

计算机集成制造系统
Computer Integrated Manufacturing Systems
ISSN 1006-5911, CN 11-5946/TP

《计算机集成制造系统》网络首发论文

题目: 混合随机反向学习和高斯变异的混沌松鼠搜索算法
作者: 冯增喜, 何鑫, 崔巍, 赵锦彤, 张茂强, 杨芸芸
收稿日期: 2022-01-14
网络首发日期: 2022-03-31
引用格式: 冯增喜, 何鑫, 崔巍, 赵锦彤, 张茂强, 杨芸芸. 混合随机反向学习和高斯变异的混沌松鼠搜索算法[J/OL]. 计算机集成制造系统.
<https://kns.cnki.net/kcms/detail/11.5946.TP.20220330.1331.002.html>



网络首发: 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式 (包括网络呈现版式) 排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

出版确认: 纸质期刊编辑部通过与《中国学术期刊 (光盘版)》电子杂志社有限公司签约, 在《中国学术期刊 (网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊 (网络版)》是国家新闻出版广电总局批准的网络连续型出版物 (ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

混合随机反向学习和高斯变异的混沌松鼠搜索算法

冯增喜^{1,2}, 何鑫¹, 崔巍¹, 赵锦彤¹, 张茂强¹, 杨芸芸¹

(1.西安建筑科技大学 建筑设备科学与工程学院, 陕西 西安 710055; 2.安徽建筑大学智能建筑与建筑节能安徽省重点实验室, 安徽 合肥 230022)

摘要：针对松鼠搜索算法(SSA)易陷入局部最优、过早收敛等问题，提出了一种混合随机反向学习和高斯变异的混沌松鼠搜索算法(RGCSSA)。该算法通过 Tent 混沌映射初始化策略生成混沌初始种群，增强初始种群分布的均匀性，实现对解空间更高效的搜索；采用非线性递减的捕食者概率策略，平衡 SSA 的全局搜索和局部开发能力；利用位置贪婪选择策略在迭代过程中不断保留种群中的优势个体，以提升算法收敛速度；引入随机反向学习和高斯变异策略，不但增加种群多样性，同时提高算法跳出局部最优的能力。使用 10 个不同的基准测试函数进行仿真实验并利用 Wilcoxon 符号秩检验来验证所提出算法的寻优性能，结果表明，RGCSSA 算法在求解精度和收敛速度以及稳定性等方面都有了极大的提升。

关键词：松鼠搜索算法；Tent 混沌映射；随机反向学习；高斯变异；Wilcoxon 符号秩检验
中图分类号：TP301.6 **文献标识码：**A

Hybrid random opposition-based learning and Gaussian mutation of chaotic squirrel search algorithm

FENG Zengxi^{1, 2}, HE Xin¹, CUI Wei¹, ZHAO Jintong¹, ZHANG Maoqiang¹, YANG Yunyun¹

(1. School of Building Services Science and Engineering, Xi'an University of Architecture and Technology, Xi'an, 710055, China; 2. Anhui Key Laboratory of Intelligent Building and Building Energy Conservation, Anhui Jianzhu University, Hefei 230022, China)

Abstract: To address the problems such as easy to fall into local optimum and premature convergence of squirrel search algorithm (SSA), a hybrid random opposition-based learning and Gaussian mutation of chaotic squirrel search algorithm (RGCSSA) is proposed. Firstly, the chaotic initial population is generated by the Tent chaotic mapping initialization strategy to enhance the uniformity of the initial population distribution and achieve a more efficient search of the solution space. Then, a nonlinear decreasing predator probability strategy is used to balance the global search and local exploitation capabilities of SSA. Next, the positional greedy selection strategy is utilized to increase the convergence speed of the algorithm by continuously retaining the dominant individuals in the population during the iterative process. Finally, random opposition-based learning and Gaussian variation strategies are introduced to increase the population diversity and improve the ability of the algorithm to jump out of the local optimum. The optimization performance of the proposed algorithm was verified by simulation experiments and Wilcoxon's signed rank test on 10 different benchmark functions. The results show that the RGCSSA algorithm has greatly improved in terms of solution accuracy and convergence speed as

收稿日期：2022-01-14；**修订日期：**2022-03-18。Received 14 Nov.2022;accepted 18 Mar. 2022.

基金项目：国家重点研发计划资助项目 (2017YFC0704104-03)；安徽建筑大学智能建筑与建筑节能安徽省重点实验室 2018 年度开放课题(IBES2018KF08)。**Foundation items:** Project supported by the National Key Research and Development Program, China(No.2017YFC0704104-03), and the 2018 Intelligent Building and Building Energy Efficiency Laboratory of Anhui University of Architecture,China(No.IBES2018KF08).

well as stability.

Keywords: squirrel search algorithm; tent chaotic map; random opposition-based learning; gaussian mutation; Wilcoxon's signed rank test

0 引言

随着科学技术的不断发展和创新, 各种产业领域内对工程中进行优化求解的需求日益增长, 并且优化问题的规模和复杂性也与日俱增。因此, 传统的优化方法, 诸如: 分支定界法、动态规划法^[1]、^[2]等已经难以满足实际工程的要求。近几年来, 受自然界生物行为和自然现象的启发, 国内外众多学者提出了如模拟鸟群觅食行为的粒子群算法(Particle Swarm Optimization, PSO)^[3], 模拟磷虾群在特定环境下的行为过程的磷虾群算法(Krill Herd Algorithm, KHA)^[4], 模拟蝴蝶的交配和觅食行为的蝴蝶优化算法(Butterfly Optimization Algorithm, BOA)^[5]等元启发式群智能算法。群智能算法能更好地解决实际工程当中的复杂问题, 已被广泛应用于各个领域。

松鼠搜索算法(Squirrel Search Algorithm, SSA)^[6]是 Mohit Jain 等人于 2018 年提出的, 其灵感来源于松鼠的自然动态觅食行为, 该算法模拟松鼠在不同种类的树(山核桃树、橡树、普通树、)之间滑行寻找食物来源和躲避捕食者来搜索全局最优解。与蝙蝠算法(Bat Algorithm, BA)^[7]、遗传算法(Genetic Algorithm, GA)^[8]、萤火虫算法(Firefly Algorithm, FFA)^[9]相比, SSA 具有更好的、更高效的对搜索空间的探索优势, 已被成功应用于投资组合优化^[10]、故障诊断^[11]、盲源分离^[12]等问题的解决。

但是, 松鼠搜索算法也有过早收敛、容易陷入局部最优等缺点。因此, 国内外众多学者对 SSA 进行了改进以提高算法性能, 如 Tongyi Zheng 等^[13]利用正态云模型代替松鼠随机跳跃搜索食物的行为过程, 同时利用多维搜索策略, 提高了算法的局部开发和全局搜索能力; Yanjiao Wang 等^[14]提出一种跳跃和渐进的搜索方法, 并通过线性回归选择策略自动选择进化过程中所使用的方法, 增强了 SSA 算法的鲁棒性; Xuncui Zhang 等^[15]将入侵杂草算法(Invasive Weed Optimization, IWO)的繁殖行为引入到传统 SSA 中, 并提出了一种自适应步长策略, 有效地改善了 SSA 的探索能力。这些改进策略都在不同程度上提升了 SSA 算法的寻优性能, 但该算法的性能仍具有较大的提升空间。

因此, 本文提出一种混合随机反向学习和高斯变异的混沌松鼠搜索算法(Hybrid Random opposition learning and Gaussian mutation of chaotic squirrel search algorithm, RGCSSA)以进一步提升松鼠搜索算法的寻优性能。在标准 SSA 算法中引入 Tent 混沌初始化策略、非线性递减的捕食者策略、位置贪婪选择策略、高斯变异和随机反向学习策略来提高算法的寻优性能, 并通过仿真实验来验证所提出算法的优越性和有效性。结果表明 RGCSSA 能够更好的兼顾局部开发和全局搜索, 其求解精度和收敛速度有了进一步提升。

1 松鼠搜索算法

松鼠搜索算法是模拟一种南方松鼠的动态觅食行为: 在秋天时, 它们通过在森林中不同树之间

的滑翔运动来寻找食物来源(山核桃树、橡树), 其中山核桃树为最佳食物来源.在冬天由于树叶覆盖率的减少, 导致被捕食的概率增加, 因此它们变得不再活跃, 直到冬天结束, 松鼠又再一次的活跃起来.这样一个不断重复的过程就形成了 SSA 算法.在建立该过程的数学模型之前, 考虑以下假设:

1)假设森林中有 n 只松鼠, 并存在 n 棵树, 每只松鼠停留在一棵树上。

2)假设森林里只有三种类型的树, 如山核桃树(山核桃仁为食物来源), 橡树(橡子坚果为食物来源), 普通树(无食物来源)。

3)假设森林中只有一棵山核桃树, N_{FS} ($1 < N_{FS} < n$) 棵橡树, 其余是普通树.标准 SSA 的 N_{FS} 取值为 3, N_{FS} 可随解决问题的不同而改变。

4)每只松鼠单独的寻找食物, 并通过动态觅食行为来优化利用食物来源。

在以上的假设下, 松鼠搜索算法的实现分为随机初始化位置、适应度排序及分类、松鼠位置更新、季节监测和冬末随机搬迁。

1.1 随机初始化位置

SSA 从随机初始位置开始, 松鼠的位置用一个 d 维向量表示。森林中有 n 只松鼠, 每个松鼠的初始位置通过式(1)来确定:

$$FS_{i,j} = ub_j + U(0,1) \times (ub_j - lb_j) \quad (1)$$

其中, $FS_{i,j}$ 为第 i ($i=1,2,\dots,n$) 只松鼠的第 j ($j=1,2,\dots,d$) 维的位置, ub_j 和 lb_j 分别为第 i 只松鼠在 j 维的上界和下界, $U(0,1)$ 为随机数, 它在 $[0,1]$ 范围内服从均匀分布。

1.2 适应度排序及分类

每只松鼠初始化位置向量之后, 将决策变量(松鼠的位置向量)输入到适应度函数(用户自己定义)中, 所得适应度值为 $fs = (fs_1, fs_2, \dots, fs_n)$, 其中每个松鼠的适应度值 fs_i 如下式(2)所示:

$$fs_i = fs_i(FS_{i,1}, FS_{i,2}, \dots, FS_{i,d}) \quad i=1,2,\dots,n \quad (2)$$

适应度值代表松鼠搜寻到的食物来源的质量, 将所有适应度值根据公式(3)升序排列:

$$[fitness, index] = sort(fs) \quad (3)$$

然后将松鼠进行分类, 适应度最小的松鼠在山核桃树 FS_{ht} 上, 接下来的三只松鼠($N_{FS} = 3$)在橡树 FS_{at} 上, 剩余松鼠在普通树 FS_{nt} 上.如式(4)所示:

$$\begin{aligned} FS_{ht} &= FS(index(1)) \\ FS_{at}(1:3) &= FS(index(2:4)) \\ FS_{nt}(1:n-4) &= FS(index(5:n)) \end{aligned} \quad (4)$$

1.3 松鼠位置更新

接下来松鼠便开始在不同树之间滑翔来寻找食物来源, 这种觅食行为受到捕食者存在概率 P_{dp} 的影响, 在觅食过程中松鼠的位置更新可能会出现三种情况:

1)在橡子树上的松鼠飞往山核桃树:

$$FS_{at}^{t+1} = \begin{cases} FS_{at}^t + d_g \times G_c \times (FS_{ht}^t - FS_{at}^t) & R_1 \geq P_{dp} \\ \text{Random location} & \text{otherwise} \end{cases} \quad (5)$$

其中, R_1 为随机数, 它在 $[0,1]$ 范围内服从均匀分布; FS_{ht}^t 为第 t 代山核桃树上的松鼠位置; FS_{at}^t 为第 t 代橡树上的松鼠位置; t 为当前迭代次数; $P_{dp} = 0.1$; G_c 为滑动常量, 取值为 1.9; d_g 为滑动距离, 它的取值如文献[6]所述。

2) 在普通树上的松鼠可能会向橡子树移动:

$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^t + d_g \times G_c \times (FS_{at}^t - FS_{nt}^t) & R_2 \geq P_{dp} \\ \text{Random location} & \text{otherwise} \end{cases} \quad (6)$$

其中, R_2 在 $[0,1]$ 范围内服从均匀分布; FS_{nt}^t 为第 t 代普通树上的松鼠位置。

3) 在普通树上的松鼠向山核桃树移动:

$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^t + d_g \times G_c \times (FS_{ht}^t - FS_{nt}^t) & R_3 \geq P_{dp} \\ \text{Random location} & \text{otherwise} \end{cases} \quad (7)$$

其中, R_3 也是 $[0,1]$ 之间的随机数。

1.4 季节监测和冬末随机搬迁

松鼠的觅食行为会受到季节变化的影响, 夏季松鼠活跃频繁, 到了冬季松鼠为了保存能量, 活跃性降低从而导致寻优停滞。所以算法中引入季节监测防止算法陷入局部最优。首先季节常数 S_c^t 的计算如式(8)所示:

$$S_c^t = \sqrt{\sum_{k=1}^d (FS_{at,k}^t - FS_{ht,k}^t)^2} \quad t = 1, 2, 3 \quad (8)$$

季节性常数的最小值 $S_{\min}$ 如式(9)所示:

$$S_{\min} = \frac{10E^{-6}}{(365)^{t/(t_{\max}/2.5)}} \quad (9)$$

其中, t 和 $t_{\max}$ 表示当前和最大迭代次数。当检测到 $S_c^t < S_{\min}$, 说明冬季结束, 将处在普通树上且不能找到食物来源的松鼠重新用公式(10)来随机定位它们的位置:

$$FS_{nt}^{\text{new}} = lb + Le'vy(n) \times (ub - lb) \quad (10)$$

$Le'vy$ 飞行是一种增强优化算法全局寻优能力的数学工具, 它能带领种群向偏离局部最优的方向进化, 有效避免种群陷入局部最优^[16]。在该算法中, $Le'vy(n)$ 的计算公式如下:

$$Le'vy(n) = 0.01 \times \frac{r_a \times \sigma}{|r_b|^{\frac{1}{\beta}}} \quad (11)$$

其中, r_a 和 r_b 是 $[0,1]$ 中两个正态分布随机数, β 为常数, 取值 1.5。 σ 的计算公式为:

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \right)^{1/\beta} \quad (12)$$

其中, $\Gamma(x) = (x-1)!$

1.5 算法终止条件

若满足最大迭代次数的条件, 算法结束寻优, 否则继续执行松鼠位置更新、季节监测和冬末随机搬迁。

2 混合随机反向学习和高斯变异的混沌松鼠搜索算法

尽管标准 SSA 被应用于不同领域, 但是仍存在下列问题: 标准 SSA 是在搜索空间内随机生成初始种群, 这种方式生成的初始位置不能均匀的遍历搜索空间, 导致搜索范围不广; 在寻优过程中 P_{dp} 保持不变, 不能有效地平衡算法的局部开发和全局搜索能力; 算法每次迭代松鼠都会生成新位置, 但新位置可能比旧位置更差, 种群当中的优势个体没有被保留下来, 导致算法寻优速度变慢; 寻优后期松鼠个体快速同化使得种群多样性降低, 此时算法易陷入局部最优。本文针对标准 SSA 算法存在的问题分别进行了改进。

2.1 Tent 混沌初始化

生成均匀遍布解空间的初始种群有助于群智能算法找到最优解, 提高算法的求解精度和收敛速度^[17]。而混沌映射具有随机性和遍历性的特点, 其中 Tent 混沌已经被证实比 Logistics 混沌映射有更好的均匀性和更快的迭代速度^[18]。因此本文引入 Tent 混沌映射, 它的函数定义如公式(13)所示:

$$Z_{k+1} = \begin{cases} Z_k / \beta & 0 < Z_k \leq \beta \\ (1 - Z_k) / (1 - \beta) & \beta < Z_k \leq 1 \end{cases} \quad (13)$$

其中 $\beta=0.4$ 。初始化松鼠种群的过程如下: 首先随机生成一个 $[0,1]$ 内的 d 维向量作为初始个体, 然后将该个体带入式(13)中对每一维迭代并生成 $n-1$ 新的松鼠个体, 最后将全部松鼠个体映射到变量的取值范围内生成 Tent 混沌初始化松鼠种群。采用 Tent 混沌映射生成的初始种群与随机生成的种群相比具有更好的多样性, 并能够在解空间中均匀分布, 从而改善算法容易过早收敛的缺陷, 使得算法寻优效率提高。

2.2 非线性递减的捕食者概率

在松鼠进行位置更新时, 捕食者概率 P_{dp} 会影响松鼠进行不同的搜索策略。在寻优前期需要更多地进行全局搜索, 更全面地搜索整个解空间, 在寻优后期松鼠个体都处在最优解附近, 此时算法需要更多的进行局部开发使得松鼠个体在自身附近继续去探寻更优的解。因此采用非线性递减的 P_{dp} 策略, P_{dp} 随着迭代次数动态的减少, 以平衡算法的全局搜索和局部开发的能力。如下式(14)所示:

$$P_{dp} = (P_{\max} - P_{\min}) \times (1 - t / t_{\max})^5 + P_{\min} \quad (14)$$

其中, $P_{\max}$ 和 $P_{\min}$ 分别是捕食者概率的最大值和最小值, 取值分别为: 0.15 和 0.02。

2.3 位置贪婪选择

在松鼠迭代产生新的位置时, 新位置的食物源质量可能比原位置的食物源质量差。所以可以对松鼠的位置进行贪婪选择: 将每次迭代生成的新位置和原位置的适应度值进行比较, 如果新位置的

适应度值优于原位置，则松鼠位置将由新位置来更新。否则，松鼠位置不发生变化。具体过程如下式(15)所示：

$$FS_i^{t+1} = \begin{cases} FS_i^{new} & \text{if } f_i^{new} < f_i^{old} \\ FS_i^{old} & \text{if } f_i^{new} > f_i^{old} \end{cases} \quad (15)$$

2.4 随机反向学习和高斯变异

在标准 SSA 算法的一次迭代中，较优位置的松鼠个体总是朝着最优位置松鼠个体(山核桃树上)的方向移动，而最优松鼠个体的位置是固定不变的，其所具有的有价值的位置信息没有被充分利用。因为在最优松鼠个体附近可能存在着更优的位置，所以需要在最优位置的松鼠附近进行局部搜索，以便搜索到更优的位置，从而加快算法收敛速度。

高斯变异已被证实可以增强元启发式算法的局部搜索能力^[19]。因此，在 SSA 算法每次迭代开始时，对山核桃树上的松鼠位置进行高斯变异，并比较变异前后的位置，选择适应度最好的位置作为山核桃树上松鼠的位置。如式(16)所示：

$$FS_{ht}^G = FS_{ht} \times [1 + Gauss(0,1)] \quad (16)$$

其中， FS_{ht}^G 表示山核桃树上松鼠个体变异后的位置， $Gauss(0,1)$ 是满足高斯分布的随机变量。

随机反向学习是 Long W 等人^[20]提出的一种策略，它根据当前解产生一个随机反向解，该策略可以增强种群多样性，提高种群避免局部最优的能力。在标准 SSA 算法寻优后期，松鼠个体快速同化会使得种群多样性骤减，导致算法陷入局部最优。为解决这一问题，选择当前适应度最好的松鼠位置，利用随机反向学习策略产生一个随机反向位置，利用贪婪策略选择适应度更好的位置作为最优松鼠位置。如式(17)所示：

$$U_{best}^t = lb + ub - rand() \times X_{best}^t \quad (17)$$

其中， X_{best}^t 表示当前最优松鼠位置， U_{best}^t 表示随机反向的松鼠位置。

通过这两种策略，能引导松鼠种群更好的搜索整个解空间，使得 SSA 算法用更少的迭代次数达到指定的收敛精度，提升了算法的收敛速度。同时增强 SSA 算法跳出局部最优的能力，获得更高的收敛精度。

2.5 RGCSSA 算法的流程

结合上述的几种改进策略可知 RGCSSA 算法的流程如下。

步骤 1：设置 SSA 参数，种群大小 $n = 50$ 和最大迭代次数 $t_{max} = 500$ ， $P_{max} = 0.15$ ， $P_{min} = 0.02$ 。

步骤 2：利用式(13)产生 Tent 混沌的初始松鼠种群。

步骤 3：对所有松鼠个体按照适应度值排序，按式(4)将松鼠个体分配在山核桃树、橡树、普通树上，保存最优松鼠个体和松鼠位置。

步骤 4：根据式(16)对山核桃树上的松鼠进行高斯变异，比较变异前后的适应度值，将较优的松鼠个体分配在山核桃树上。

步骤 5: 按式(14)更新 P_{dp} 的值并通过式(5)(6)(7)分别计算不同树上松鼠的新位置.最后按(15)更新所有松鼠个体的位置。

步骤 6: 按式(8)更新季节常数。若满足 $S'_c < S_{\min}$ ，则用式(10)重新定位普通树上的松鼠个体。

步骤 7: 重复步骤 3，对全局最优解根据式(17)进行随机反向学习生成新的解，计算新解的适应度值，若新解的质量优于之前的解，将新解更新为全局最优解。

步骤 8: 若达到最大迭代次数则停止寻优并输出最优松鼠位置以及适应度值；否则继续执行步骤 4 至步骤 6。

RGCSSA 算法的流程如图 1 所示。

RGCSSA算法的流程图

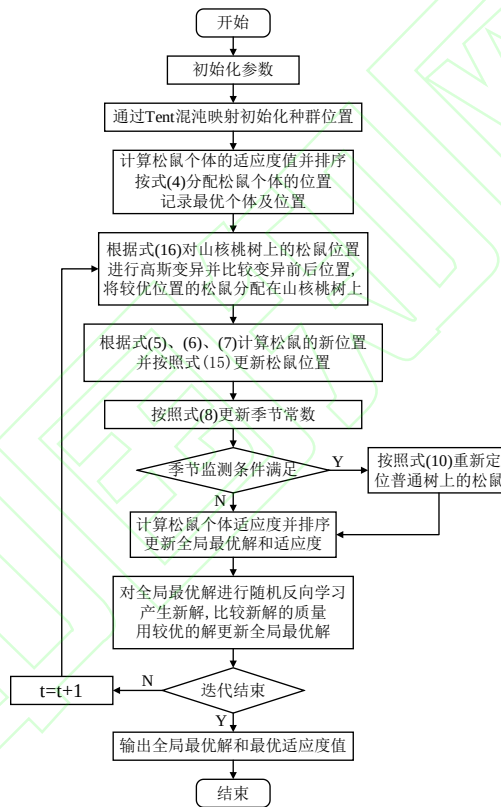


图 1 RGCSSA 算法的流程图

2.6 时间复杂度分析

时间复杂度体现了算法的运行效率，是评判算法性能优劣的重要因素^[21]。SSA 和 RGCSSA 都可以分为以下两个阶段：种群的初始化阶段和算法更新迭代阶段。在标准 SSA 中，首先假设松鼠种群的个体总数为 n ，搜索空间维度为 d ，迭代次数为 $t_{\max}$ 。所以 SSA 在初始化阶段的时间复杂度为：

$$T_1 = O(n \times d) \quad (18)$$

假设 n_1 为从橡树上往山核桃树上移动的松鼠总数， n_2 和 n_3 分别为普通树上往橡树和山核桃树移动的松鼠总数.满足季节监测条件的迭代次数为 t_1 ，所以 SSA 算法在更新迭代阶段的时间复杂度为：

$$T_2 = O[(n_1 + n_2 + n_3) \times t \times d] + ((n_2 + n_3) \times t_1 \times d) = O(n \times t \times d) \quad (19)$$

所以 SSA 的时间复杂度为:

$$T = O(n \times d) + O(n \times t \times d) = O(n \times t \times d) \quad (20)$$

在 RGCSSA 中, 生成一个 d 维随机数的时间为 ω_1 , 根据式(13)产生 Tent 混沌初始松鼠种群的时间为 ω_2 。所以 RGCSSA 在初始化阶段时间复杂度为:

$$T_3 = O(n \times (\omega_1 + \omega_2) d) = O(n \times d) \quad (21)$$

按公式(16)变异山核桃树上松鼠个体所需的时间为 ω_3 , 按公式(17)对最优松鼠个体随机反向学习所需的时间为 ω_4 , 更新 P_{dp} 的时间为 ω_5 , 位置贪婪选择的时间为 ω_6 。所以 RGCSSA 在更新迭代阶段的时间复杂度为:

$$T_4 = O[\omega_6 n \times \omega_5 t \times d] + (\omega_6 (n_2 + n_3) \times \omega_5 t_1 \times d) + ((\omega_3 + \omega_4) d \times \omega_5 t) \quad (22)$$

省略低阶项和系数之后 $T_4 = O(n \times t \times d)$ 。由上述分析可得 RGCSSA 的时间复杂度为:

$$T' = O(n \times d) + O(n \times t \times d) = O(n \times t \times d) \quad (23)$$

综上所述, RGCSSA 与标准 SSA 的时间复杂度是一致的, 从而表明 4 种改进策略的引入没有使算法的时间复杂度增加。

3 RGCSSA 算法的性能测试和分析

为了验证所改进的 RGCSSA 算法的寻优性能, 并验证不同改进策略的有效性, 本文进行了以下三个性能测试实验:

1) 将改进的松鼠搜索算法(RGCSSA)与基本的松鼠搜索算法(SSA)、粒子群算法(PSO)^[22]、磷虾群算法(KHA)^[4]、以及在全局优化问题中具有良好性能的黏菌优化算法(SMA)^[23]和蝴蝶优化算法(BOA)^[5]进行比较, 验证改进的松鼠搜索算法的竞争性和优越性。

2) 为验证 RGCSSA 算法与对比算法的寻优差异, 将 RGCSSA 算法和对比算法的寻优结果进行 Wilcoxon 符号秩检验。

3) 将 RGCSSA 算法与改进 Tent 混沌初始化的 SSA1、改进非线性递减的捕食者概率的 SSA2、结合贪婪思想的松鼠位置贪婪选择策略的 SSA3、引入随机反向学习和高斯变异策略的 SSA4 进行对比, 来验证不同的改进策略对于基本 SSA 的有效性。

3.1 测试函数的选取

在性能测试实验中选取了 10 个具有不同寻优特征的测试函数, 其中 F_1 至 F_5 是连续的单峰函数, 可测试算法的收敛精度和速度, F_6 至 F_{10} 是复杂非线性的多峰函数, 可测试算法跳出局部最优和全局搜索的能力^[19]。测试函数的具体信息如下表 1 所示。

表 1 测试函数选取

类型	函数公式	名称	理论值	维数	搜索区域
单峰	$F_1(x) = \sum_{a=1}^d x_a^2$	Sphere	0	30	[-100,100]
测试函数	$F_2(x) = \sum_{a=1}^d (x_a + 0.5)^2$	Stpe	0	30	[-100,100]

	$F_3(x) = \sum_{a=1}^{d-1} [100(x_{a+1} - x_a)^2 + (x_a - 1)^2]$	Rosenbrock	0	30	[-30,30]
	$F_4(x) = \sum_{a=1}^d ix_a^4 + random[0,1]$	Quartic	0	30	[-1.28,1.28]
	$F_5(x) = \sum_{a=1}^d x_a + \prod_{a=1}^d x_a $	Schwefel 2.22	0	30	[-10,10]
	$F_6(x) = \sum_{a=1}^d x_a \sin(x_a) + 0.1x_a $	Alpine	0	30	[-10,10]
多峰	$F_7 = \sum_{a=1}^d (-x_a \sin(\sqrt{ x_a }) + 418.9829)$	Schwefel 2.26	0	30	[-500,500]
测试	$F_8(x) = \sum_{a=1}^d [x_a^2 - 10 \cos(2\pi x_a) + 10]$	Rastrigin	0	30	[-5.12,5.12]
函数	$F_9(x) = 20 + \exp(1) - 20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{a=1}^d x_a^2}\right) - \exp\left(\frac{1}{d} \sum_{a=1}^d \cos(2\pi x_a)\right)$	Ackley	0	30	[-32,32]
	$F_{10}(x) = \frac{1}{4000} \sum_{a=1}^d x_a^2 - \prod_{a=1}^d \cos\left(\frac{x_a}{\sqrt{a}}\right) + 1$	Griewank	0	30	[-600,600]

3.2 RGCSSA 与其它智能算法的比较

为了测试本文所提算法的优越性，选择松鼠搜索算法(SSA)，改进的松鼠搜索算法(RGCSSA)，粒子群算法(PSO)^[22]，磷虾群算法(KHA)^[4]、以及在全局优化问题中具有良好性能的黏菌优化算法(SMA)^[23]和蝴蝶优化算法(BOA)^[5]进行测试函数寻优对比。

同时，为保证测试实验的客观性和公平性，不同的算法均使用相同的硬件和软件平台，硬件环境为 CPU 为 Intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz 的，软件环境为 Win10 操作系统上的 Matlab2018b 仿真软件。所有算法的种群规模设为 50，最大迭代次数设为 500。除了采用 Tent 混沌初始化改进的 RGCSSA 算法之外，其它群智能算法的初始化结果也保持一致。不同算法的其它参数设定如表 2 所示。

表 2 不同算法的参数设定

算法名称	参数
SSA	$P_{dp} = 0.1, G_c = 1.9, sf = 18$
RGCSSA	$P_{max} = 0.15, P_{min} = 0.02, G_c = 1.9, sf = 18$
PSO	$C1 = C2 = 2.5, W$ 从 0.5 线性递增至 0.9
KHA	$V_f = 0.02, D_{max} = 0.005, N_{max} = 0.01$
SMA	$Z = 0.03$
BOA	$C = 0.01, \alpha$ 从 0.1 线性递增至 0.3, $P = 0.8$

利用以上 6 种算法分别对 10 个基准测试函数进行 30 次的独立寻优实验，以便消除偶然性带来的误差。表 3 列出了 30 次独立寻优后得到的实验结果。其中最优值表示算法收敛的最佳精度，平均值表示算法的平均收敛精度，标准差代表算法收敛精度的稳定性。

表 3 测试函数结果对比

统计量	算法	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
最优值	PSO	5.79E-01	4.59E+00	5.60E+01	6.94E-03	1.21E+00	1.20E-01	2.78E+03	4.11E+01	1.54E+00	9.98E-01
	BOA	1.29E-07	3.04E+00	2.88E+01	5.05E-04	7.29E-14	6.72E-06	8.28E+03	0.00E+00	3.68E-05	3.76E-08
	SMA	0.00E+00	5.52E-04	3.51E-04	2.91E-07	6.44E-270	0.00E+00	4.43E-04	0.00E+00	8.88E-16	0.00E+00
	SSA	2.68E-09	8.11E-08	9.29E-07	2.92E-05	1.15E-04	5.29E-06	3.83E-04	1.52E-07	1.54E-04	1.42E-06
	KHA	2.08E-03	2.60E-03	2.31E+01	6.06E-03	1.28E+01	4.13E-04	3.77E+03	2.01E+00	1.89E-03	8.46E-03
	RGCSSA	0.00E+00	8.32E-31	3.25E-28	1.21E-07	2.29E-237	2.63E-237	3.82E-04	0.00E+00	8.88E-16	0.00E+00
平均值	PSO	1.32E+01	1.22E+01	1.98E+02	2.59E-02	3.55E+00	2.82E+00	4.75E+03	7.25E+01	2.50E+00	1.09E+00
	BOA	2.92E-07	4.72E+00	2.89E+01	1.79E-03	7.83E-11	2.17E-05	8.83E+03	5.82E-10	5.75E-05	1.76E-07

	SMA	0.00E+00	3.77E-03	1.40E+00	1.27E-04	2.30E-165	5.44E-163	1.97E-01	0.00E+00	1.10E-15	0.00E+00
	SSA	9.14E-04	1.01E-03	4.78E-03	5.67E-04	1.59E-02	3.16E-04	6.80E-03	1.69E-04	6.06E-03	1.46E-03
	KHA	6.85E-03	6.35E-03	2.74E+01	1.91E-02	2.11E+01	1.57E-03	4.65E+03	6.13E+00	1.04E-02	1.47E-02
	RGCSSA	0.00E+00	1.27E-30	4.97E-28	7.87E-05	1.62E-208	2.61E-186	3.82E-04	0.00E+00	8.88E-16	0.00E+00
标准差	PSO	7.96E+00	4.50E+00	1.36E+02	1.64E-02	4.21E+00	2.61E+00	1.50E+03	2.18E+01	3.14E+00	6.74E-02
	BOA	7.41E-08	6.09E-01	2.48E-02	6.23E-04	2.08E-10	9.91E-06	2.60E+02	2.82E-09	9.93E-06	9.97E-08
	SMA	0.00E+00	2.07E-03	2.33E+00	1.13E-04	0.00E+00	0.00E+00	1.35E-01	0.00E+00	1.13E-15	0.00E+00
	SSA	1.69E-03	1.99E-03	1.08E-02	3.14E-04	1.37E-02	3.24E-04	1.31E-02	2.42E-04	5.72E-03	1.96E-03
	KHA	2.94E-03	2.36E-03	1.51E+00	6.59E-03	4.10E+00	1.71E-03	3.84E+02	2.39E+00	2.06E-01	2.88E-03
	RGCSSA	0.00E+00	5.44E-31	2.13E-28	8.24E-05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
平均运行时间 /s	PSO	0.149 5	0.0906	0.1271	0.3125	0.1203	0.1776	0.1635	0.1563	0.2047	0.2474
	BOA	0.551 0	0.3208	0.3964	0.8214	0.4672	0.5458	0.4406	0.4927	0.6031	0.6792
	SMA	4.808 3	2.8302	2.9885	3.6552	3.5958	3.4198	2.7443	3.7089	3.6917	4.2396
	SSA	0.325 5	0.2047	0.2271	0.4521	0.2438	0.2870	0.2354	0.2953	0.3703	0.3766
	KHA	3.390 6	3.2354	3.8089	3.2516	3.2578	3.6563	3.8901	3.8135	3.3630	4.1266
	RGCSSA	0.446 9	0.3281	0.3698	0.4443	0.3604	0.3901	0.3385	0.3298	0.4151	0.3863

由表 3 分析可得, RGCSSA 在最优值、平均值和标准差方面都表现出了比其它 5 种对比算法更强的寻优性能。对于单峰函数 F_1 , 多峰函数 F_6 和 F_8 , RGCSSA 都分别寻找到了理论最优值。在单峰测试函数 F_2 、 F_3 和多峰测试函数 F_7 中, RGCSSA 算法的 3 个评价指标比其它算法要高出几个甚至是几十个数量级, 寻优精度得到了显著提高。对于单峰函数 F_4 和多峰函数 F_7 和 F_8 , 几种算法的寻优结果相差不大, 但总体上 RGCSSA 算法的寻优性能略优于对比算法。对于单峰函数 F_5 , 虽然 SMA 算法比 RGCSSA 算法搜索到的最优值更好, 但从平均值和标准差结果来看, 表明 RGCSSA 算法的鲁棒性和稳定性均优于 SMA 算法。在函数 F_1 、 F_9 、 F_{10} 中, SMA 和 RGCSSA 算法的 3 个评价指标一致, 但在平均运行时间方面, RGCSSA 明显比 SMA 所用时间更短。不管是单峰函数还是多峰函数, RGCSSA 算法的寻优结果明显优于其它算法, 证明了 RGCSSA 算法能够更快更高效的搜索整个解空间, 并用更少的时间去搜索到更精确的解。

从平均运行时间的结果可知, 由于不同的寻优机制导致不同算法之间的运行时间差异比较大, 总体上 PSO 的运行时间最少, KHA 的运行时间最长。RGCSSA 的平均运行时间比 SSA 稍大, 这是因为改进策略的引入适当增加了算法运行时间, 它们之间的差异在实际应用时可忽略不计。这也验证了改进的 RGCSSA 和标准的 SSA 相比, 并没有增加过多的计算开销, 进一步体现了 RGCSSA 的优越性。

为了更直观的比较不同算法的收敛速度、收敛精度和跳出局部最优的能力, 描绘出 10 个基准测试函数分别运行 30 次的平均收敛曲线, 其中纵坐标为 10 的对数, 代表适应度值, 横坐标为迭代次数。图 2 至图 6 为分别为 5 个单峰函数的平均收敛曲线, 可看出 RGCSSA 的收敛精度更高, 收敛速度更快。图 7 至图 11 分别为多峰函数的平均收敛曲线, 可知 RGCSSA 算法在搜索过程中能更快速的跳出局部最优, 寻找更好质量的解。

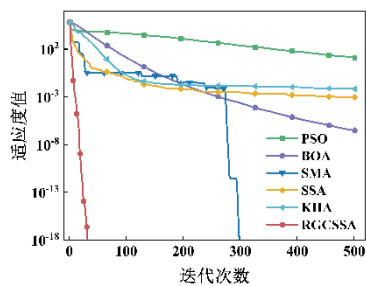


图 2 F_1 函数的平均收敛曲线

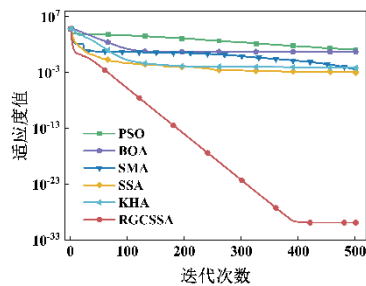


图 3 F_2 函数的平均收敛曲线

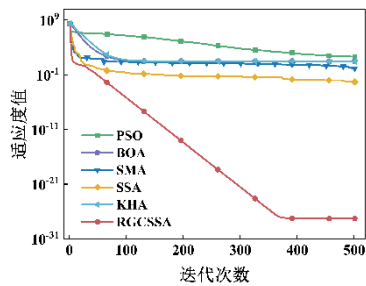


图 4 F_3 函数的平均收敛曲线

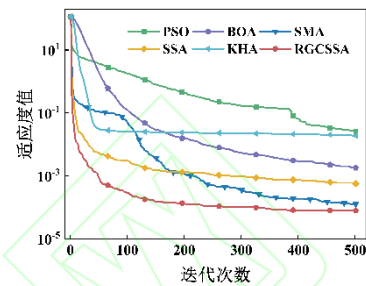


图 5 F_4 函数的平均收敛曲线

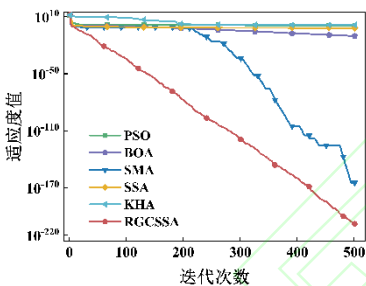


图 6 F_5 函数的平均收敛曲线

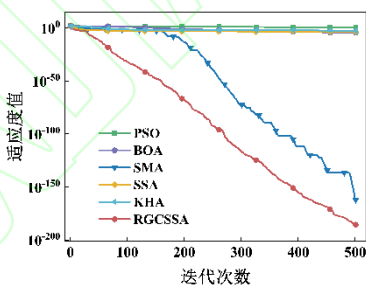


图 7 F_6 函数的平均收敛曲线

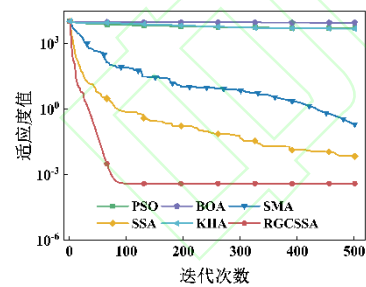


图 8 F_7 函数的平均收敛曲线

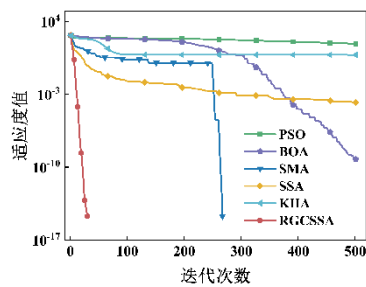


图 9 F_8 函数的平均收敛曲线

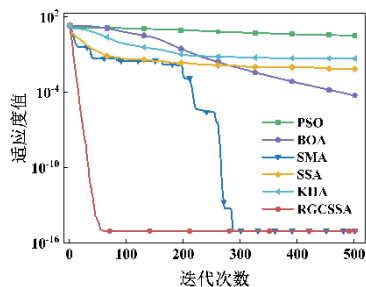


图 10 F_9 函数的平均收敛曲线

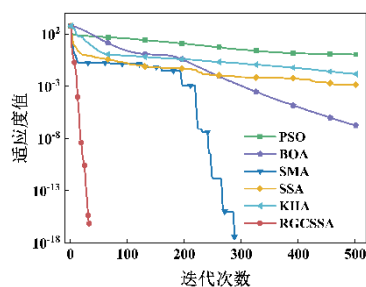


图 11 F_{10} 函数的平均收敛曲线

不论针对哪种类型的测试函数，在收敛到相同精度的情况下，RGCSSA 算法所用的迭代次数最

少。这是因为在算法寻优时，Tent 混沌初始化策略的引入使得初始松鼠种群均匀的遍布解空间，非线性递减的捕食者策略平衡了算法的局部开发和全局搜索的能力，松鼠种群中的高质量个体通过位置贪婪选择策略被保留下来，以此来提高算法收敛速度。PSO、SSA、BOA、KHA、SMA 的收敛曲线随着迭代次数的增加逐渐平缓，有的甚至存在停滞，陷入局部最优。而 RGCSSA 的收敛曲线不断的阶梯式下降，表明随机反向学习和高斯变异策略能够帮助算法有效的跳出局部最优状态，继续搜寻更精确的解。

3.3 Wilcoxon 符号秩检验

上述仿真实验中根据 30 次寻优结果的平均值和标准差来评价算法的性能，并没有将不同算法每次运行的结果相互比较。为了进一步体现算法的稳定性并保证公平性，采用统计学检验的方法来评估改进算法相比于其它算法的优越性。Wilcoxon 符号秩检验是用来比较两个独立样本之间是否具有差异的方法。

为了比较 RGCSSA 算法与其它算法之间是否具有显著差异，将 RGCSSA 算法 30 次独立寻优的结果和其它算法的寻优结果在显著性水平为 5%下进行 Wilcoxon 符号秩检验。其中 P 值小于 0.05 说明对比算法之间有显著性差异，Na 说明对比算法的寻优结果相近，不能进行显著性判断。“R”的结果为符号“+”、“-”和“=”分别表示 RGCSSA 的性能优于、劣于、相当于对比算法^[24]。最终的检验结果如表 4 所示。

表 4 Wilcoxon 秩和检验 P 值										
函数	RGCSSA-PSO		RGCSSA-BOA		RGCSSA-SMA		RGCSSA-SSA		RGCSSA-KHA	
	P	R	P	R	P	R	P	R	P	R
F_1	1.21E-12	+	1.21E-12	+	Na	=	1.21E-12	+	1.21E-12	+
F_2	1.78E-11	+	1.78E-11	+	1.78E-11	+	1.78E-11	+	1.78E-11	+
F_3	1.78E-11	+	1.78E-11	+	1.78E-11	+	1.78E-11	+	1.78E-11	+
F_4	3.02E-11	+	3.02E-11	+	1.22E-10	+	1.56E-08	+	3.02E-11	+
F_5	3.02E-11	+	3.02E-11	+	3.48E-08	+	3.02E-11	+	3.02E-11	+
F_6	3.02E-11	+	3.02E-11	+	8.88E-01	-	3.02E-11	+	3.02E-11	+
F_7	1.21E-12	+	1.21E-12	+	1.21E-12	+	1.21E-12	+	1.21E-12	+
F_8	1.21E-12	+	1.66E-11	+	Na	=	1.21E-12	+	1.21E-12	+
F_9	1.21E-12	+	1.21E-12	+	1.21E-12	=	1.21E-12	+	1.21E-12	+
F_{10}	1.21E-12	+	1.21E-12	+	Na	=	1.21E-12	+	1.21E-12	+
+/-/-		10/0/0	10/0/0		6/3/1		10/0/0		10/0/0	

由表 4 可知，在 10 个测试函数中，RGCSSA 算法对比 PSO、BOA、SSA、KHA 的 P 值都小于 0.05，表明 RGCSSA 与这四种算法之间具有显著性差异，且 RGCSSA 显著更优。RGCSSA 算法在 6 个测试函数中优于 SMA，在 3 个测试函数和 SMA 相当，仅有一个测试函数劣于 SMA。综上所述，RGCSSA 的寻优结果相比于其它算法具有显著优势，从统计学角度进一步验证了 RGCSSA 算法的优越性。

3.4 改进策略的有效性分析

上述实验将改进的 RGCSSA 算法与其它智能算法作对比，证实了 RGCSSA 算法在其它对比算法中的优越性。为了进一步探究 4 种改进策略对 RGCSSA 算法的有效性以及对它们对性能的影响程度，将 RGCSSA 算法与改进 Tent 混沌初始化的 SSA1、改进非线性递减的捕食者概率的 SSA2、结合贪婪思想的松鼠位置贪婪选择策略的 SSA3、引入随机反向学习和高斯变异策略的 SSA4 通过表 1

中的 10 个基准测试函数进行对比, 各算法的参数设置与标准 SSA 相同, 将它们分别独立寻优 30 次的结果如表 5 所示。

表 5 不同改进策略对比结果

统计量	算法	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
最优值	SSA	5.06E-08	6.49E-07	6.62E-09	5.02E-06	5.68E-04	1.31E-05	3.84E-04	5.22E-08	6.86E-04	2.52E-06
	SSA1	1.80E-07	1.07E-07	1.54E-06	3.00E-05	8.77E-05	1.39E-07	3.82E-04	4.62E-08	4.84E-05	4.50E-07
	SSA2	4.60E-26	2.96E-23	9.03E-28	1.02E-05	3.23E-13	1.83E-14	3.82E-04	0.00E+00	8.88E-16	0.00E+00
	SSA3	1.26E-37	8.32E-31	3.25E-28	6.91E-06	1.31E-20	7.56E-22	3.82E-04	5.68E-14	7.99E-15	0.00E+00
	SSA4	0.00E+00	1.81E-09	4.74E-09	2.25E-05	1.33E-238	5.24E-237	3.82E-04	0.00E+00	8.88E-16	0.00E+00
	RGCSSA	0.00E+00	8.32E-31	3.25E-28	3.54E-06	3.17E-237	1.11E-234	3.82E-04	0.00E+00	8.88E-16	0.00E+00
平均值	SSA	1.35E-03	4.31E-04	8.35E-03	4.69E-04	1.29E-02	2.57E-04	1.11E-02	1.47E-04	7.65E-03	1.95E-03
	SSA1	9.60E-04	7.29E-05	4.50E-04	4.15E-04	9.58E-03	5.62E-05	3.17E-03	2.77E-05	5.52E-04	4.90E-04
	SSA2	2.23E-14	5.50E-09	2.61E-07	1.37E-04	2.16E-05	4.58E-05	3.82E-04	6.33E-07	8.38E-06	1.05E-07
	SSA3	1.48E-35	5.30E-30	4.57E-28	1.09E-04	4.68E-03	1.66E-15	3.82E-04	1.12E-08	9.89E-15	7.40E-18
	SSA4	0.00E+00	3.08E-06	6.60E-05	4.21E-04	4.83E-203	1.58E-183	9.06E-03	0.00E+00	8.88E-16	0.00E+00
	RGCSSA	0.00E+00	1.20E-30	4.89E-28	7.41E-05	7.92E-202	7.25E-197	3.82E-04	0.00E+00	8.88E-16	0.00E+00
标准差	SSA	3.31E-03	5.60E-04	1.76E-02	4.09E-04	1.10E-02	2.65E-04	1.91E-02	2.53E-04	8.19E-03	4.55E-03
	SSA1	1.56E-03	9.82E-05	8.55E-04	4.29E-04	9.05E-03	7.57E-05	4.71E-03	3.47E-05	7.70E-04	1.13E-03
	SSA2	6.84E-14	2.37E-08	1.39E-06	1.18E-04	7.90E-05	2.00E-04	1.95E-06	3.08E-06	3.98E-05	5.63E-07
	SSA3	2.92E-35	2.17E-29	1.65E-28	7.94E-05	2.46E-11	5.50E-15	1.08E-19	6.06E-08	1.77E-15	2.77E-17
	SSA4	0.00E+00	4.75E-06	9.98E-05	3.23E-04	0.00E+00	0.00E+00	1.31E-02	0.00E+00	9.86E-32	0.00E+00
	RGCSSA	0.00E+00	5.16E-31	2.14E-28	9.96E-05	0.00E+00	0.00E+00	1.08E-19	0.00E+00	9.86E-32	0.00E+00
平均运行时间/s	SSA	0.170 8	0.242 2	0.254 7	0.546 9	0.192 7	0.256 8	0.253 6	0.216 1	0.229 2	0.257 3
	SSA1	0.191 1	0.250 5	0.259 9	0.570 8	0.208 3	0.268 2	0.288 5	0.236 5	0.235 9	0.270 8
	SSA2	0.250 5	0.502 6	0.651 0	0.602 9	0.215 1	0.206 8	0.225 5	0.429 2	0.427 6	0.374 5
	SSA3	0.278 6	0.751 6	0.522 9	0.563 2	0.223 4	0.245 8	0.228 1	0.456 8	0.494 8	0.378 6
	SSA4	0.191 7	0.290 1	0.315 6	0.620 1	0.230 2	0.287 0	0.288 5	0.251 6	0.253 6	0.289 6
	RGCSSA	0.274 0	0.295 8	0.407 8	0.656 3	0.306 3	0.282 8	0.366 7	0.490 6	0.549 5	0.431 3

由表 5 可知, 不论是单峰函数还是多峰函数, SSA1 的三个评价指标比标准 SSA 高出一到两个数量级, 这说明采用 Tent 混沌初始化策略会使寻优算法的性能有一定提升, 因为引入 Tent 混沌初始化会让初始种群均匀遍布解空间, 使算法快速高效地搜寻解空间, 改善了算法的全局搜索能力; SSA2 的三个评价指标相比于标准 SSA 有了明显的提升, 这是因为使用非线性递减的捕食者概率能动态的平衡算法的全局勘探和局部开发, 在寻优前期更多地进行全局搜索, 使得算法更全面地搜索整个解空间, 提高了算法的收敛速度, 在寻优后期松鼠个体都处在最优解附近, 此时算法更多的进行局部开发可使松鼠个体在自身附近继续去探寻更优的解, 增加了算法的收敛精度。而 SSA3 和 SSA4 对与不同类型的函数有着不同的提升效果。从三个评价指标来看, 在单峰函数中 SSA3 的寻优性能更好, 说明了位置贪婪选择策略的引入, 在迭代的过程中不断保留高质量的松鼠个体, 加快了算法的收敛速度; 在多峰函数中, SSA4 的寻优性能更好, 说明加入随机反向学习和高斯变异的策略, 使得算法在寻优后期出现停滞时能较快的跳出局部最优, 去寻找更精确地解, 从而提高了算法的收敛精度。

综上所述, 在 SSA 中分别引入 4 种改进策略, 在寻优结果的最优值、平均值、标准差上都存在不同程度的提升。但单一的策略改进对 SSA 的提升是有限的, 不符合实际工程当中复杂多变的要求。而融合了 4 种改进策略的 RGCSSA 算法在仿真实验当中表现出更加优秀和稳定的寻优性能。

4 结束语

本文针对松鼠搜索算法(SSA)存在过早收敛、易于陷入局部最优的问题, 提出了一种混合高斯变异和随机反向学习的混沌松鼠搜索算法(RGCSSA)。该算法在种群初始化阶段引入 Tent 混沌映射策略, 使得初始松鼠种群中均匀分布于解空间; 非线性递减的捕食者策略更好地平衡了算法的局部开发和全局搜索能力; 在迭代的过程当中, 位置贪婪选择策略的加入保留了优势个体, 提高算法收敛速度; 随机反向学习和高斯变异策略使算法不断地跳出局部最优, 提高算法的收敛精度。三个仿真

实验结果表明: RGCSSA 对基准测试函数的求解结果明显优于 SSA、BOA、KHA、PSO、SMA 算法, Wilcoxon 符号秩检验的结果也表明所提出的算法比其它对比算法具有更优的显著性差异, 通过对不同改进策略的寻优分析, 验证了不同的改进策略对算法性能提升的有效性。下一步研究工作中, 考虑将 RGCSSA 应用到不同领域内的实际工程问题中, 在实际问题中来验证算法的求解性能。

参考文献:

- [1] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.* 2011, 22(12): 2226-2236.
- [2] Narendra and Fukunaga, "A branch and bound algorithm for feature subset selection" *IEEE Trans. Comput.* 1977, C26(9): 917-922.
- [3] Kennedy J, Eberhart R. Particle swarm optimization[C]. *IEEE International Conference on Neural Networks*. Perth: IEEE, 1995: 1942-1948.
- [4] Gandomi A H, Alavi A H. Krill herd: A new bio-inspired optimization algorithm[J]. *Communications in Nonlinear Science and Numerical Simulation* (S1007-5704), 2012, 17(12): 4831-4845.
- [5] Arora, Sankalp, Singh, Satvir. Butterfly optimization algorithm: a novel approach for global optimization[J]. *Soft Computing - A Fusion of Foundations, Methodologies & Applications*. 2019, 23(3): 715-734.
- [6] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm and Evolutionary Computation*, 2019, 44: 148-175.
- [7] X.-S. Yang, "A new metaheuristic bat-inspired algorithm", in *Proceedings of the Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Springer Berlin Heidelberg, 2010, 284: 65-74.
- [8] J. H. Holland, "Genetic algorithms and the optimal allocation of trials", *SIAM Journal on Computing*, 1973, 2(2): 88-105.
- [9] I. Fister, I. Fister Jr., X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms", *Swarm and Evolutionary Computation*, 2013, 13: 34-46.
- [10] Mahdi Dhaini¹; Nashat Mansour². Squirrel search algorithm for portfolio optimization[J]. *Expert Systems with Applications*. 2021: 114968.
- [11] XIA Jun, JIA Minping. Rolling bearing fault diagnosis with a resonance-based sparse decomposition and squirrel optimization algorithm [J]. *Journal of Vibration and Shock*, 2021, 40(4): 250-254. (in Chinese).
- [12] 夏俊, 贾民平. 基于共振稀疏分解和松鼠优化算法的滚动轴承故障诊断[J]. *振动与冲击*, 2021, 40(4): 250-254.
- [13] ZHANG Xiaotong. Research on the Improvement of Squirrel Search Algorithm and Its Application in Blind Source Separation [D]. *North University of China*, 2021 (in Chinese).
- [14] 张晓通. 松鼠算法的改进及其在盲源分离中的应用研究[D]. 中北大学, 2021.
- [15] Zheng, TY (Zheng, Tongyi)¹; Luo, WL (Luo, Weili)¹. An Improved Squirrel Search Algorithm for Optimization. [J]. *Complexity*. 2019: 1-31.
- [16] Wang, YJ (Wang, Yanjiao)¹; Du, TL (Du, Tianlin)¹. An Improved Squirrel Search Algorithm for Global Function Optimization. [J]. *Algorithms*. 2019, 12(4): 80.
- [17] Xuncai Zhang, Kai Zhao, Lingfei Wang, et al. Ying Niu. An Improved Squirrel Search Algorithm With Reproductive Behavior[J]. *IEEE Access*. 2020: 101118-101132.
- [18] LI Haojun, ZHANG Pengwei, GUO Haidong. Adaptive multi-objective particle swarm optimization algorithm based on population Manhattan distance [J]. *Computer Integrated Manufacturing Systems*, 2020, 26(4): 1019-1032 (in Chinese).
- [19] 李浩君, 张鹏威, 郭海东. 基于种群曼哈顿距离的自适应多目标粒子群优化算法[J]. *计算机集成制造系*

统,2020, 26(4):1019-1032.]

[17] TAN Wenan, ZHAO Yao. Web service composition based on chaos genetic algorithm [J]. Computer Integrated Manufacturing Systems, 2018, 24(7):1822-1829(in Chinese).

[谭文安,赵尧.基于混沌遗传算法的 Web 服务组合[J].计算机集成制造系统,2018,24(7):1822-1829.]

[18] SHAN Liang,Qiang Hao,Li Jun,et al. Chaotic optimization algorithm based on Tent map[J]. Control and Decision,2005, 20(2):179-182(in Chinese).

[单梁,强浩,李军,等.基于 Tent 映射的混沌优化算法[J].控制与决策,2005,20(2):179-182.]

[19] LI Yu,PEI Yuhang,LIU Jingsen. Bat optimal algorithm combined uniform mutation with Gaussian mutation [J].Control and Decision,2017, 32(10):1775-1781(in Chinese).

[李煜,裴宇航,刘景森.融合均匀变异与高斯变异的蝙蝠优化算法[J].控制与决策,2017,32(10):1775-1781.]

[20] Wen Long,Jianjun Jiao,Ximing Liang, et al. Ming Xu.A Random Opposition-Based Learning Grey Wolf Optimizer[J].IEEE Access.2019:113810-113825.

[21] LIU Jingsen, MA Yixiang, LI Yu. Improved whale algorithm for solving engineering design optimization problems [J].Computer Integrated Manufacturing Systems, 2021, 27(7):1884-1897(in Chinese).

[刘景森,马义想,李煜.改进鲸鱼算法求解工程设计优化问题[J].计算机集成制造系统,2021,27(7):1884-1897.]

[22] FENG Zengxi, ZHANG Cong, LI Binghui. Optimization of MFAC Parameters Based on Improved Particle Swarm Optimization Algorithm [J].Control Engineering of China,2021,28(4):766-773(in Chinese).

[冯增喜,张聪,李丙辉.基于改进粒子群优化算法的 MFAC 参数寻优[J].控制工程,2021,28(4):766-773.]

[23]Li Shimin, ChenHuiling,Wang Mingjing, et al. Slime mould algorithm: A new method for stochastic optimization[J].Future Generation Computer Systems.2020:300-323.

[24]FU Hua, LIU Hao. Improved sparrow search algorithm with multi-strategy integration and its application [J].Control and Decision, 2022,37(1):87-96 (in Chinese).

[付华,刘昊.多策略融合的改进麻雀搜索算法及其应用[J].控制与决策,2022,37(1):87-96.]

作者简介:

冯增喜(1979-), 男, 陕西韩城人, 副教授, 博士, 硕士生导师, 研究方向: 空调节能优化控制、建筑能耗预测、空调负荷预测等, E-mail:fengzengxi2000@163.com;

何 鑫(1998-), 男, 陕西咸阳人, 硕士研究生, 研究方向: 智能优化算法、中央空调节能优化控制等, E-mail:765070298@qq.com;

崔 巍(1998-), 男, 内蒙赤峰人, 硕士研究生, 研究方向: 智慧城市、噪声预测控制等, E-mail:760431680@qq.com;

赵锦彤(1997-), 男, 陕西宝鸡人, 硕士研究生, 研究方向: 智能优化算法、建筑节能技术等, E-mail:1006254510@qq.com;

张茂强(1997-), 男, 陕西西安人, 硕士研究生, 研究方向: 人工智能、空调负荷预测等, E-mail:zhangmaoqiang2021@163.com;

杨芸芸(1998-), 女, 陕西咸阳人, 硕士研究生, 研究方向: 智能优化算法、建筑能耗优化等, E-mail:2483834525@qq.com。