1.30e+03 ± 4.53e-02	1.30e+03 ± 9.47e-02
$f_{14}$	1.40e+03 ± 7.79e-02	1.40e+03 ± 3.86e-02	1.40e+03 ± 1.63e-01	1.40e+03 ± 2.26e-02	1.40e+03 ± 7.76e-02
$f_{15}$	1.50e+03 ± 1.81e-01	1.50e+03 ± 1.48e-01	1.50e+03 ± 4.48e-01	1.50e+03 ± 2.38e-01	1.50e+03 ± 5.68e-01
$f_{16}$	1.60e+03 ± 3.18e-01	1.60e+03 ± 4.33e-01	1.60e+03 ± 3.30e-01	1.60e+03 ± 5.77e-01	1.60e+03 ± 4.10e-01
$f_{17}$	2.16e+03 ± 2.47e+02	2.45e+03 ± 3.62e+02	3.87e+03 ± 2.80e+03	3.89e+03 ± 1.23e+03	1.21e+04 ± 3.07e+05
$f_{18}$	1.91e+03 ± 6.00e+01	6.54e+03 ± 3.13e+00	8.68e+03 ± 2.71e+03	2.43e+03 ± 3.90e+03	4.02e+03 ± 5.01e+04
$f_{19}$	1.90e+03 ± 3.48e-01	1.90e+03 ± 6.41e-01	1.90e+03 ± 8.40e-01	1.90e+03 ± 6.95e-01	1.90e+03 ± 4.52e-01
$f_{20}$	2.10e+03 ± 6.34e+01	2.02e+03 ± 2.25e+01	2.07e+03 ± 1.42e+03	2.01e+03 ± 2.09e+00	2.56e+04 ± 2.78e+04
$f_{21}$	2.36e+03 ± 6.19e+00	2.55e+03 ± 2.10e+02	2.37e+03 ± 7.96e+02	2.22e+03 ± 5.13e+01	1.20e+04 ± 5.65e+04
$f_{22}$	2.22e+03 ± 7.12e+01	2.22e+03 ± 5.66e+01	2.23e+03 ± 3.30e+01	2.20e+03 ± 1.00e+01	2.22e+03 ± 4.51e+01
$f_{23}$	2.40e+03 ± 7.12e+01	2.63e+03 ± 1.26e-07	2.63e+03 ± 1.91e+01	2.63e+03 ± 1.02e-12	2.63e+03 ± 1.28e-01
$f_{24}$	2.52e+03 ± 4.68e+00	2.52e+03 ± 4.87e+00	2.54e+03 ± 1.24e+01	2.51e+03 ± 4.04e+00	2.53e+03 ± 7.41e+00
$f_{25}$	2.61e+03 ± 9.46e+00	2.62e+03 ± 1.79e+01	2.68e+03 ± 9.89e+00	2.70e+03 ± 3.72e+01	2.70e+03 ± 9.47e-02
$f_{26}$	2.70e+03 ± 4.46e-02	2.70e+03 ± 1.62e-02	2.70e+03 ± 4.45e-02	2.70e+03 ± 4.31e-02	2.70e+03 ± 2.60e+01
$f_{27}$	2.70e+03 ± 6.15e+01	3.00e+03 ± 1.37e+02	3.08e+03 ± 1.86e+02	3.10e+03 ± 1.67e+02	3.11e+03 ± 1.84e+02
$f_{28}$	3.00e+03 ± 6.35e+01	3.16e+03 ± 1.43e+02	3.19e+03 ± 1.20e+01	3.19e+03 ± 7.26e+01	3.17e+03 ± 2.69e+01
$f_{29}$	3.14e+03 ± 9.52e+01	3.33e+03 ± 2.41e+05	3.67e+03 ± 2.30e+02	3.23e+03 ± 8.25e+05	3.57e+03 ± 5.15e+02
$f_{30}$	3.47e+03 ± 6.87e+01	3.75e+03 ± 1.59e+02	3.92e+03 ± 3.03e+02	3.63e+03 ± 2.46e+02	3.65e+03 ± 2.59e+02
最优总数	21	11	7	17	9

## 4 结束语

为克服原始鸡群优化算法求解效率较低且收敛性较弱的问题,提出了一种改进的鸡群优化算

法。该算法在原始鸡群优化算法的种群分组更新机制的基础上,首先引入了狼群优化算法和粒子群优化算法的解更新方法,提高了自身的搜索深

度和广度。另外,引入了去重操作算子,提高了算法的种群多样性。由实验结果可知,与其他几种群智能优化算法相比,本文ICSO算法具有更高的求解精度和更快的收敛速度,可作为求解优化问题的有效工具。

#### 参考文献:

- [1] Meng X, Liu Y, Gao X, et al. A New Bio-inspired Algorithm: Chicken Swarm Optimization[M]. Berlin: Springer International Publishing, 2014.
- [2] Banerjee S, Chattopadhyay S. Improved serially concatenated convolution turbo code (SCCTC) using chicken swarm optimization[C]//IEEE Power, Communication and Information Technology Conference (PCITC), Bhubaneswar, India, 2015.
- [3] Hafez A I, Zawbaa H M, Emary E, et al. An innovative approach for feature selection based on chicken swarm optimization[C]//IEEE International Conference on Soft Computing & Pattern Recognition, Tokyo, Japan, 2016.
- [4] Chen P, Mao Y. Wireless sensor network node localization algorithm based on chicken swarm optimization and multi-power mobile anchor[C]//The 3rd International Conference on Materials Engineering, Manufacturing Technology and Control, Taiyuan, China, 2016.
- [5] Mu Y, Zhang L, Chen X, et al. Optimal trajectory planning for robotic manipulators using chicken swarm optimization[C]//IEEE International Conference on Intelligent Human-machine Systems & Cybernetics, Chengdu, China, 2016.
- [6] Roslina, Zarlis M, Yanto I T R, et al. A framework of training ANFIS using chicken swarm optimization for solving classification problems[C]//IEEE International Conference on Informatics & Computing, Mataram, Indonesia, 2017.
- [7] Yi Z, Liu J, Wang S, et al. PAPR reduction technology based on CSO algorithm in CO-OFDM system [C]//The International Conference on Optical Communications & Networks, Wuzhen, China, 2017.
- [8] Wu Yu, Yan Bo, Qu Xiang-ju. Improved chicken swarm optimization method for reentry trajectory optimization[J]. Mathematical Problems in Engineering, 2018(2): 1-13.
- [9] Wu Z, Yu D, Kang X. Application of improved chicken swarm optimization for MPPT in photovoltaic system[J]. Optimal Control Applications and Methods, 2018, 39(2):1029-1042.
- [10] Han M, Liu S. An improved binary chicken swarm optimization algorithm for solving 0-1 Knapsack problem[C]//The 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, 2017.
- [11] Yang, S X. Firefly algorithm, stochastic test functions and design optimisation[J]. International Journal of Bio-inspired Computation, 2010, 2(2):78-84.
- [12] 王沈娟, 高晓智. 萤火虫算法研究综述[J]. 微型机与应用, 2015, 34(8):8-11.  
Wang Shen-juan, Gao Xiao-zhi. A survey of firefly algorithm[J]. Microcomputer and Its Applications, 2015, 34(8):8-11.
- [13] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(3):46-61.
- [14] 龙文, 赵东泉, 徐松金, 等. 求解约束优化问题的改进灰狼优化算法[J]. 计算机应用, 2015, 35(9):2590-2595.  
Long Wen, Zhao Dong-quan, Xu Song-jin, et al. Improved grey wolf optimization algorithm for constrained optimization problem[J]. Journal of Computer Applications, 2015, 35(9):2590-2595.
- [15] Liang Jing, Qu B Y, Suganthan Ponnuthurai N, et al. Problem definitions and evaluation criteria for the CEC 2013 special session and competition on single objective real-parameter numerical optimization[C]//IEEE Congress on Evolutionary Computation, Beijing, China, 2014.