

文章编号: 1000-5862(2018)03-0323-08

基于模糊推理的鸡群优化算法

史旭栋¹, 高岳林^{2*}

(1. 宁夏大学数学统计学院, 宁夏 银川 750021; 2. 北方民族大学信息与系统科学研究所, 宁夏 银川 750021)

摘要: 针对鸡群算法(CSO)在求解高维复杂优化问题时往往会陷入局部解的问题, 提出了基于模糊推理的鸡群优化算法. 该方法利用模糊推理改进了母鸡和小鸡的位置更新公式, 增强了鸡群全局搜索能力; 利用惯性粒子, 增强了鸡群的信息共享, 从而增加了局部搜索能力, 并用 Tent 映射对粒子进行扰动. 数值试验结果表明: 该算法能快速收敛到全局最优解, 而且具有较高的全局寻优能力和计算精度.

关键词: 鸡群优化; 模糊推理; 惯性粒子

中图分类号: TP 18 **文献标志码:** A **DOI:** 10.16357/j.cnki.issn1000-5862.2018.03.17

0 引言

群智能算法(Swarm Intelligence, SI)是分散化、自组织化的集体行为, 它是模拟鱼群、鸟群^[1]等动物的社会行为简化而成的全局优化方法. 受生物系统的灵感, 各种进化算法在过去几十年中得到了较大的发展. 群智能算法数不胜数, 如粒子群优化算法(Particle Swarm Optimization, PSO)^[2]、蚁群算法(Ant Colony Optimization, ACO)^[3]、蝙蝠群算法(Bat Algorithm, BA)^[4]等. 群智能算法也广泛地应用于工程中^[5-8]. 鸡群优化算法(Chicken Swarm Optimization, CSO)是在2014年由Meng Xiangbing等^[9]提出的一种模拟鸡群社会行为的全局优化算法. 孔飞等^[10]将公鸡学习部分加入到小鸡的更新公式中, 崔东文^[11]将鸡群算法应用于洪旱灾评估模型, 李振壁等^[12]将模拟退火算法与鸡群算法融合. CSO在解决低维简单问题时具有收敛速度快、计算精度高的特点, 但在求解高维复杂问题时, 鸡群算法易于陷入局部最优解. 为了提高群智能算法的搜索能力, Wei-der Chang等^[13]从亚群体选择一个领域粒子为其他粒子提供搜索方向, 从而使粒子的搜索范围扩大. 此外, Liu Yu等^[14]将中心的额外粒子加入算法, 使整个算法更接近全局最优解. Li Naijen等^[15]利用惯性粒子去提供搜索方向. 在过去几年中, 由于模糊理论广泛且有效地在工程中应用, 结合模糊逻辑控制

(fuzzy logic control, FLC)的进化算法^[16-20]层出不穷.

近年来, 鸡群优化算法已成为进化计算和群体智能等领域研究的热点之一. 与其他智能算法类似, 鸡群优化算法也存在早熟收敛和局部寻优能力差等缺点. 目前解决这些问题的主要方法是增加种群局部搜索能力. 因此, 解决这个问题的主要方法引入模糊推理.

1 鸡群优化算法的基本概念以及模糊推理的简述

1.1 鸡群优化算法

鸡群优化算法是一个新的智能优化算法, 它模拟鸡群的行为和等级制度. 在该算法中, 鸡群被分为若干组, 每个组是由一只公鸡、若干只母鸡和小鸡组成. 假设 N_g 、 N_H 、 N_C 、 N_M 分别表示公鸡、母鸡、小鸡和母鸡妈妈的数量. 最好的 N_R 假设为公鸡, 最差的 N_C 为小鸡, 其余的均为母鸡. 在 N 只鸡的鸡群中, 用 $x_{ij}(t)$ 表示第 t 代中第 i 只鸡的第 j 维分量, px_{ij} 表示第 i 只鸡的最优位置^[21].

不同的鸡遵循不同的运动规律, 适应度较好的公鸡比较差的公鸡更接近食物, 其位置更新公式为

$$x_{ij}(t+1) = px_{ij}(t) \times (1 + r_{an}(0, \sigma^2)), \quad (1)$$
$$\sigma^2 = \begin{cases} 1, & f_1 \leq f_k, \\ \exp\left(\frac{f_k - f_i}{|f_i| + \varepsilon}\right), & \text{其他}, \end{cases}$$

收稿日期: 2017-08-06

基金项目: 国家自然科学基金(61561001)和北方民族大学重点科研基金(2015KJ10)资助项目.

通信作者: 高岳林(1963-), 陕西榆林人, 教授, 博士, 主要从事最优化理论方法及应用, 智能计算机与智能信息处理研究.

E-mail: 1787958385@qq.com

其中 $r_{an}(0, \sigma^2)$ 是均值为 0、方差为 σ^2 的高斯分布。 ε 为一个极小量 k 为公鸡群中不为 i 的随机数 f 为 x 所对应的适应度值。

母鸡的更新公式为 $x_{ij}(t+1) = px_{ij}(t) + c_1 \cdot r_a(x_{r_1j}(t) - x_{ij}(t)) + c_2 r_a(x_{r_2j}(t) - x_{ij}(t))$ $\rho_1 = e^{(f_i - f_{r_1}) / (1/f_i + \varepsilon)}$ $\rho_2 = e^{f_{r_2} - f_i}$ 其中 r_a 为 $[0, 1]$ 之间的随机数 r_1 表示第 i 只母鸡所在子鸡群的公鸡 r_2 表示随机从鸡群中挑选一只, 并且 $r_1 \neq r_2$ 。

小鸡的更新公式为 $x_{ij}(t+1) = px_{ij}(t) + F(px_{mj} - x_{ij})$ 其中 m 为第 i 只小鸡的母鸡 F 为 $[0, 2]$ 之间的随机数。

1.2 模糊推理

本文利用文献 [22-23] 中的经典模糊推理, 其形式为: 若 θ_1 是 A_1^l θ_2 是 A_2^l, \dots, θ_k 是 A_k^l 则 y 是 B^l , 其中 θ_k 是模糊输入变量 A_1^l 和 B^l 分别前隶属度函数和后隶属度函数 l 为规则序号, 输出 y 为

$$y = \sum_{i=1}^l (B^i \prod_{j=1}^k A_j^i(\theta_j)) / \sum_{i=1}^l (\prod_{j=1}^k A_j^i(\theta_j)).$$

2 基于模糊推理的鸡群优化算法

2.1 惯性粒子

首先可以求出第 t 代整个鸡群中最好的那只鸡 $x^G(t)$ 与全局最差的那只鸡 $x^B(t)$, 然后每个子群中的公鸡均为该种群中适应度最好的, 故将公鸡作为该子群中每只鸡的当前最好个体, 记为 $x_i^P(t)$ 。将惯性权重作为一个较好的搜索方向 $x^W(t) = [x_1^W(t), x_2^W(t), \dots, x_j^W(t), \dots, x_D^W(t)]$,

$$x_j^W(t) = \sum_{i=1}^P c_i^W x_{ij}^P(t) / \sum_{i=1}^P c_i^W, j = 1, 2, \dots, D, \quad (2)$$

其中

$$c_i^W = \frac{f(x^G(t)) - f(x_i^P(t)) + \varepsilon}{f(x^B(t)) - f(x^G(t)) + \varepsilon}, \quad (3)$$

$i = 1, 2, \dots, M$ ε 为一个极小的量, 防止 (3) 式中分母为 0, 而导致的 c_i^W 不可计算, 对于第 t 代粒子来说, $x^W(t)$ 就好像一个重心。

2.2 速度和位置更新公式的改进

在文献 [24] 中, 将模糊推理引入粒子群优化算法, 使得粒子群优化算法的全局搜索能力和局部搜索能力得到较好的提高。而本文将模糊推理引入鸡群算法, 主要对母鸡和小鸡的更新公式进行改进, 由于母鸡的全局收敛性相对其他算法比较好, 但是局部搜索能力相对较差, 所以对母鸡的更新公式进行

改进; 而小鸡则是“追随”母鸡, 故将母鸡更新公式改进的同时, 也将小鸡的局部搜索能力提高, 同时改进小鸡的更新公式, 使得小鸡在寻优过程中的全局搜索能力和局部搜索能力增强。

引入惯性粒子将母鸡和小鸡的更新公式改为

$$x_{ij}(t+1) = x_{ij}(t) + u_{ij} c_3 r_a(x_j^W(t) - x_{ij}(t)) + \bar{u}_{ij} c_1 r_a(x_{r_1j}(t) - x_{ij}(t)) + \bar{u}_{ij} c_2 r_a(x_{r_2j}(t) - x_{ij}(t)), \quad (4)$$

$$x_{ij}(t+1) = x_{ij}(t) + u_{ij} c_4 r_a(x_j^W(t) - x_{ij}(t)) + \bar{u}_{ij} F(x_{mj}(t) - x_{ij}(t)), \quad (5)$$

其中 r_a 为 $[0, 1]$ 之间均匀分布的随机数; r_1 为第 i 只母鸡自身所在群中的公鸡; r_2 为整个鸡群中公鸡和母鸡中随机选取的任意个体, 且 $r_1 \neq r_2$; $x^W(t)$ 为惯性权重; c_1, c_2, c_3, c_4 为学习因子 $\mu = r_a \alpha_1$, $u + \bar{u} = 1$, m 为第 i 只公鸡对应的母鸡; F 为 0 与 2 之间跟随系数。

2.3 模糊推理

在文献 [24] 中, 利用 3 个参数 nf_i, α_i 和 w_i 对算法进行自适应改变, 其中 $nf_i(f(x_i(t)) - F_{\min} + \varepsilon) / (F_{\max} - F_{\min} + \varepsilon)$ 。

利用表 1 和表 2 对参数进行调整, 可得出 $\Delta\alpha_i$ 和 Δw_i 的类型, 再通过隶属度函数, 如图 1 和图 2 所示。

表 1 参数 α_i 调整的模糊规则

| 规则号 | 输入 | | 结果 |
|-----|--------|------------|------------------|
| | nf_i | α_i | $\Delta\alpha_i$ |
| 1 | S | S | PL |
| 2 | S | M | PS |
| 3 | S | L | Z |
| 4 | M | S | PS |
| 5 | M | M | Z |
| 6 | M | L | NS |
| 7 | L | S | Z |
| 8 | L | M | NS |
| 9 | L | L | NL |

表 2 参数 w_i 调整的模糊规则

| 规则号 | 输入 | | 结果 |
|-----|--------|-------|--------------|
| | nf_i | w_i | Δw_i |
| 1 | S | S | PL |
| 2 | S | M | PL |
| 3 | S | L | Z |
| 4 | M | S | NL |
| 5 | M | M | Z |
| 6 | M | L | Z |
| 7 | L | S | Z |
| 8 | L | M | NL |
| 9 | L | L | NL |

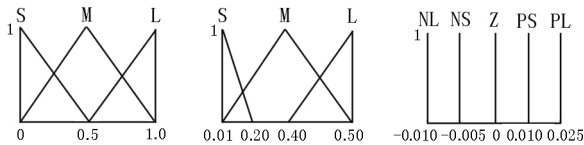
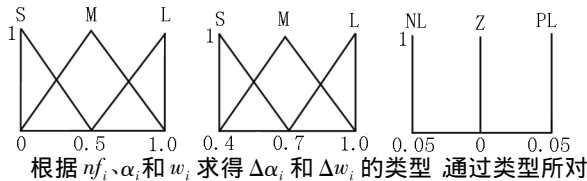


图 1 $\Delta\alpha_i$ 的模糊推理规则



根据 nf_i 、 α_i 和 w_i 求得 $\Delta\alpha_i$ 和 Δw_i 的类型 通过类型所对应的值 根据隶属度函数求出具体的增量。

图 2 Δw_i 的模糊推理规则

2.4 混沌 Tent 映射扰动

为了使算法在迭代后期仍然具有较好的多样性 本文采用混沌扰动 算法 1 如下:

- (i) 随机生成一个初始点 x_0 ,设最大混沌迭代次数为 M ,令 $m = 1$;
- (ii) 令粒子数 $n = 1$;
- (iii) 令维数 $d = 1$;
- (iv) 根据 Tent 映射^[25] 混沌方程计算混沌序列 x_d ,即

$$x_d = \begin{cases} x_{d-1}/0.7, & x_{d-1} < 0.7, \\ 10/3x_{d-1}(1 - x_{d-1}) & \text{其他;} \end{cases}$$

- (v) 对第 n 个支配解的第 d 维进行混沌局部搜索 $x_{id} = x_{id} + \alpha \cdot \beta \cdot x_d$,其中 $\alpha = r_a$ $\beta = (-1)^d$;
- (vi) 对超出边界的粒子进行处理;
- (vii) $d = d + 1$ 若 $d \leq D$ 则转(iv);
- (viii) $n = n + 1$ 若 $n \leq N$ 则转(iii);
- (ix) 判断新生成的解是否为最优解;

- (x) $m = m + 1$ 若 $m \leq M$ 则转(ii) ,否则结束搜索.

2.5 算法总流程

算法的总流程为:(i) 随机生成鸡群;(ii) 求得每只鸡的适应度 ,并求出最好的鸡 $x^G(t)$ 和最差的鸡 $x^B(t)$;(iii) 根据(2) 式和(3) 式求得惯性粒子 x^w ;(iv) 将鸡群分成若干子群 ,确定公鸡母鸡和小鸡;(v) 根据表 1 和表 2 计算 α_i 与 w_i ;(vi) 利用算法 1 对鸡群进行混沌扰动;(vii) 利用(1) 式、(4) 式和(5) 式对公鸡、母鸡、小鸡分别进行更新;(viii) 判断是否满足终止条件 ,如果满足就输出 gbest ,否则 ,转(ii) .

3 数值分析

本文的测试函数如表 3 所示 ,数值分析结果如表 4 所示. 每一个算法在不同维度上独立运行 30 次所得数据如表 5 和表 6 所示.

表 3 测试函数

| 函数名 | 函数代号 | 取值范围 | 最优值 |
|------------------|------|-----------------|-----|
| Sphere | F1 | [- 100 ,100] | 0 |
| Rosenbrock | F2 | [- 30 ,30] | 0 |
| Rastrigrin | F3 | [- 5.12 ,5.12] | 0 |
| Griewank | F4 | [- 600 ,600] | 0 |
| Ackley | F5 | [- 32 ,32] | 0 |
| High Conditioned | F6 | [- 100 ,100] | 0 |
| Elliptic | | | |
| Bent Cigar | F7 | [- 100 ,100] | 0 |
| Discus | F8 | [- 100 ,100] | 0 |

表 4 参数设置

| 算法 | 参数 |
|------|--|
| PSO | $c_1 = c_2$ $\mu = 0.73$ |
| BA | $\alpha = \gamma = 0.5$ $f_{\min} = 0$ $f_{\max} = 2$ $A_0 \in [0, 2]$ $r_0 \in [0, 1]$ |
| CSO | $N_R = 0.2N$ $N_H = 0.6$ $N_c = N - N_R - N_H$ $N_M = 0.1N_H$ $G = 10$ $F \in r_{an}(0.4, 1)$ |
| ICSO | $N_R = 0.2N$ $N_H = 0.6$ $N_c = N - N_R - N_H$ $N_M = 0.1N_H$ $G = 10$ $F \in r_{an}(0.4, 1)$ $C = 0.4$ |
| MCSO | $N_R = 0.2N$ $N_H = 0.6$ $N_c = N - N_R - N_H$ $N_M = 0.1N_H$ $G = 10$ $F \in r_{an}(0.4, 1)$ $c_3 = 1.0$ $c_4 = 2.0$, $\alpha = 0.01$ $\mu = 0.4$ |

表 5 不同算法对于 10 维测试函数的数据对比

| Term | 维数 | F1 | | | | |
|------|----|----------------------------|-------------|-----------------------------|-----------------------------|------|
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 10 | $4.771\ 5 \times 10^{-10}$ | 90.034 4 | $2.699\ 1 \times 10^{-129}$ | $1.553\ 7 \times 10^{-137}$ | 0 |
| 最差 | 10 | $9.914\ 1 \times 10^{-8}$ | 1 067.932 4 | $2.375\ 3 \times 10^{-123}$ | $6.929\ 1 \times 10^{-131}$ | 0 |
| 平均 | 10 | $2.772\ 7 \times 10^{-8}$ | 547.433 1 | $1.589\ 2 \times 10^{-124}$ | $6.148\ 0 \times 10^{-132}$ | 0 |
| 标准差 | 10 | $2.785\ 8 \times 10^{-8}$ | 327.348 8 | $6.131\ 5 \times 10^{-124}$ | $1.799\ 8 \times 10^{-131}$ | 0 |

表 5(续)

| Term | 维数 | F2 | | | | |
|------|----|--------------------------|-----------------------|----------------------------|----------------------------|----------------------------|
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 10 | 0.226 41 | 0.131 25 | 6.285 10 | 6.252 70 | 3.868×10^{-5} |
| 最差 | 10 | 29.688 70 | 978.681 70 | 7.204 7 | 7.241 90 | 8.491×10^{-2} |
| 平均 | 10 | 6.044 60 | 149.276 50 | 6.840 10 | 6.829 10 | 4.570×10^{-3} |
| 标准差 | 10 | 6.071 70 | 272.497 20 | 0.296 66 | 0.295 15 | 2.606×10^{-2} |
| Term | 维数 | F3 | | | | |
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 10 | $3.027 9 \times 10^{-4}$ | 8.954 7 | 0 | 0 | 0 |
| 最差 | 10 | 6.992 7 | 56.712 4 | 0 | 0 | 19.402 0 |
| 平均 | 10 | 3.112 8 | 25.371 5 | 0 | 0 | 2.371 5 |
| 标准差 | 10 | 1.955 0 | 12.918 3 | 0 | 0 | 1.837 7 |
| Term | 维数 | F4 | | | | |
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 10 | 0.039 381 | 3.829 0 | 0 | 0 | 0 |
| 最差 | 10 | 0.348 680 | 33.414 7 | 0 | 0 | $1.464 4 \times 10^{-1}$ |
| 平均 | 10 | 0.148 720 | 15.260 0 | 0 | 0 | $8.153 9 \times 10^{-2}$ |
| 标准差 | 10 | 0.091 360 | 7.092 1 | 0 | 0 | $6.821 4 \times 10^{-2}$ |
| Term | 维数 | F5 | | | | |
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 10 | $9.792 0 \times 10^{-6}$ | 7.349 7 | $8.881 8 \times 10^{-16}$ | 0 | 0 |
| 最差 | 10 | $1.783 2 \times 10^{-4}$ | 13.242 1 | $2.664 5 \times 10^{-15}$ | 0 | 0 |
| 平均 | 10 | $4.427 5 \times 10^{-5}$ | 11.285 8 | $6.513 3 \times 10^{-16}$ | 0 | 0 |
| 标准差 | 10 | $3.849 2 \times 10^{-5}$ | 1.652 6 | $9.173 1 \times 10^{-16}$ | 0 | 0 |
| Term | 维数 | F6 | | | | |
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 10 | $9.841 5 \times 10^{-7}$ | $7.724 3 \times 10^6$ | $1.108 2 \times 10^{-128}$ | $1.355 7 \times 10^{-122}$ | $8.841 4 \times 10^{-243}$ |
| 最差 | 10 | $8.929 4 \times 10^{-5}$ | $1.452 4 \times 10^8$ | $2.698 7 \times 10^{-123}$ | $4.927 7 \times 10^{-119}$ | $5.407 9 \times 10^{-240}$ |
| 平均 | 10 | $2.623 8 \times 10^{-5}$ | $4.997 6 \times 10^7$ | $2.100 9 \times 10^{-124}$ | $3.470 1 \times 10^{-120}$ | $4.662 9 \times 10^{-240}$ |
| 标准差 | 10 | $2.722 1 \times 10^{-5}$ | $3.580 2 \times 10^7$ | $6.899 5 \times 10^{-124}$ | $1.269 1 \times 10^{-120}$ | 0 |
| Term | 维数 | F7 | | | | |
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 10 | $2.203 5 \times 10^{-4}$ | $3.516 7 \times 10^6$ | $4.185 1 \times 10^{-125}$ | $2.602 6 \times 10^{-120}$ | $1.645 4 \times 10^{-102}$ |
| 最差 | 10 | $9.610 8 \times 10^{-3}$ | $6.361 5 \times 10^9$ | $6.032 9 \times 10^{-119}$ | $2.429 3 \times 10^{-113}$ | $4.304 8 \times 10^{-90}$ |
| 平均 | 10 | $2.680 9 \times 10^{-3}$ | $3.348 8 \times 10^8$ | $1.239 2 \times 10^{-119}$ | $3.076 5 \times 10^{-114}$ | $1.663 1 \times 10^{-91}$ |
| 标准差 | 10 | $2.362 0 \times 10^{-3}$ | $1.557 4 \times 10^8$ | $2.328 7 \times 10^{-119}$ | $7.569 3 \times 10^{-114}$ | $1.298 5 \times 10^{-91}$ |

表 5(续)

| Term | 维数 | F8 | | | | |
|------|----|---------------------------|------------------------|-----------------------------|-----------------------------|-----------------------------|
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 10 | $2.811\ 8 \times 10^{-9}$ | $1.568\ 0 \times 10^4$ | $2.096\ 0 \times 10^{-131}$ | $3.113\ 5 \times 10^{-135}$ | $6.724\ 3 \times 10^{-186}$ |
| 最差 | 10 | $1.284\ 9 \times 10^{-6}$ | $9.562\ 2 \times 10^4$ | $1.050\ 9 \times 10^{-122}$ | $1.358\ 4 \times 10^{-129}$ | $2.123\ 6 \times 10^{-183}$ |
| 平均 | 10 | $1.695\ 6 \times 10^{-7}$ | $3.754\ 7 \times 10^4$ | $7.375\ 2 \times 10^{-124}$ | $9.622\ 2 \times 10^{-131}$ | $1.710\ 2 \times 10^{-184}$ |
| 标准差 | 10 | $2.904\ 5 \times 10^{-7}$ | $2.263\ 5 \times 10^4$ | $2.704\ 6 \times 10^{-123}$ | $3.494\ 4 \times 10^{-130}$ | 0 |

表 6 不同算法对于 100 维测试函数的数据对比

| Term | 维数 | F1 | | | | |
|------|-----|------------------------|------------------------|---------------------------|----------------------------|----------------------------|
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 100 | 21 896.387 5 | 16 275.123 4 | $5.071\ 5 \times 10^{-7}$ | $8.739\ 3 \times 10^{-20}$ | $2.008\ 5 \times 10^{-24}$ |
| 最差 | 100 | 36 896.574 5 | 42 796.546 5 | 155.319 3 | 21.354 7 | $2.628\ 3 \times 10^{-16}$ |
| 平均 | 100 | 26 894.748 0 | 27 738.270 1 | 29.668 9 | 1.867 5 | $4.350\ 8 \times 10^{-18}$ |
| 标准差 | 100 | 3 759.994 8 | 7 626.996 9 | 50.028 5 | 5.642 4 | $1.339\ 2 \times 10^{-17}$ |
| Term | 维数 | F2 | | | | |
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 100 | $3.045\ 8 \times 10^7$ | $2.252\ 2 \times 10^5$ | $1.514\ 3 \times 10^6$ | $6.473\ 6 \times 10^6$ | $3.618\ 4 \times 10^1$ |
| 最差 | 100 | $8.675\ 4 \times 10^7$ | $2.201\ 9 \times 10^7$ | $1.429\ 2 \times 10^7$ | 104.739 | $1.161\ 3 \times 10^1$ |
| 平均 | 100 | $4.810\ 2 \times 10^7$ | $7.041\ 8 \times 10^6$ | $6.526\ 9 \times 10^6$ | 72.329 3 | $4.503\ 3 \times 10^1$ |
| 标准差 | 100 | $1.373\ 1 \times 10^7$ | $5.024\ 3 \times 10^6$ | $3.485\ 8 \times 10^6$ | 45.179 1 | $1.660\ 8 \times 10^1$ |
| Term | 维数 | F3 | | | | |
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 100 | 779.520 7 | 133.359 6 | $3.984\ 7 \times 10^{-9}$ | 0 | 0 |
| 最差 | 100 | 1 109.512 5 | 311.457 8 | $1.278\ 9 \times 10^{-5}$ | $3.908\ 0 \times 10^{-14}$ | $1.911\ 1 \times 10^2$ |
| 平均 | 100 | 958.344 1 | 197.385 2 | $3.261\ 0 \times 10^{-6}$ | $2.605\ 3 \times 10^{-15}$ | $3.211\ 8 \times 10^1$ |
| 标准差 | 100 | 78.690 1 | 50.407 5 | $3.865\ 7 \times 10^{-6}$ | $1.009\ 0 \times 10^{-14}$ | $3.221\ 0 \times 10^1$ |
| Term | 维数 | F4 | | | | |
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 100 | 169.066 3 | 201.335 1 | $1.532\ 2 \times 10^{-7}$ | 0 | 0 |
| 最差 | 100 | 339.621 1 | 466.504 0 | 9.603 6 | 0 | 0 |
| 平均 | 100 | 251.302 0 | 322.494 1 | 1.345 2 | 0 | 0 |
| 标准差 | 100 | 41.479 5 | 74.951 7 | 2.552 0 | 0 | 0 |
| Term | 维数 | F5 | | | | |
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 100 | 14.181 3 | 12.724 2 | $9.180\ 3 \times 10^{-5}$ | $3.568\ 2 \times 10^{-22}$ | $1.159\ 6 \times 10^{-16}$ |
| 最差 | 100 | 16.602 5 | 15.060 8 | 9.838 4 | $4.446\ 1 \times 10^{-19}$ | $2.311\ 7 \times 10^{-16}$ |
| 平均 | 100 | 15.409 6 | 13.824 6 | 5.290 4 | $1.636\ 6 \times 10^{-19}$ | $7.743\ 6 \times 10^{-16}$ |
| 标准差 | 100 | 0.710 16 | 0.699 25 | 3.405 2 | $1.060\ 3 \times 10^{-19}$ | $5.658\ 1 \times 10^{-16}$ |

表 6(续)

| Term | 维数 | F6 | | | | |
|------|-----|------------------------|------------------------|---------------------------|----------------------------|----------------------------|
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 100 | $1.019\ 4 \times 10^8$ | $5.296\ 2 \times 10^8$ | $9.215\ 7 \times 10^{-5}$ | $7.247\ 8 \times 10^{-48}$ | $2.257\ 4 \times 10^{-15}$ |
| 最差 | 100 | $2.208\ 9 \times 10^8$ | $2.854\ 9 \times 10^9$ | 1 911.089 8 | $3.442\ 4 \times 10^{-9}$ | $1.152\ 9 \times 10^{-6}$ |
| 平均 | 100 | 1.555×10^8 | $1.181\ 7 \times 10^9$ | 346.250 1 | $3.236\ 5 \times 10^{-10}$ | $2.368\ 4 \times 10^{-8}$ |
| 标准差 | 100 | $3.556\ 6 \times 10^7$ | $5.607\ 1 \times 10^7$ | 566.431 9 | $9.147\ 5 \times 10^{-10}$ | $6.568\ 3 \times 10^{-7}$ |

| Term | 维数 | F7 | | | | |
|------|-----|---------------------------|---------------------------|---------------------------|----------------------------|----------------------------|
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 100 | $1.877\ 5 \times 10^{10}$ | $8.525\ 3 \times 10^9$ | $9.023\ 8 \times 10^{-8}$ | $8.516\ 5 \times 10^{-47}$ | $1.778\ 8 \times 10^{-73}$ |
| 最差 | 100 | $3.493\ 7 \times 10^{10}$ | $4.294\ 8 \times 10^{10}$ | $2.038\ 1 \times 10^8$ | $1.555\ 4 \times 10^{-33}$ | $9.196\ 2 \times 10^{-71}$ |
| 平均 | 100 | $2.798\ 1 \times 10^{10}$ | $2.746\ 9 \times 10^{10}$ | $4.810\ 5 \times 10^7$ | $1.782\ 5 \times 10^{-34}$ | $5.355\ 1 \times 10^{-72}$ |
| 标准差 | 100 | $4.724\ 8 \times 10^9$ | $7.173\ 7 \times 10^9$ | $6.604\ 9 \times 10^7$ | $4.704\ 8 \times 10^{-34}$ | $6.899\ 4 \times 10^{-72}$ |

| Term | 维数 | F8 | | | | |
|------|-----|------------------------|------------------------|---------------------------|----------------------------|--------------------------|
| | | PSO | BA | CSO | ICSO | MCSO |
| 最好 | 100 | $2.211\ 5 \times 10^4$ | $1.071\ 1 \times 10^5$ | $2.696\ 0 \times 10^{-6}$ | $1.960\ 0 \times 10^{-68}$ | 1.121×10^{-163} |
| 最差 | 100 | $4.871\ 3 \times 10^4$ | $6.121\ 0 \times 10^5$ | 38.147 7 | $5.910\ 5 \times 10^{-67}$ | 8.752×10^{-155} |
| 平均 | 100 | $3.220\ 7 \times 10^4$ | $2.709\ 9 \times 10^5$ | 4.644 9 | $1.334\ 2 \times 10^{-67}$ | 6.677×10^{-157} |
| 标准差 | 100 | 6 807.950 5 | $1.207\ 0 \times 10^5$ | 11.699 4 | $1.616\ 7 \times 10^{-67}$ | 6.453×10^{-156} |

由表 5 可得 ,对于 8 个测试函数来说 ,MCSO 的求解结果除了 F3、F4、F7 以外效果均比 ICSO 要好 ,其中对 F7 而言 ,结果相差不多 .而 MCSO 与 ICSO 都比 PSO、BA 和 CSO 要好.

由表 6 可知 ,对于所有的测试函数来说 ,除了 F5 与 F6 以外 ,MCSO 的效果均比 ICSO 要好 ,而且要好得多 .明显可以看出 CSO、ICSO 和 MCSO 在处理高维复杂问题要远远好于 PSO 和 BA .下面是在 100 维的条件下 ,对 8 个测试函数所得到的收敛曲线如图 3 ~ 图 10 所示 ,由此可得 PSO 与 BA 的精度远不及 CSO、ICSO、MCSO 3 种算法 ,明显可以看出 PSO 与 BA 2 种算法都相差不多 .在收敛速度方面 ,也是 3 种算法更胜一筹 .

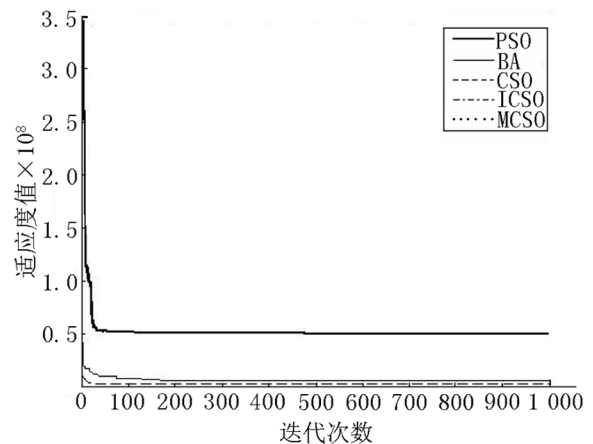


图 4 5 种算法对 F2 的收敛曲线

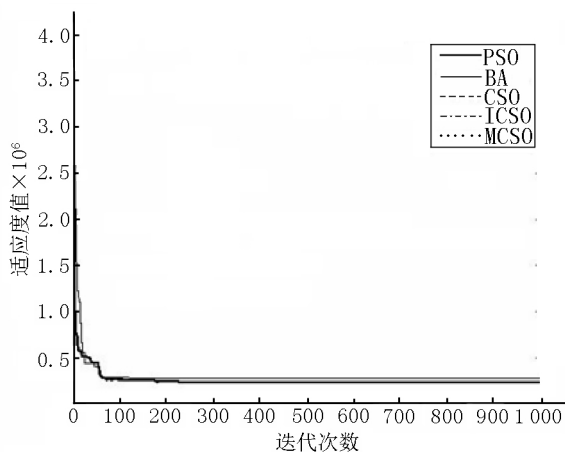


图 3 5 种算法对 F1 的收敛曲线

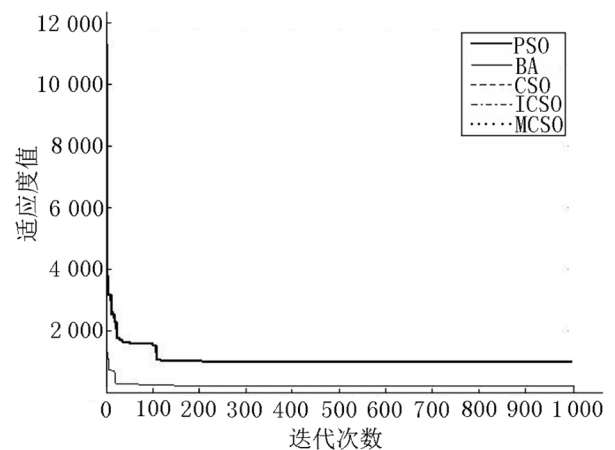


图 5 5 种算法对 F3 的收敛曲线

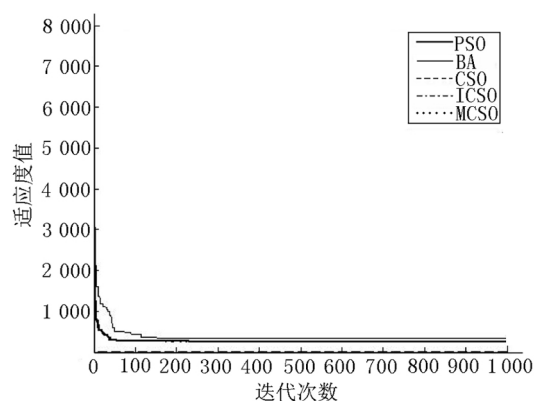


图6 5种算法对F4的收敛曲线

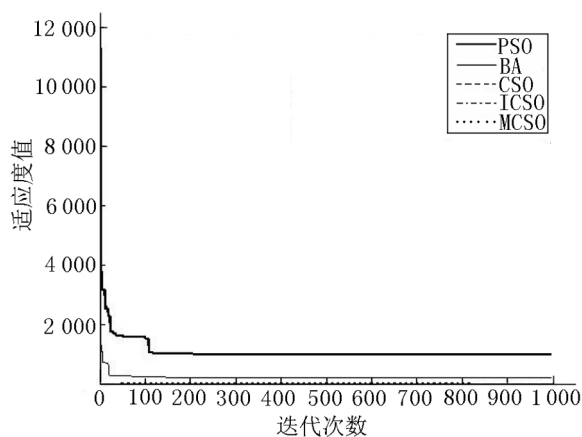


图7 5种算法对F5的收敛曲线

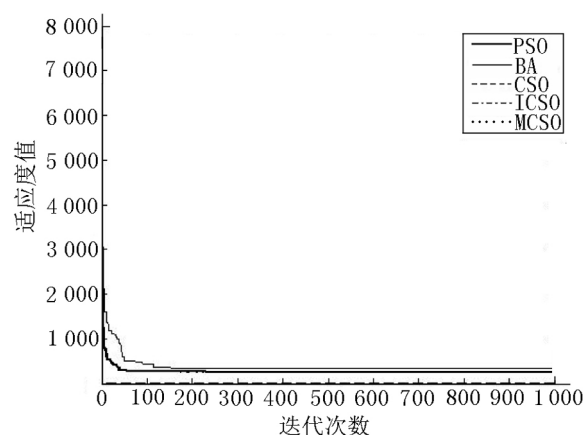


图8 5种算法对F6的收敛曲线

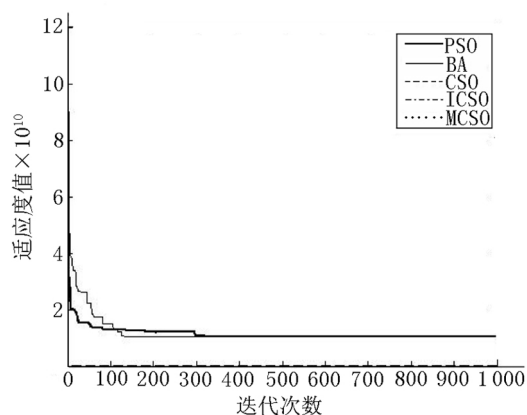


图9 5种算法对F7的收敛曲线

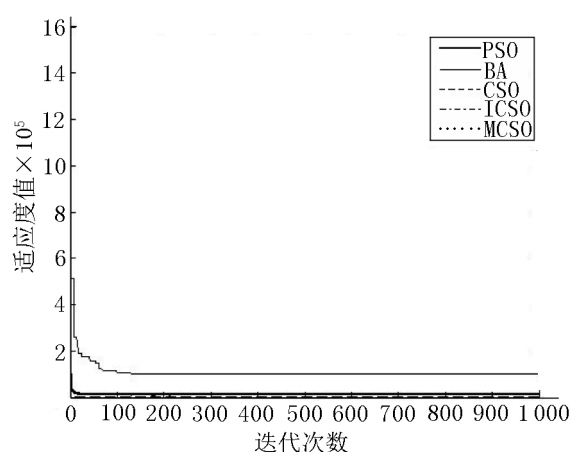


图10 5种算法对F8的收敛曲线

4 结束语

MCSO 利用惯性粒子,使算法在全局搜索方面有了较大的提高.算法利用模糊推理来改进母鸡和小鸡的更新公式,使算法在全局搜索和局部搜索方面有了较大地提升.然后利用 Tent 混沌映射使算法后期的多样性提高.尽管如此,算法在解决高维复杂问题时,仍存在差距,算法仍有待提高,需要进一步研究.

5 参考文献

- [1] Reynolds C W. Flocks, herds, and school: a distributed behavioral model [J]. ACM, 1998, 21(4): 25-34.
- [2] Kennedy J, Eberhart R C. Particle swarm optimization [J]. IEEE International Conference on Neural Networks, 1995, 4: 1942-1948.
- [3] Dorigo M, Maniezzo V, Colomni A. Ant system: optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41.
- [4] Yang Xinshe. A new metaheuristic bat-inspired algorithm [J]. Computer Knowledge & Technology, 2010, 284: 65-74.
- [5] Kundu R, Das S, Mukherjee S, et al. An improved particle swarm optimizer with difference mean based perturbation [J]. Neurocomputing, 2014, 129: 315-333.
- [6] Gaing Z L. A particle swarm optimization approach for optimum design of PID controller in AVR system [J]. IEEE Trans Energy Convers, 2004, 19: 384-391.
- [7] Hong Yingyi, Lin Faa-Jeng, Chen Syuan-Yi, et al. A novel adaptive elite-based particle swarm optimization applied to VAR optimization in electric power systems [J]. Mathematical Problems in Engineering, 2014(4): 1-14.
- [8] Cavuslu M A, Karakuzu C, Karakaya F. Neural identification of dynamic system on FPGA with improved PSO

- learning [J]. Appl Soft Comput 2012 ,12:2707-2718.
- [9] Meng Xianbing ,Liu Yu ,Gao Xiaozhi ,et al. A new bio-inspired algorithm: chicken swarm optimization [C] //5th International Conference in Swarm Intelligence. Hefei: Springer International Publishing 2014:86-94.
- [10] Kong Fei ,Wu Dinghui. An improved chicken swarm optimization algorithm [J]. Journal of Jiangnan University: Natural Science Edition 2015 ,14(6):681-689.
- [11] Li Zhenbi ,Wang Kang ,Jiang Yuanyuan. The study of improved chicken swarm optimization algorithm based on simulated annealing [J]. Microelectronics and Computer , 2016 ,34(2):30-38.
- [12] Cui Dongwen. Projection pursuit model for evaluation of flood and drought disasters based on chicken swarm optimization algorithm [J]. Advances in Science and Technology of Water Resources 2016 ,36(2):16-24.
- [13] Wei-der Chang ,Shih S P. PID controller design of nonlinear systems using an improved particle swarm optimization approach [J]. Communications in Nonlinear Science & Numerical Simulation 2010 ,15(11):3632-3639.
- [14] Liu Yu ,Qin Zheng ,Shi Zhewen ,et al. Center particle swarm optimization [J]. Neurocomputing ,2007 ,70(4):672-679.
- [15] Li Nai-Jen ,Wang Wen-June ,James Hsu C C ,et al. Enhanced particle swarm optimizer incorporating a weighted particle [J]. Neurocomputing 2014 ,124(2):218-227.
- [16] Feng Hsuan-Ming. Particle swarm optimization learning fuzzy system design [EB/OL]. [2016-12-17]. <http://www.nqu.edu.tw/upload/educsie/attachment/a56c6dad-1534fa61cff0a13726373e49.pdf>.
- [17] Shi Yuhui ,Eberhart R C. Fuzzy adaptive particle swarm optimization [J]. Congress on Evolutionary Computation , 2001 ,1(12):101-106.
- [18] Zhan Zhihui ,Zhang Jun ,Li Yun ,et al. Adaptive particle swarm optimization [J]. IEEE Trans Syst Man Cybern Part B: Cybern 2009 ,39:1362-1381.
- [19] Bevrani H ,Habibi F ,Bahahajyani P ,et al. Intelligent frequency control in an AC microgrid: online PSO-based fuzzy tuning approach [J]. IEEE Transactions on Smart Grid 2012 ,3(4):1935-1944.
- [20] Prado R P ,Garcia-Galan S ,Mufioz Exposito J E ,et al. Knowledge acquisition in fuzzy-rule-based systems with particle swarm optimization [J]. IEEE Trans Fuzzy Syst , 2010 ,18:1083-1097.
- [21] Meng Xianbing ,Yu Liu ,Gao Xiaozhi ,et al. A new bio-inspired algorithm: chicken swarm optimization [M]. New York: Springer International 2014:86-94.
- [22] Wen-June Wang ,Hwan-Rong Lin. Fuzzy control design for the trajectory tracking on uncertain nonlinear systems [J]. IEEE Trans Fuzzy Syst ,1999 ,7(1):53-62.
- [23] Teng Youwei ,Wen-June Wang. Constructing a user-friendly Ga-based fuzzy system directly from numerical data [J]. IEEE Trans Syst Man ,Cybern ,Part B: Cybern 2004 ,34:2061-2070.
- [24] Li Naijen ,Wang Wenjune ,Chen-chien James Hsu. Hybrid particle swarm optimization incorporating fuzzy reasoning and weighted particle [J]. Neurocomputing ,2015 ,167:188-501.
- [25] Liu Bo ,Wang Ling ,Jin Yihui ,et al. Improved particle swarm optimization combined with chaos [J]. Chaos Solitons Fractals 2005 ,25(5):1261-1271.
- [26] Kong Fei ,Wu Dinghui. An improved chicken swarm optimization algorithm [J]. Journal of Jiangnan University , 2015 ,14(6):681-688.

The Chicken Swarm Optimization Algorithm Based on Fuzzy Reasoning

SHI Xudong¹ ,GAO Yuelin^{2*}

(1. School of Mathematics and Computer ,Ningxia University , Yinchuan Ningxia 750021 ,China; 2. Research Institute of Information and System Computation Science ,Beifang University of Nationalities ,Yinchuan Ningxia 750021 ,China)

Abstract: According to the fact that CSO algorithm is often trapped in local solution when solving high dimensional and complex optimization problem ,the fuzzy reasoning has been used to modify position update formula of hens and chickens ,which enhances global search ability of chicken group. And then the weighted particle is used to improve information sharing among chicken so that local search ability is increased. Finally ,the Tent mapping has been used to perturb particle ,and the chicken swarm optimization algorithm based on fuzzy reasoning is proposed. Numerical experiments show that the algorithm can converge to the global optimal solution quickly ,and has high global optimization ability and computational accuracy.

Key words: chicken swarm optimization; fuzzy reasoning; weighted particle

(责任编辑: 冉小晓)