

多尺度正余弦优化算法

申元霞[†], 张学锋, 方馨, 汪小燕

(安徽工业大学 计算机科学与技术学院, 安徽 马鞍山 243000)

摘要: 针对标准正余弦算法进化后期的停滞问题, 本文对进化过程中的种群多样性进行了分析, 得出标准正余弦算法的种群多样性受控制因子的直接影响, 且种群多样性表达式中控制因子随迭代次数的增加而指数下降。为了改善标准正余弦算法进化后期的探索与开采, 提出了多尺度正余弦优化算法。新算法中通过自适应的多尺度控制因子来调节群体多样性从而实现多层次的搜索; 同时设计了协助种群实施局部搜索, 其种群独立进化, 个体可以直接学习主群或协助种群中的最优个体, 用来加快收敛速度和提高解的质量。将提出的新算法与改进的正余弦算法和新型群智能算法进行了对比实验, 23 个标准函数测试的统计结果表明新算法较好地平衡进化过程中的探索与开采, 提高了全局优化能力。

关键词: 正余弦算法; 停滞问题; 种群多样性; 探索与开采; 控制因子; 全局优化

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2021.0513

A Multi-Scale Sine Cosine Algorithm for Optimization Problems

Shen Yuan-xia[†], Zhang Xue-feng, Fang Xin, Wang Xiao-yan

(School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243000, China)

Abstract: In order to address the stagnation problem in the late stage of evolution of standard sine cosine algorithm (SCA), this paper makes the analysis of population diversity. The analysis results show that the control factor affects directly population diversity and is decreased exponentially with increase of iterations in the expression of population diversity. In order to improve the ability of exploration and exploitation in the late stage of evolution of SCA, a multi-scale sine cosine algorithm (MSCA) is presented. In the MSCA, an adaptive multi-scale control factor is designed to regulate population diversity for achieving the search with different layers. Meanwhile, an assisted swarm is developed to coordinate the local search for accelerating the convergence speed and improving calculation accuracy. The assisted swarm evolves independently and each individual can learn from the best experience of the mast swarm or the assisted swarm. MSCA was evaluated on 23 benchmark functions and compared with the improved versions of SCA and new swarm intelligence algorithms. The numerical results show that MSCA can better coordinate the exploitation and exploration capabilities and improve the global optimization ability.

Keywords: SCA; stagnation problem; population diversity; exploration and exploitation; control factor; global optimization

0 引言

群体智能算法是目前求解复杂优化问题的有效方法之一, 特别是近 20 年, 多种群体智能范式被提出, 有模拟动物行为的范式, 如人工蜂群 (Artificial Bee Colony, ABC) 算法、灰狼优化 (Grey Wolf Optimization, GWO) 算法等; 有受自然界现象启发的范式, 如教与学算法 (Teaching-learning-based optimization, TLBO) 算法等; 有受人类心理学和社会行为启发, 如粒子群优化 (Particle Swarm Optimization, PSO) 算法等。

2016 年学者 Mirjalili 通过正弦和余弦函数的震荡性来模拟群体趋于最优解的过程, 并提出正余弦算法 (Sine Cosine Algorithm, SCA)^[1]。SCA 具有收敛速度快、解的质量高等特点。SCA 已被证明在给定迭代次数下其收敛精度优于多种经典群智能算法^[2], 并被应用到目标跟踪^[3]、特征选择^[4]等工程领域。但是 SCA 在求解复杂问题时也会陷入局部最优, 特别在进化后期出现进化停滞现象。

为了改善 SCA 求解复杂优化问题时出现的进化停滞现象, 2017 年文献 [5] 将反向学习引入 SCA

收稿日期: 2021-03-29; 修回日期: 2021-07-29。

基金项目: 安徽高校自然科学研究项目 (KJ2019A0063); 安徽省自然科学基金项目 (1808085MF196)。

[†]通讯作者. E-mail: yuanxiashen@163.com.

来缓解后期的进化停滞; 2018 年文献 [6] 提出结合 PSO 和 Levy 飞行的 SCA, 算法中个体有条件地选择按 PSO 或 SCA 更新方程来更新自身位置, 同时对个体采用 Levy 飞行方式进行随机游走, 使其跳出局部最优; 2019 年文献 [7] 提出一种混合自适应的 SCA 算法, 算法中采用反向个体和在更新方程中添加扰动项的方式来保持群体多样性; 同年文献 [8] 提出精英混沌搜索策略的交替正余弦算法, 通过非线性对数曲线的控制参数作为调整策略来平衡算法的开发与探索能力; 同时利用精英个体的混沌搜索策略增强算法的开发能力; 2020 年文献 [9] 提出基于记忆引领 SCA 算法, 该算法中个体不再只学习群体最优解, 而是依据给定的条件可以学习个体保留的个体最优解; 同年文献 [10] 提出多策略 SCA 算法, 该算法将突变操作、交叉操作、混沌搜索和反向学习策略融合到算法中来提高算法的全局搜索能力。上述改进算法中采用了 Levy 飞行、混沌搜索、反向学习、多榜样学习等策略改善 SCA 进化后期的搜索能力, 虽然辅助算子从一定程度上增加了群体进化的能力, 但是并没有对导致 SCA 进化过程中停滞现象的原因进行理论分析。

为了探究 SCA 的进化停滞现象, 本文首先对 SCA 种群多样性的进化方程进行分析, 指出影响种群多样性的因素。在理论分析的基础上, 提出多尺度 SCA, 算法中利用多尺度的控制因子来控制群体的多样性, 从而实现协调算法的探索与利用。同时设计了一个协助种群实施局部搜索, 以提高算法的收敛速度和精度。与 4 种改进的 SCA 和 4 种群智能算法的对比实验结果表明: 多尺度 SCA 能够兼顾收敛速度和收敛精度, 统计结果优于对比算法。

1 标准正余弦算法 (SCA)

SCA 利用正弦和余弦函数的数学特性实现群体的寻优过程。与其他群体智能算法类似, SCA 对下一代位置的更新也是通过对当前最优解的信息加工。假设 X_t 表示第 t 代种群的个体, 第 $t+1$ 代种群的个体依照式 (1) 更新。

$$X_{t+1}(i) = \begin{cases} X_t(i) + a(t) \sin(r_1) \times \\ |r_2 X_t^*(i) - X_t(i)|, r_3 < 0.5 \\ X_t(i) + a(t) \cos(r_1) \times \\ |r_2 X_t^*(i) - X_t(i)|, r_3 \geq 0.5 \end{cases} \quad (1)$$

式中 X_t^* 表示当前群体最优个体。 r_1 为服从 $(0, 2\pi)$ 均匀分布的随机数。 r_2 为服从 $(0, 2)$ 均匀分布的随机数, 是赋予当前群体最优个体 X_t^* 的随机权重, 用来

调节 X_t^* 对 X_t 更新的影响。 r_3 为服从 $(0, 1)$ 均匀分布的随机数, 用于选择 \sin 函数或 \cos 函数作为更新策略。 $a(t)$ 为控制因子, 用来控制 X_t 更新步长, 并随着迭代次数变化, 其更新方程如式 (2) 所示。

$$a(t) = \lambda(1 - t/T) \quad (2)$$

式中, T 为最大迭代次数。随着迭代次数的增加, $a(t)$ 的值从 λ 递减到 0; $a(t)$ 起着平衡全局探索和局部搜索能力的作用。当 $|a(t)\sin(r_1)| < 1$ 或 $|a(t)\cos(r_1)| < 1$ 时, 利于下一代个体 X_{t+1} 趋向 X_t^* 运动, 强调算法的对最优解的利用能力; 当 $|a(t)\sin(r_1)| \geq 1$ 或 $|a(t)\cos(r_1)| \geq 1$ 时助于下一代个体 X_{t+1} 背离 X_t^* 运动, 加强算法的探索。

2 SCA 寻优性能分析

种群多样性影响着群体智能算法的寻优特性。种群多样性高利于全局探索, 但延缓收敛; 种群多样性低利于局部搜索, 但易陷入停滞。通过调控种群多样性来平衡全局和局部搜索一直是研究热点。下面将分析 SCA 种群多样性在群体进化过程中的变化规律。采用种群重心定义群体多样性的方法。

定义 1 ^[15] 第 t 代种群多样性定义为

$$I(X_{t+1}) = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M [X_{t+1}(i, j) - \frac{1}{N} (\sum_{i=1}^N X_{t+1}(i, j))]^2 \quad (3)$$

式 1 中的 r_1, r_2, r_3 均为随机数, 第 $t+1$ 时刻的个体 X_{t+1} 是随机的, 因此第 $t+1$ 时刻种群多样性 $I(X_{t+1})$ 也是随机的。期望效用 $E(I(X_{t+1}))$ 用来刻画 $I(X_{t+1})$ 的统计特征。

$$E(I(X_{t+1})) = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M E[X_{t+1}(i, j) - \frac{1}{N} (\sum_{i=1}^N X_{t+1}(i, j))]^2 \quad (4)$$

定理 1 SCA 在 $t+1$ 时刻种群多样性期望为

$$\frac{(N-1)^2}{MN^3} \sum_{i=1}^N \sum_{j=1}^M (D(X_t(i, j)) + \frac{1}{2} a^2(t) * E[|r_2 X_t^*(i, j) - X_t(i, j)|^2]) + \frac{1}{NM} I(E(X_t(i, j))).$$

证明 由式 (4) 展开可得下式

$$E[I(X_{t+1})] = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \{E[X_{t+1}(i, j) - EX_{t+1}(i, j)]^2 - 2E([X_{t+1}(i, j) - EX_{t+1}(i, j)][\frac{1}{N} \sum_{i=1}^N X_{t+1}(i, j) -$$

$$EX_{t+1}(i, j)] + E\left[\frac{1}{N} \sum_{i=1}^N X_{t+1}(i, j) - EX_{t+1}(i, j)\right]^2 \quad (5)$$

对式(5)中的第二项展开计算, 得式(6),

$$\begin{aligned} & E([X_{t+1}(i, j) - EX_{t+1}(i, j)] \\ & * [\frac{1}{N} \sum_{i=1}^N X_{t+1}(i, j) - EX_{t+1}(i, j)]) \\ & = \frac{1}{N} E([X_{t+1}(i, j) - EX_{t+1}(i, j)] \\ & * [\sum_{k=1}^N X_{t+1}(k, j) - N \cdot EX_{t+1}(i, j)]) \\ & = \frac{1}{N} \left(\sum_{k=1, k \neq i}^N E([X_{t+1}(i, j) - EX_{t+1}(i, j)] \right. \\ & \left. [X_{t+1}(k, j) - EX_{t+1}(i, j)]) + \right. \\ & \left. E[X_{t+1}(i, j) - EX_{t+1}(i, j)]^2 \right) \\ & = (1/N)(E[X_{t+1}(i, j) - EX_{t+1}(i, j)]^2) \\ & = (1/N)D(X_{t+1}(i, j)) \end{aligned} \quad (6)$$

式(6)中 $D(X_{t+1}(i, j))$ 表示第 i 个个体 X_{t+1} 第 j 维的方差. 对式(5)中的第三项展开计算, 得式(7):

$$\begin{aligned} & E\left[\frac{1}{N} \sum_{k=1}^N X_{t+1}(k, j) - EX_{t+1}(i, j)\right]^2 \\ & = D\left(\frac{1}{N} \sum_{k=1}^N X_{t+1}(k, j) - EX_{t+1}(i, j)\right) + \\ & (E\left[\frac{1}{N} \sum_{k=1}^N X_{t+1}(k, j) - E(X_{t+1}(i, j))\right])^2 \\ & = \frac{1}{N^2} \left(\sum_{k=1}^N D(X_{t+1}(k, j)) \right) + \\ & (E[X_{t+1}(i, j) - \frac{1}{N} \sum_{k=1}^N E(X_{t+1}(k, j))])^2 \end{aligned} \quad (7)$$

为了研究第 $t+1$ 时刻个体 X_{t+1} 方差与更新方程中参数的关系, 将式中的 r_1, r_3 的关系记作式(8).

$$b(r_1, r_3) = \begin{cases} \sin r_1, & r_3 > 0.5 \\ \cos r_1, & r_3 < 0.5 \end{cases} \quad (8)$$

由式(1)和式(8), 得式(9)

$$\begin{aligned} & D(X_{t+1}(i, j)) \\ & = E((X_{t+1}(i, j))^2 - (E(X_{t+1}(i, j)))^2) \\ & = E[X_t(i, j) + a(t)b(r_1, r_3) |r_2 X_t^*(i, j) - X_t(i, j)|]^2 - \\ & (E(X_{t+1}(i, j)))^2 \end{aligned} \quad (9)$$

由于

$$E(\sin(r_1)) = \int_0^{2\pi} \sin r \cdot \frac{1}{2\pi} dr = \frac{1}{2\pi} (-\cos r) \Big|_0^{2\pi} = 0$$

类似地, $E(\cos(r_1)) = 0$, 故得式(10)和(11),

$$\begin{aligned} & E(b(r_1, r_3)) \\ & = E(\sin r_1) \cdot P\{r_3 > 0.5\} + E(\cos r_1)P\{r_3 < 0.5\} \\ & = 0 \end{aligned} \quad (10)$$

$$\begin{aligned} & D(b(r_1, r_3)) \\ & = E(b^2(r_1, r_3)) \\ & = E(\sin^2 r_1) \cdot P\{r_3 > 0.5\} + E(\cos^2 r_1)P\{r_3 < 0.5\} \\ & = 0.5 \end{aligned} \quad (11)$$

由式(10)得到式(12)

$$\begin{aligned} & EX_{t+1}(i, j) \\ & = E[X_t(i, j) + a(t)b(r_1, r_3) |r_2 X_t^*(i, j) - X_t(i, j)|] \\ & = EX_t(i, j) + a(t)Eb(r_1, r_3)E|r_2 X_t^*(i, j) - X_t(i, j)| \\ & = EX_t(i, j) \end{aligned} \quad (12)$$

由式(9)和(12)得到式(13)

$$\begin{aligned} & D(X_{t+1}(i, j)) \\ & = E[X_t(i, j) + a(t)b(r_1, r_3) |r_2 X_t^*(i, j) - X_t(i, j)|]^2 - \\ & (E(X_{t+1}(i, j)))^2 \\ & = E[X_t(i, j)]^2 + 2a(t)Eb(r_1, r_3)E[X_t(i, j) \\ & |r_2 X_t^*(i, j) - X_t(i, j)|] + a^2(t)Eb^2(r_1, r_3) \\ & E[|r_2 X_t^*(i, j) - X_t(i, j)|]^2 - (E(X_{t+1}(i, j)))^2 \\ & = D(X_t(i, j)) + \frac{1}{2}a^2(t)E[|r_2 X_t^*(i, j) - X_t(i, j)|]^2 \end{aligned} \quad (13)$$

将式(6)、式(7)和式(13)代入到式(5), 得到式(14)

$$\begin{aligned} & E[I(X_{t+1})] = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \{D(X_{t+1}(i, j)) - \\ & \frac{2}{N} D(X_{t+1}(i, j)) + \frac{1}{N^2} D(X_{t+1}(i, j)) + \\ & (E(X_{t+1}(i, j)) - \frac{1}{N} \sum_{i=k}^N E(X_{t+1}(k, j)))^2\} \\ & = \frac{(N-1)^2}{MN^3} \sum_{i=1}^N \sum_{j=1}^M D(X_{t+1}(k, j)) + \\ & \frac{1}{NM} I(E(X_{t+1}(i, j))) \\ & = \frac{(N-1)^2}{MN^3} \sum_{i=1}^N \sum_{j=1}^M \{D(X_t(i, j)) + \\ & \frac{1}{2}a^2(t)E[|r_2 X_t^*(i, j) - X_t(i, j)|]^2\} + \end{aligned}$$

$$\frac{1}{NM} I(E(X_t(i, j))) \quad (14)$$

式中的 $I(E(X_t(i, j)))$ 为第 t 时刻, 个体 X_t 期望的多样性. 证毕. \square

定理 1 给出了第 $t + 1$ 代 SCA 群体多样性与算法参数的关系. 当第 t 代种群给定, 种群规模、解的维度和种群个体的取值已确定, 使得 X_t 的方差 $D(X_t(i, j))$ 和 $I(E(X_t(i, j)))$ 的值也确定. 通过式 (14) 可知, 式 (14) 中求和的第二项是 $a(t)^2$ 和 $E |r_2 X_t^*(i, j) - X_t(i, j)|$ 乘积, 由于种群取值已确定, 因此随机数 r_2 和控制因子 $a(t)$ 影响第 $t + 1$ 代群体多样性. 种群多样性与控制因子成指数关系, 当 $a(t) > 1$ 时, $a^2(t)$ 对 $a(t)$ 值是指数放大, 起到利于维持群体多样性作用. 当 $a(t) < 1$ 时, $a^2(t)$ 对 $a(t)$ 值是指数递减, 致使群体多样性快速下降.

在标准 SCA 算法中, 由式 (2) 可知控制因子 $a(t)$ 随着迭代次数从 2 下降到 0. 当 $t = T/2$ 时, $a(t) = 1$. 当 $t > T/2$ 时, $a(t) < 1$, 使得群体加速收敛. 如果陷入局部最优, 群体易出现停滞情况.

3 多尺度正余弦算法 (MSCA)

通过对标准 SCA 种群多样性的分析可知, 当控制因子 $a(t) < 1$, 将使得群体加速收敛. 标准 SCA 采用的线性递减策略, 将使得一半的进化时间都用于局部搜索, 加速群体收敛. 该策略针对单峰函数的搜索是快速有效的, 但是针对复杂多模问题, 出现进化后期探索能力不足的问题. 为解决该问题, 本文提出多尺度控制因子策略, 使得在进化过程中群体可以实现全局或局部的交替搜索. 在标准 SCA 算法中, 由于随机因子 r_2 的影响使得算法的局部搜索能力偏弱, MSCA 采用双种群策略, 主种群采用多尺度控制因子学习策略, 辅助群采用局部学习算法.

3.1 多尺度控制因子策略

假设总迭代次数为 T , 将整个进化过程划分成 k 段, 每个过程迭代次数分别设置为 T_1, T_2, \dots, T_k , 并满足 $T_1 + T_2 + \dots + T_k = T$. 在每段进化过程, 群体可以采用不同的控制因子, 多尺度控制因子定义如式 (15) 所示.

$$a(t) = f(\lambda_i, \beta_i, T_i), i \in k \quad (15)$$

其中 λ_i 和 β_i 分别表示控制因子的起始值和终止值; T_i 表示在第 i 段的迭代次数; $f(\cdot)$ 为定义的函数, 可为线性函数或非线性函数. 本文设计了两阶段多尺度控制因子如式 (16) 所示.

$$a(t) = \begin{cases} \lambda_1(1 - \frac{t}{T_1}) + \beta_1, & t < T_1 \\ \lambda_2(1 - \frac{t - T_1}{T - T_1}), & t \geq T_1 \end{cases} \quad (16)$$

当 $t < T_1$ 时, 控制因子 $a(t)$ 从 $\lambda_1 + \beta_1$ 递减到 β_1 ; 当 $t \geq T_1$ 时, 控制因子 $a(t)$ 从 λ_2 递减到 0, 其中 $\lambda_2 > \beta_1$. MSCA 通过多尺度的控制因子可以实施每个阶段探索和利用, 从而在整个进化过程中实现全局或局部的交替搜索, 使得群体避免停滞.

3.2 群体学习机制

MSCA 利用多尺度控制因子参数值的设置可以实现算法的局部搜索, 但是群体的学习榜样 (X_t^*) 受 r_2 的影响, 使得群体无法及时跟踪到 X_t^* , 从而降低收敛速度. 为了保持快速的收敛速度, MSCA 采用双种群策略, 即主群和协助群. 主群中采用多尺度控制因子学习策略; 协助群专注对已有最优解的利用学习. 协助群局部学习算法的思想来源于 PSO 算法. 在 PSO 算法中粒子同时学习群体最优粒子 $gbest$ 和个体历史最优粒子 $pbest$, 最终收敛于 $gbest$ 和 $pbest$ 的算术平均点. 为了加快收敛速度, 本文设计的局部学习算法直接学习种群 $gbest$ 和个体 $pbest$ 的算术平均点. 更新方程如下:

$$X_{t+1}(i) = \begin{cases} X_t(i) + b(t) \cdot \xi(X_t(i)) \\ - 0.5(G_t + P_t(i)), f(G_t) > f(X_t^*) \\ X_t(i) + b(t) \cdot \xi(X_t(i)) \\ - 0.5(X_t^* + P_t(i)), f(G_t) \leq f(X_t^*) \end{cases} \quad (17)$$

式中的 ξ 是服从 $[0, 1]$ 分布的随机数. $b(t)$ 为学习步长, X_t^* 表示主群群体最优个体. G_t 表示协助群群体最优个体. P_t 为个体历史最优粒子. $b(t) = 2(1 - t/T) + 2$, 式中 T 为总迭代次数.

主群和协助群独立进化, 当主群种群的最优解适应度值大于协助群的最优解适应度值时, 协助群不再学习自身的最优解而是学习主群的最优解.

3.3 MSCA 算法流程

MSCA 算法具体步骤如下.

步骤 1: 种群初始化. 种群 $S = S_m \cup S_s$, 其中 S_m 为主群, S_s 为协助群. 设 S_m 种群规模为 $N1$, S_s 种群规模为 $N2$, $N2 < N1$; 按给定搜索空间的范围随机生成初始种群 S 的个体.

步骤 2: 计算 S_m 和 S_s 个体的适应度 $f(X_i)$, 并分别将 S_m 和 S_s 适应度最大的个体赋值给 S_m 群体最优个体 X^* 和 S_s 群体最优个体 G ;

步骤3: 对于 $X_i \in S_m (i = 1, \dots, N1)$, 按照(1)式和(17)式产生个体 X_i 的新位置 $f(X_i^{new})$. 如果适应度 $f(X_i^{new}) > f(X_i)$, 则将 $X_i \leftarrow X_i^{new}$;

步骤4: 如果 $f(X^*) > f(G), G \leftarrow X^*$; 否则跳转步骤5;

步骤5: 对于 $X_i \in S_s (i = 1, \dots, N2)$, 按照(18)式产生个体 X_i 的新位置 $f(X_i^{new})$. 如果适应度 $f(X_i^{new}) > f(X_i)$, 则将 $X_i \leftarrow X_i^{new}$;

步骤6: 判断是否达到最大迭代次数 T , 如若满足条件则停止迭代, 将 G 作为结果输出, 否则转至步骤3.

4 数值实验与结果分析

4.1 测试函数和参数设置

为了验证本文算法的优化性能, 选取文献[5-10]中采用的23个测试函数, 包括单调测试函数($f_1 \sim$

f_7)、多模测试函数($f_8 \sim f_{13}$)和固定维数多模测试函数($f_{14} \sim f_{23}$).

表1 参数设置

算法	参数设置
SCA	$a(t) : 2 \rightarrow 0$
COSCA	$a(t) : 1 \rightarrow 0, \eta = 1, pr = 0.1$
HSCA	$a(t) : 2 \rightarrow 0$
MESCA	$a(t) : 2 \rightarrow 0$, 设置不同引导个体 Δ_t
TVACPSO	$b1:2.5 \rightarrow 0.5, b2:0.5 \rightarrow 2.5, \omega:0.9 \rightarrow 0.4$
GWO	$a:2 \rightarrow 0$
QABC	$bestlimit = 45$
DTLBO	$J_r = 0.3, \omega = 10, tf \in (0, 2)$
MSCA	$t > 0.5T, a(t) : 2.5 \rightarrow 0.5; t \leq 0.5T, a(t) : 1.5 \rightarrow 0$

表2 9种算法优化结果

函数	SCA	COSCA	HSCA	MESCA	PSO	GWO	ABC	DTBLO	MSCA	f_{min}
f_1	2.00E-04 (3.88E-03)	2.43E-78 3.22E-80	2.53E-06 (3.19E-06)	2.87E-73 (4.73E-73)	5.49E-06 (1.24E-06)	1.51E-31 (1.80E-31)	7.16E-05 (6.25E-06)	5.80E-76 (4.64E-76)	1.04E-75 (2.23E-74)	0
f_2	7.22E-1 (1.19E-1)	1.69E-37 (2.88E-37)	1.17E-05 (1.82E-6)	1.61E-36 (2.23E-35)	1.02E-01 (3.45E-02)	8.62E-19 (4.95E-19)	7.84E-03 (3.09E-03)	9.23E-33 (8.88E-39)	4.45E-35 (2.15E-36)	0
f_3	3.45E+02 (3.72E+02)	4.27E-16 (4.88E-17)	1.70E-01 (3.12E-01)	6.27E-27 (7.30E-27)	6.95E-01 (2.34E-01)	9.31E-09 (1.49E-08)	6.24E+01 (4.18E+01)	2.37E-13 (2.55E-13)	5.66E-35 (3.12E-35)	0
f_4	5.36E+00 (6.38 E+00)	8.54E-35 (1.75E-35)	6.55E-1 (1.46E-1)	5.61E-28 (1.25E-28)	6.95E-01 (2.34E-01)	4.18E-08 (4.43E-08)	4.40E+01 (2.34E+00)	1.21E-25 (1.75E-25)	3.25E-37 (2.02E-37)	0
f_5	3.81E+01 (1.82E+01)	2.13E+01 (2.67E+00)	2.84E+01 (4.26E-01)	2.81E+01 (5.20E-01)	5.32E+01 (3.15E-01)	2.71E+01 (5.20E-01)	9.46E+01 (3.61E+01)	2.68E+01 (5.80E-01)	1.63E-01 (2.50E-02)	0
f_6	4.02E+00 (2.42E-01)	2.98E+00 (1.12E-3)	6.03E-01 (1.21E-01)	1.82E+00 (3.02E-01)	1.88E-02 (3.52E-02)	6.28E-01 (4.02E-01)	1.52E-03 (5.21E-05)	2.28E-02 (3.09E-02)	4.11E-03 (1.21E-03)	0
f_7	3.65E-01 (3.61E-03)	3.22E-04 (1.62E-04)	5.55E-03 (2.32E-03)	2.59E-03 (1.41E-03)	6.42E-02 (5.23E-03)	3.30E-01 (2.73E-01)	3.68E-01 (3.54E-02)	1.82E-03 (4.26E-04)	2.49E-04 (1.62E-04)	0
f_8	5.62E+03 (8.49E+02)	5.98E+01 (7.53E+00)	3.21E+03 (7.15E+02)	5.14E+03 (1.00E+03)	2.24E+01 (1.23E+01)	5.14E+03 (1.00E+03)	1.88E+02 (9.49E+01)	9.24E+02 (5.90E+02)	7.16E-02 (2.63E-02)	0
f_9	8.11E-01 (9.03E-01)	0.00E+00 (0.00E+00)	8.71E+00 (5.13E+00)	0.00E+00 (0.00E+00)	3.31E+01 (1.21E+01)	0.00E+00 (0.00E+00)	6.76E+00 (2.15E+00)	1.46E+01 (9.15E+00)	0.00E+00 (0.00E+00)	0
f_{10}	2.77E-01 (3.39E-01)	0.00E+00 (0.00E+00)	1.22E-04 (3.18E-05)	9.43E-05 (3.18E-05)	2.12E+00 (5.78E-02)	1.06E-13 (3.18E-14)	1.13E-01 (8.62E-02)	4.25E-15 (1.58E-15)	0.00E+00 (0.00E+00)	0
f_{11}	4.25E-01 (1.04E-01)	0.00E+00 (0.00E+00)	2.45E-03 (3.09E-04)	7.56E-08 (1.49E-08)	9.96E-02 (3.45E-03)	3.56E-03 (6.49E-03)	4.21E-02 (3.38E-02)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0
f_{12}	8.17E-01 (5.34E-01)	3.45E-01 (5.79E-02)	6.43E-02 (8.14E-03)	5.01E-02 (2.08E-02)	1.84E+01 (1.17E+01)	8.01E-02 (2.08E-2)	8.98E-05 (1.43E-04)	3.24E+00 (6.54E-1)	6.32E-04 (1.38E-4)	0
f_{13}	2.58E+00 (2.43E-01)	2.32E+00 (3.13E-01)	1.40E+00 (8.24E-01)	2.73E+00 (3.55E-01)	8.32E+01 (4.74E+01)	8.38E-01 (3.08E-01)	1.22E-04 (5.01E-05)	1.14E-01 (5.84E-02)	6.78E-03 (8.25E-04)	0

表 2 续表

函数	SCA	COSCA	HSCA	MESCA	PSO	GWO	ABC	DTBLO	MSCA	f_{min}
f_{14}	1.511 (3.84E-01)	3.552 (2.18E-02)	0.998 (1.82E-01)	0.998 (1.02E-03)	0.998 (7.01E-14)	1.11E+01 (3.04E+00)	0.998 (6.15E-09)	0.998 (0.00E+00)	0.998 5.68E-10	1
f_{15}	1.12E-03 (2.54E-03)	7.879E-04 (6.34E-04)	5.01E-04 (1.00E-04)	5.67E-04 (1.52E-04)	5.06E-04 (3.51E-04)	4.85E-04 (4.50E-04)	5.99E-04 (1.58E-04)	7.15E-04 (5.26E-04)	4.04e-04 (4.617e-05)	3e-04
f_{16}	-1.0301 (4.46E-05)	-1.0311 (1.17e-05)	-1.0316 (0.00E+00)	-1.0311 (3.58e-06)	-1.0316 (0.00E+00)	-1.0316 (3.58E-06)	-1.0316 (0.00E+00)	-1.0316 (0.00E+00)	-1.0316 (6.027e-09)	-1.0316
f_{17}	0.399 (1.43E-02)	0.398 (0.00E+00)	0.397 (0.00E+00)	0.398 (1.14E-03)	0.397 (0.00E+00)	0.397 (2.95E-00)	0.397 (9.41E-11)	0.397 (0.00E+00)	0.398 (1.99E-02)	0.398
f_{18}	3.000 (1.56E-04)	3.000 (7.94E-09)	3.000 (1.00e-04)	3.000 (3.51E-05)	2.999 (6.58E-15)	3.000 (7.05E-06)	2.999 (1.69E-15)	2.999 (0.00E+00)	3.000 (4.757e-09)	3
f_{19}	-3.8550 (2.26E-03)	-3.8589 (1.32E-04)	-3.8627 (2.01E-04)	-3.8599 (1.04E-03)	-3.8627 (2.26E-16)	-3.8581 (2.81E-03)	-3.8427 (4.26E-16)	-3.8620 (2.26E-16)	-3.8458 (3.10E-02)	-3.86
f_{20}	-2.8734 (1.16E-01)	-3.162 (3.416E-02)	-3.3180 (5.01E-03)	-3.169 (6.26E-02)	-3.270 (5.92E-2)	-3.320 (7.03E-6)	-3.320 (6.25E-16)	-3.320 (4.25E-11)	-3.181 (0.1407)	-3.32
f_{21}	-2.236 (1.71)	-9.5834 (9.216E-04)	-10.0124 (1.99E-01)	-9.9234 (2.25E-01)	-10.1512 (2.34E-05)	-10.1520 (2.351E-04)	-10.1531 (2.96E-05)	-10.1531 (3.55e-05)	-10.1531 (1.96E-05)	-10.1532
f_{22}	-3.3982 (2.09E+00)	-10.3208 (3.12E-06)	-10.2706 (2.12E-01)	-10.3391 (6.21E-02)	-10.2706 (2.51E-06)	-10.4020 (7.29E-04)	-10.4024 (1.36E-05)	-10.4026 (1.25e-05)	-10.4026 (9.50E-05)	-10.4028
f_{23}	-3.3445 (1.320)	-10.5360 (2.42E-05)	-10.3660 (1.13E-01)	-10.5021 (3.52E-02)	-10.5360 (8.28E-07)	-10.4821 (3.12E-04)	-10.5360 (1.57E-10)	-9.1962 (2.99E+00)	-10.5362 (1.50E-05)	-10.5362
rank	7.91(9)	3.96(3)	4.83(6)	4.26(4)	5.52(8)	4.39(5)	4.83(6)	3.70(2)	1.78(1)	

本文算法 (MSCA) 与标准 SCA 和改进的 SCA 进行实验比较, 包括基于反向学习的混合 SCA (HSCA)^[7], 基于精英混沌学习 SCA(COSCA)^[8]、基于记忆导向的 SCA(MESCA)^[9]、时变加速系数 PSO 算法 (TVACPSO)^[12]、灰狼算法 (GWO)^[13]、人工蜂群算法 (ABC)^[14] 和动态教与学算法 (DTLBO)^[15] 进行比较. 各比较算法的参数设置如表 1 所示.

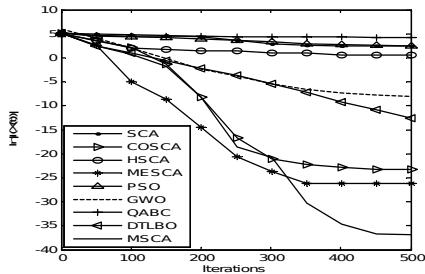
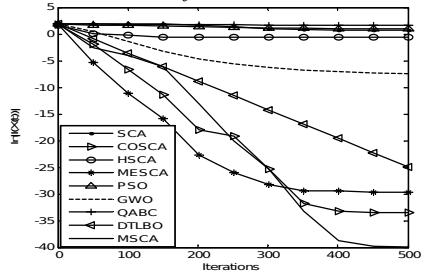
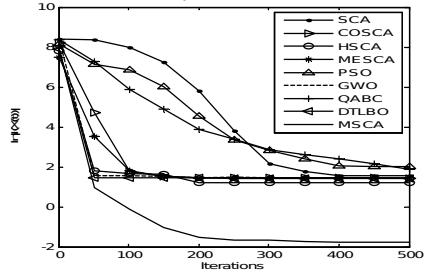
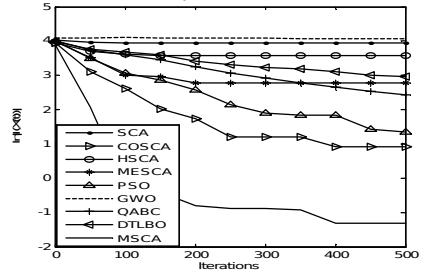
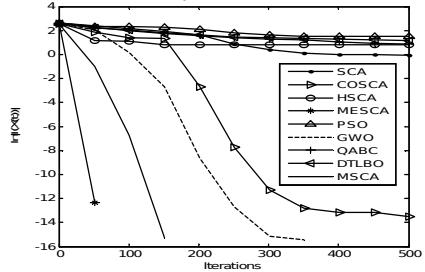
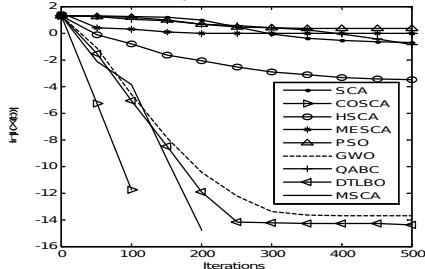
仿真实验计算机配置: Intel(R) Core(TM) i5-10210U CPU @2.30 GHz 1.60 GHz, 16 GB 内存. 在对比实验中, 所有算法均使用相同的种群规模 $N=30$, 其中 MSCA 主群规模为 20, 协助群规模为 10. 算法的最大迭代次数 $T=500$, 独立运行次数为 25 次, 维数为 30. 每种算法运行后记录其平均适应度值和标准差, 并用 Wilcoxon 秩和检验 (显著性水平 $\alpha=0.05$) 来评价 MSCA 与其他算法之间性能差异.

4.2 优化性能分析

9 种算法运行 25 次的平均适应度和标准差如表 2 所示, 对于每个函数获取的最优结果用加粗字体显示. 由表 2 可知, MSCA 已找到 5 个函数的理论最优值, 分别是 $f_9, f_{10}, f_{11}, f_{16}$ 和 f_{18} , 其中 f_9, f_{10} 和

f_{11} 为复杂多模函数, 存在多个局部极值, MSCA 可以找到理论最优值, 说明该算法具有良好的处理多模函数的能力. f_5 为 Rosenbrock 函数, 虽然是单调函数, 也只有 MSCA 的解可以到达 10^{-1} 数量级. f_8 为 Schwefel's 函数, 其复杂性是由于在远离理论最优值的地方, 存在一个深度很深的局部值, 致使很多优化算法优化陷入局部极小值, MSCA 找到的解可以到达 10^{-1} 数量级, 这说明 MSCA 具有较好的全局搜索能力. 对于 10 个固定维数的函数问题, 9 种算法均可以获得满意解, MSCA 在 7 个函数上获得了最优解, 说明 MSCA 的也具有较好的局部搜索能力.

在 $\alpha = 0.05$ 的显著性水平下, MSCA 与其他对比算法的 Wilcoxon 秩和检测的概率值 p 和检测结果 D 如表 3 所示. 结果 D 用符号“+”、“=” 和“-”分别表示了 MSCA 优于、相当和劣于某个算法.“+ / = / -”显示了算法比较的最终统计结果, DTLBO 算法的统计结果是“2/5/16”, 表示与 MSCA 相比, 有 2 个函数的优化结果优于 MSCA, 5 个函数的优化结果相当于 MSCA, 16 个函数的优化结果劣于 MSCA. 依据

图 1 f_3 收敛曲线图 2 f_4 收敛曲线图 3 f_5 收敛曲线图 4 f_8 收敛曲线图 5 f_9 收敛曲线图 6 f_{10} 收敛曲线

Wilcoxon 秩和检测的结果, 可获得 9 个算法平均排名, 结果如表 2 中”rank”所示, 根据平均排名结果可知 MSCA 的综合优化性能最好, DTBLO 次之, COSCA 为第三。

为了直观地显示各算法的进化过程, 6 个测试函数的 9 种算法收敛曲线如图 1~图 6 所示。从图中可以看出对于 6 个测试函数, SCA, HSCA 和 MESCA 算法在进化中期后逐渐停滞。虽然 HSCA 融入新的操作算子, MESCA 采用了多榜样学习的方式, 但是由于进化中期后控制因子小于 1 加速了群体的聚集, 使其陷入停滞。COSCA 算法中增加了混沌搜索和方向搜索增加进化后期的多样性。MSCA 能在进化后期能保持搜索能力, 说明多尺度控制因子策略是有效的。GWO、DTBLO、PSO 和 ABC 虽然在多数函数上进化后期没有停滞, 但是收敛速度稍缓慢。MSCA 较好平衡了算法的收敛速度和收敛精度。

4.3 算法的时间复杂度分析

设种群规模为 N , 待搜索空间的维数为 M , 适应度计算复杂度为 $O(f_s)$ 。标准的 SCA 在运行过程中群体计算适应度的复杂度为 $O(Nf_s)$ 。个体更新的计算复杂度为 $O(NM)$ 。适应度比较运算的复杂度为 $O(N)$, 所以 SCA 计算复杂度为:

$$O(SCA) = O(NM + Nf_s + N) \quad (18)$$

与标准的 SCA 相比, MESCA 在运行过程中增加个体的选择的比较运算, 其复杂度为 $O(N)$, 所以计算复杂度为:

$$O(MESCA) = O(NM + Nf_s + 2N) \quad (19)$$

HSCA 和 COSCA 算法中含有快速排序操作, 对于 N 个个体快速排序的计算复杂为 $O(N \log N)$, 最差的情况为 $O(N^2)$, 将快速排序的计算复杂度记作 O_s , $O(HSCA)$ 和 $O(COSCA)$ ^[8] 分别为:

$$O(HSCA) = O(NM + Nf_s + N + O_s) \quad (20)$$

$$O(COSCA) = O(1.1NM + Nf_s + N + O_s + O_g) \quad (21)$$

式 (21) 中, O_g 为 COSCA 精英混沌策略计算复杂度。MSCA 主群在运行过程中计算控制因子需判断当前的迭代次数, 其计算复杂度为 $O(1)$ 。评价个体适应度的计算复杂度是 $O((N_1 + N_2)f_s)$ 。主群的个体更新计算复杂度为 $O(N_1 M)$ (N_1 为主群规模), 协助群的个体更新计算复杂度为 $O(N_2 M)$ (N_2 为协助群规模), 由于 $N = N_1 + N_2$, 所以个体更新计算复杂度为 $O(NM)$ 。与标准的 SCA 相比, MSCA 的协助群增加了个体最优值更新的计算复杂度 $O(N_2)$, 所以

MSCA 计算复杂度为:

$$O(MSCA) = O(NM + Nf_s + N + N_2) \quad (22)$$

算法运行时 CPU 的总耗时可以从一定程度上反映算法的时间复杂度, 表 4 给出了 9 种算法优化每个函数运行 25 次的总耗时。从表 4 可以看出, 9 种算法中人工蜂群算法 (ABC) 的总耗时 86.266s 是最短的。SCA、MESCA 和 MSCA 的总耗时分别为 129.165s、129.363 和 129.227s。COSCA 总耗时 170.698s 为最长。

MSCA 计算开销与 SCA 和 MESCA 计算开销相仿, 低于 COSCA、HSCA 和 DTLBO 的计算开销, 但是高于 PSO、GWO 和 ABC 的计算开销。

4.4 参数设置讨论

MSCA 的控制参数 $a(t)$ 起着协调算法的探索与开发, 当 $a(t)$ 取较大的值时, 利于群体的探索。而取

较小的值时, 加强群体的开发。MSCA 采用了两阶段多尺度控制因子, 式 (15) 中含有 λ_1 、 β_1 和 λ_2 , 本节对参数取不同的值时算法的性能进行分析。 λ_1 取值为 2 和 2.5; β_1 取值为 0.5 和 1. λ_2 取值为 2 和 1.5, 形成 8 个不同的参数组合。MSCA 算法在不同的参数组合下获得函数 f_3 、 f_5 、 f_6 、 f_8 、 f_{12} 和 f_{13} 的优化结果如表 5 所示, 表中 Mean(std) 表示均值和方差。R 表示排序 (Rank)。p 表示在 $\alpha=0.05$ 的显著性水平下 Friedman 检验的概率值。由表 5 可见, 测试函数对应的 Friedman 检测值 P 均小于 0.05, 表明 5 种 pr 取值对应的算法结果之间存在显著差异。对于 8 个不同的参数组合, 其中 $\lambda_1=2.5, \beta_1=0.5, \lambda_2=1.5$ 获得的平均排名最高。

表 3 MSCA 与对比算法的检验值 P 和检验结果 D

函数	SCA		COSCA		HSCA		MESCA		TVACPSO		GWO		ABC		DBLO	
	P	D	P	D	P	D	P	D	P	D	P	D	P	D	P	D
f_1	1.41E-9	+	1.72E-4	-	1.41E-9	+	1.72E-4	+	1.41E-9	+	1.41E-9	+	1.41E-9	+	0.309	=
f_2	5.46E-6	+	1.72E-4	-	1.41E-9	+	0.622	=	1.41E-9	+	1.41E-9	+	1.41E-9	+	2.41E-6	+
f_3	1.41E-9	+	1.41E-9	+	1.41E-9	+	1.72E-4	+	1.41E-9	+	1.41E-9	+	1.41E-9	+	1.41E-9	+
f_4	1.41E-9	+	1.41E-9	+	1.41E-9	+	1.72E-4	+	1.41E-9	+	1.41E-9	+	1.41E-9	+	.41E-9	+
f_5	1.41E-9	+	1.41E-9	+	1.41E-9	+	1.41E-9	+	1.41E-9	+	1.41E-9	+	1.41E-9	+	1.41E-9	+
f_6	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	1.82E-4	+	5.46E-6	+	1.72E-4	-	5.46E-6	+
f_7	5.46E-6	+	1.82E-4	+	5.46E-6	+	3.99E-5	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	1.27E-5	+
f_8	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+
f_9	2.19E-4	+	NAN	=	4.23E-6	+	NAN	=	4.23E-6	+	NAN	=	4.23E-6	+	4.23E-6	+
f_{10}	4.23E-6	+	NAN	=	4.23E-6	+	4.23E-6	+	4.23E-6	+	2.38E-6	+	4.23E-6	+	4.23E-6	+
f_{11}	4.23E-6	+	NAN	=	4.23E-6	+	4.23E-6	+	4.23E-6	+	4.23E-6	+	4.23E-6	+	NAN	=
f_{12}	4.23E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	1.72E-4	-	5.46E-6	+
f_{13}	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	5.46E-6	+	1.72E-4	-	5.46E-6	+
f_{14}	2.49E-4	+	2.49E-4	+	5.21E-6	+	1.08E-5	+	0.568	=	5.35E-5	+	1.56E-8	+	3.35E-8	-
f_{15}	2.49E-6	+	2.49E-6	+	2.49E-4	+	2.49E-4	+	2.49E-4	+	2.49E-4	+	2.49E-4	+	2.49E-6	+
f_{16}	2.49E-6	+	2.49E-6	+	6.15E-5	-	2.49E-6	+	6.15E-5	-	6.15E-5	-	6.15E-5	-	6.15E-5	-
f_{17}	5.46E-6	+	5.15E-4	-	2.89E-9	+	6.15E-5	-	9.7E-11	+	1.41E-9	+	9.7E-11	+	9.7E-11	+
f_{18}	1.33E-2	+	0.521	=	1.33E-2	+	2.99E-3	+	9.9E-10	+	0.521	=	1.22E-9	+	3.9E-10	+
f_{19}	1.59E-4	-	1.59E-4	-	3.99E-4	-	1.59E-4	-	1.59E-4	-	1.59E-4	-	1.31E-2	+	1.59E-4	-
f_{20}	1.95E-4	+	2.19E-9	+	3.32E-4	-	2.19E-9	+	2.10E-8	-	1.14E-9	-	1.14E-9	-	1.14E-9	-
f_{21}	1.95E-4	+	1.41E-9	+	3.99E-9	+	1.41E-9	+	3.5E-10	+	0.521	=	0.521	=	0.521	=
f_{22}	1.95E-4	+	1.59E-9	+	8.61E-9	+	2.56E-8	+	8.61E-9	+	9.7E-10	+	3.1E-10	+	0.245	=
f_{23}	1.7E-10	+	1.7E-10	+	1.59E-9	+	7.7E-10	+	1.71E-10	+	8.51E-9	+	1.7E-10	+	3.99E-4	+
- / = / +	1/0/22		3/4/16		2/1/20		1/2/20		2/2/19		2/3/18		4/2/17		2/5/16	

表 4 9 种算法运行 25 次总耗时 (/s)

函数	SCA	COSCA	HSCA	MESCA	PSO	GWO	ABC	DTBLO	MSCA
f_1	5.056	6.879	5.842	5.180	4.771	3.083	2.258	8.449	5.012
f_2	5.485	7.521	7.218	5.762	5.274	3.905	3.052	8.160	5.504
f_3	8.930	12.873	12.431	8.627	8.762	7.001	5.954	16.984	8.990
f_4	4.947	6.629	6.152	4.471	4.785	4.167	3.912	10.661	4.805
f_5	6.138	8.998	8.843	6.884	5.675	4.862	4.939	12.674	6.110
f_6	5.730	7.855	7.931	5.160	5.323	4.148	3.247	9.936	5.662
f_7	6.691	8.738	8.728	6.699	5.884	4.347	3.462	10.437	6.689
f_8	5.001	6.839	5.683	5.180	4.155	3.390	3.462	9.990	5.160
f_9	5.425	7.131	6.187	5.447	4.734	3.403	3.181	12.156	5.502
f_{10}	6.845	7.367	7.921	6.282	5.652	4.104	3.540	11.070	6.976
f_{11}	5.530	7.148	6.397	5.568	5.457	4.750	3.714	10.906	5.618
f_{12}	6.236	8.192	7.690	6.374	5.959	4.176	3.967	11.291	6.323
f_{13}	6.533	8.241	7.585	6.556	5.294	6.692	5.752	10.729	6.518
f_{14}	6.556	8.842	7.693	6.561	5.798	5.285	3.842	11.987	6.560
f_{15}	4.003	6.473	6.196	4.019	3.851	3.282	2.168	7.401	4.005
f_{16}	5.091	6.843	6.630	5.096	4.899	3.029	2.310	8.980	5.104
f_{17}	4.077	5.832	5.447	4.081	3.691	2.836	3.148	8.974	4.002
f_{18}	4.267	6.688	6.196	4.273	3.982	2.960	3.264	7.682	4.226
f_{19}	4.789	5.439	5.375	4.796	4.728	2.192	2.180	8.295	4.740
f_{20}	4.381	5.352	5.864	4.488	4.372	3.085	4.092	8.649	4.346
f_{21}	5.942	7.349	7.831	6.045	5.262	3.306	4.596	8.522	5.945
f_{22}	5.889	6.293	6.730	5.990	5.543	3.118	4.960	10.085	5.802
f_{23}	5.623	7.176	6.965	5.824	5.628	4.056	5.266	10.682	5.628
总时	129.165	170.698	163.535	129.363	119.479	91.177	86.266	234.700	129.227

表 5 参数的设置对算法的影响

参数设置	F3 (p=7.15E-3)		F5 (p=1.56E-4)		F6 (p=6.83E-3)		F8 (p=7.21E-4)		F12 (p=1.25E-4)		F13 (p=7.15E-3)	
	Mean(std)	R	Mean(std)	R	Mean(std)	R	Mean(std)	R	Mean(std)	R	Mean(std)	R
$\lambda_1=2.5, \beta_1=1$	7.15E-30	5	6.71E+0	7	6.73E-2	7	1.02E+1	5	9.25E-4	4	8.95E-2	6
$\lambda_1=2$	(1.42E-29)		(7.12E+0)		(6.54E-2)		(1.11E+1)		(1.54E-3)		(9.63E-2)	
$\lambda_1=2.5, \beta_1=0.5$	4.16E-32	3	4.72E-1	2	7.62E-3	3	1.93E-1	4	6.89E-4	2	9.78E-3	4
$\lambda_1=2$	(1.21E-33)		(7.02E-1)		(8.54E-3)		(1.65E-1)		(7.34E-4)		(8.54E-3)	
$\lambda_1=2, \beta_1=0.5$	7.82E-31	4	8.87E-1	4	9.18E-3	4	8.62E-2	3	1.50E-3	5	8.63E-3	2
$\lambda_1=2$	(1.42E-31)		(6.11E-1)		(7.86E-3)		(7.43E-2)		(1.01E-3)		(8.31E-3)	
$\lambda_1=2.5, \beta_1=1$	1.71E-25	7	6.58E+0	6	5.11E-2	5	4.17E+1	7	1.21E-3	6	5.56E-1	7
$\lambda_1=1.5$	(3.32E-24)		(3.23E+0)		(5.42E-2)		(2.46E+1)		(9.25E-4)		(4.70E-1)	
$\lambda_1=2, \beta_1=1$	4.56E-23	8	8.57E+0	8	5.46E-2	6	8.63E+1	8	2.32E-3	7	7.43E-1	8
$\lambda_1=1.5$	(6.34E-23)		(6.91E+0)		(4.34E-2)		(7.86E+1)		(1.26E-3)		(6.32E-1)	
$\lambda_1=2.5, \beta_1=0.5$	5.66E-35	1	1.63E-1	1	4.11E-3	1	7.16E-2	1	6.32E-4	1	6.78E-3	1
$\lambda_1=1.5$	(3.12E-35)		(2.50E-2)		(1.21E-3)		(2.63E-2)		(1.38E-4)		(8.25E-4)	
$\lambda_1=2, \beta_1=0.5$	8.92E-35	2	5.52E-1	3	7.31E-3	2	8.03E-2	2	7.28E-4	3	9.11E-3	3
$\lambda_1=1.5$	(9.21E-35)		(3.24E-1)		(6.28E-3)		(7.32E+1)		(5.12E-4)		(8.16E-3)	

5 结论

本文首先给出了标准 SCA 进化过程中群体多样性期望, 并指出了控制因子的设置是导致在进化后期多样性缺失的重要因素之一. 接着提出了多尺度

SCA(MSCA), 新算法中通过自适应的多尺度控制因子来调节群体多样性从而实现多层次的搜索, 同时设计了协助种群实施局部搜索兼顾加快收敛速度和提高解的质量. 将提出的新算法与改进的正余弦算法和新型群智能算法进行了对比, 实验统计结果表

明新算法较好地平衡进化过程中的探索与开采,提高了全局优化的能力.

参考文献 (References)

- [1] Mirjalili S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems[J]. Knowledge-Based Systems, 2016, 96:120-133.
- [2] Abualigah L, Diabat A. Advances in Sine Cosine Algorithm: A comprehensive survey[J]. Artificial Intelligence Review, 2021, 54:2567–2608.
- [3] Nenavath H, Kumar R J , Das S. A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking[J]. Swarm Evolution Computation, 2018, 43: 1–30.
- [4] Mouhou B, Mohamed T, Farid N, et al. An Improved Sine Cosine Algorithm to select features for text categorization[J]. Journal of King Saud University-Computer and Information Sciences, 2020, 32:454–464.
- [5] Abd Elaziz M , Oliva D , Xiong S W. An Improved Opposition-Based Sine Cosine Algorithm for Global Optimization[J]. Expert Systems with Applications, 2017, 90:484-500.
- [6] Saeed N C, Ahmad B, Farid N. PSOSCALF: A new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems [J]. Applied Soft Computing Journal, 2018, 73:697–726.
- [7] Gupta S , Deep K . A hybrid self-adaptive sine cosine algorithm with opposition based learning[J]. Expert Systems with Application, 2019, 119(4):210-230.
- [8] 郭文艳, 王远, 戴芳, 刘婷基于精英混沌搜索策略的交替正余弦算法 [J]. 控制与决策, 2019, 34 (8): 1654-1661.
(Guo W Y, Wang Y, Dai F, et al. Alternating sine cosine algorithm based on elite chaotic search strategy[J]. Control and Decision, 2019, 34(8):1654-1662.)
- [9] Gupta S , Deep K , Engelbrecht A P . A memory guided sine cosine algorithm for global optimization[J]. Engineering Applications of Artificial Intelligence, 2020, 93:103718.
- [10] Chen H , Wang M , Zhao X . A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems[J]. Applied Mathematics and Computation, 2020, 369:124872.
- [11] Shi YH, Eberhart RC .Population diversity of particle swarms[C]. IEEE congress evolutionary computation .Hongkong: IEEE press, 2008:1063-1067.
- [12] Ratnaweera A , Halgamuge S K , Watson H C . Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3):240-255.
- [13] Mirjalili S, Mirjalili S M, Lewis A. Grey Wolf Optimizer[J]. Advances in Engineering Software, 2014, 69 (3): 46-61.
- [14] Aslan S , Badem H , Karaboga D . Improved quick artificial bee colony (iqABC) algorithm for global optimization[J]. Soft Computing, 2019, 23:13161–13182.
- [15] Xu Y , Yang Z , Li X , et al. Dynamic opposite learning enhanced teaching-learning-based optimization[J]. Knowledge-Based Systems, 2019, 188:1-16.

作者简介

申元霞(1979-),女,副教授,博士,从事智能算法、智能信息处理等研究,E-mail: yuanxiashen@163.com;

张学锋(1977-),男,教授,博士,从事虚拟现实技术与人工智能等研究,E-mail: zxf-06@ahut.edu.cn;

方馨(1997-),女,硕士生,从事智能算法的研究, E-mail: fangxin2345@163.com;

汪小燕(1974-),女,副教授,硕士,从事粗糙集,粒计算等研究,E-mail: wxyzjx@ahut.edu.cn.