

基于反向学习的DE-RSO混合优化算法

徐子岳, 梁晓丹

天津工业大学计算机科学与技术学院, 天津

收稿日期: 2021年11月6日; 录用日期: 2021年12月6日; 发布日期: 2021年12月13日

摘要

鼠群优化算法(Rat Swarm Optimizer, RSO)是一种可以解决全局优化问题的新型仿生优化算法, 它的灵感主要来自于自然界种鼠群追逐猎物和与猎物搏斗的行为。然而, 它具有收敛速度过慢和收敛精度不高的缺陷, 为解决这一问题, 本文提出了一种改进的鼠群优化算法——基于反向学习的DE-RSO混合优化算法(Opposition-Based Learning DE-RSO Hybrid Optimizer, OBLDE-RSO)。该算法使用了DE-RSO混合策略和反向学习策略, DE-RSO混合策略可以保持种群多样性并降低算法陷入局部最优的可能性, 反向学习策略可以针对性地扩大个体的搜索范围, 使个体以更高概率找到潜在的更加理想的求解区域。本文用29个IEEE CEC2017基准测试函数对OBLDE-RSO进行测试, 并与其他经典算法的测试结果进行对比, 实验结果表明, 该算法在收敛精度和收敛速度方面都具有良好的性能。

关键词

鼠群优化算法, 混合优化, 反向学习

Opposition-Based Learning DE-RSO Hybrid Optimizer

Ziyue Xu, Xiaodan Liang

School of Computer Science and Technology, Tiangong University, Tianjin

Received: Nov. 6th, 2021; accepted: Dec. 6th, 2021; published: Dec. 13th, 2021

Abstract

Rat Swarm Optimizer Algorithm (RSO) is a novel bio-inspired optimization algorithm that can solve global optimization problems. It is inspired by the behavior of wild rats chasing and fighting prey. However, it has the shortcomings of slow convergence speed and low convergence accuracy. To solve this problem, this paper proposes an improved rat swarm optimization algorithm—Opposition-Based Learning DE-RSO Hybrid Optimizer (OBLDE-RSO). The algorithm uses

the DE-RSO hybrid strategy and the Opposition-Based Learning strategy. The DE-RSO hybrid strategy can maintain the diversity of the population and reduce the possibility of the algorithm falling into the local optimum. The Opposition-Based Learning strategy can expand the search range of the individual in a targeted manner and make the individual find a potentially more ideal solution area with a higher probability. In this paper, 29 IEEE CEC2017 benchmark functions are used to test OBLDE-RSO and compared with the test results of other classic algorithms. The experimental results show that the algorithm has good performance in terms of convergence accuracy and convergence speed.

Keywords

Rat Swarm Optimizer Algorithm, Hybrid Optimization, Opposition-Based Learning

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

元启发算法在解决实际优化问题中得到了广泛的应用,它们具有多功能性、有效性和鲁棒性等诸多优良的特点。它们不仅可以用于处理具有连续和可微目标函数和约束的优化问题,还可以用于处理具有连续和离散变量的优化问题[1]。

现在,有许多元启发算法被提出并应用。算法可以被分为四类:基于进化的、基于物理的、基于群智能的和基于人类行为的。基于进化的算法有:遗传算法(GA) [2]、差分进化算法(DE) [3]和进化策略(ES) [4]等;基于物理的算法有:引力搜索算法(GSA) [5]、人工化学反应优化算法(ACROA) [6]和射线优化算法(RO) [7]等;基于群智能的算法有:粒子群算法(PSO) [8]、灰狼算法(GWO) [9]和狮群算法(LSO) [10]等;基于人类行为的算法有:儿童绘画发展优化算法(CDDO) [11]和头脑风暴算法(BSO) [12]等。

鼠群优化算法(Rat Swarm Optimizer, RSO) [13]是一种由 Gaurav Dhiman 等人在 2020 年提出的新型的仿生优化算法。RSO 的灵感来自于自然界种群鼠群追逐猎物和与猎物搏斗的行为。老鼠具有很丰富的社交智慧,它们通过追逐、跳跃和搏斗等各种行为相互沟通。它们具有很强的领土意识,当它们被侵犯时,就会变得非常有攻击性。RSO 在数学上模拟了这种行为,并可以有效解决全局优化问题,此外它还具有结构简单和参数较少的优点。由于 RSO 的新颖性,关于修改或应用 RSO 的文献很少,但是它已经开始被应用到医药领域[14] [15]。

然而,由于模拟鼠群追逐猎物的过程过于随机且不够精确,因此,RSO 具有速度过慢和容易陷入局部最优等缺点。

为了改善 RSO 的性能,本文提出了一种基于反向学习的 DE-RSO 混合优化算法(Opposition-Based Learning DE-RSO Hybrid Optimizer, OBLDE-RSO),该算法首先使用一种 DE-RSO 混合策略,将差分进化算法的变异机制应用到 RSO 中,以增加种群的多样性,然后用反向学习策略(Opposition-Based Learning, OBL) [16],针对性地扩大个体的搜索范围,使个体以更高概率找到潜在的更加理想的求解区域。同时,为了加速算法收敛,对 RSO 本身的参数进行了调整优化。为了评估 OBLDE-RSO 的性能,本文使用了 IEEE CEC2017 基准测试函数集对算法进行测试。

本文的组织结构安排如下,引言后,第 2 节简单介绍了 RSO 算法,第 3 节阐述了 OBLDE-RSO 算法的详细信息,第 4 节对算法进行了测试并分析了实验结果,第 5 节对全文做了总结。

2. 鼠群优化算法

鼠群优化算法(RSO)是一种新型的群智能算法, 它模拟了鼠群追逐猎物和与猎物搏斗的过程。

2.1. 追逐猎物

老鼠是一种社会性很强的动物, 它们是通过社会群体行为来追逐猎物的。为了从数学上定义这种行为, 将种群中最优个体的位置设定为猎物的位置, 其他个体通过当前最优个体更新它们的位置。这个过程被定义为公式(1):

$$\bar{P} = A \cdot \bar{P}_i(x) + \left| C \cdot (\bar{P}_r(x) - \bar{P}_i(x)) \right| \quad (1)$$

其中, \bar{P}_i 是鼠群中个体的位置, \bar{P}_r 是当前最优个体的位置, 参数 A 和 C 分别通过公式(2)和公式(3)计算:

$$A = R - t \cdot \left(\frac{R}{Max_{Iteration}} \right) \quad (2)$$

其中, $t = 0, 1, 2, \dots, Max_{Iteration}$, t 是当前迭代次数, $Max_{Iteration}$ 是最大迭代次数。

$$C = 2 \cdot rand() \quad (3)$$

参数 R 随机分布在区间[1, 5], 参数 C 随机分布在区间[0, 2]。通过调整参数 A 和 C 的取值, 以在平衡算法局部搜索和全局搜索之间取得折衷。

2.2. 与猎物搏斗

鼠群与猎物搏斗的过程可以被定义为公式(4):

$$\bar{P}_i(x+1) = \bar{P}_i(x) - \bar{P} \quad (4)$$

其中, $\bar{P}_i(x+1)$ 是更新后的下一代个体的位置。该公式保存了全局最优个体的位置并且基于最优位置引导其他个体逐渐向最优位置靠近。表 1 给出了鼠群优化算法的伪代码。

Table 1. Pseudocode of RSO algorithm

表 1. RSO 的伪代码

输入: 鼠群 $P_i (i = 1, 2, \dots, N)$
 输出: 全局最优位置

1. 初始化 RSO 的参数 A , C 和 R
2. 计算每一个个体的适应度值
3. $P_r \leftarrow$ 全局最佳位置
4. **While** 当前迭代次数 $t <$ 设定的最大迭代次数 $Max_{iteration}$ **do**
5. **For** 每一个个体 **do**
6. 通过公式(4)更新当前个体的位置
7. **End for**
8. 更新参数 A , C 和 R 。
9. 如果有个体超出了搜索空间, 对其进行修正
10. 评估每一个个体的适应度值
11. 如果当前个体适应度值优于历史全局适应度值 P_r , 则将当前位置更新为全局最佳位置
12. $t \leftarrow t + 1$
13. **End while**
14. **Return** 全局最优个体 P_r 的位置
15. **End**

3. 基于反向学习的 DE-RSO 混合优化算法

本节将 DE-RSO 混合策略和反向学习策略应用于 RSO, 提出一种基于反向学习的 DE-RSO 混合优化算法(OBLDE-RSO)。

3.1. DE-RSO 混合策略

为了维持种群的多样性, 将 DE 中的变异机制引入到 RSO 中, 对种群中的个体进行振荡, 同时在算法中引入一个参数 B 以提高算法的收敛速度。我们使用一个比例参数 pcr 来随机选择采用何种更新方式, 所以鼠群搏斗的过程可以描述为公式(5):

$$\bar{P}_i(x+1) = \begin{cases} \bar{P}_r(x) + F \cdot (\bar{P}_{r_1}(x) - \bar{P}_{r_2}(x)), pcr \leq 0.9 \text{ 且 } j = j_{rand} \\ \bar{P}_r(x) - B \cdot \bar{P}, \text{其他} \end{cases} \quad (5)$$

其中, F 为缩放参数, 随机分布在区间[0.2, 0.8]中, r_1 和 r_2 是两个互不相等的随机整数, 即 $\bar{P}_{r_1}(x)$ 和 $\bar{P}_{r_2}(x)$ 是种群中随机选择的两个不同的个体, j 是种群中个体的维度, j_{rand} 是集合 $\{1, 2, \dots, D\}$ 中的一个随机数。如公式(5), pcr 是一个比例参数, 算法设置当 $pcr \leq 0.9$ 且 $j = j_{rand}$ 时, 算法将选择利用 DE 中的变异机制来对个体进行振荡, 减小种群中个体落入局部最优的几率, 使算法尽可能找到精确的全局最优解。否则, 在公式(4)中引入一个衰减参数 B , 如公式(5), B 会随着迭代过程不断变化, 变化过程可由公式(6)和公式(7)描述:

$$B = 2 * b * rand() - b \quad (6)$$

$$b = 2 * \log_2 \left(2 - \frac{t}{Max_{iteration}} \right) \quad (7)$$

其中, t 是当前迭代次数, $Max_{iteration}$ 是最大迭代次数, 由公式(7)可知, b 会随着迭代次数的增加而由慢变快地变小, 因此由公式(6)可知, B 的变化过程与 b 是类似的, 即 B 也是由慢变快的逐渐减小的。所以, 由公式(5)可知, 当 $pcr \geq 0.9$ 时, 在算法搜索前期, 个体的搜索范围波动较大, 有更大的几率找到最优值, 随着 B 的又慢变快地减小, 在搜索的中后期, 个体的搜索范围会缩小到现有的最优值附近, 这样可以加速种群收敛于这个最优值。

3.2. 反向学习策略

学习和搜索是优化算法的基本任务, 大多数情况下, 学习是从一个随机点开始的, 也就是说, 从一开始沿着现有最优点的方向移动。例如, PSO 的权值的随机初始化, GA 中参数的随机配置等都是基于随机性。当我们要求得一个实际问题的最优点 x 时, 通常会先找一个估计点 \hat{x} 。这个估计值并不精确, 它可以是基于经验或者完全随机的猜测。在某些情况下, 估计值 \hat{x} 是令人满意的, 但是一般情况下需要进一步缩小估计值 \hat{x} 和最优点 x 的距离。从这个意义上讲, 如果将一些优化问题转化为近似函数, 那么必须求出其计算复杂度, 但是在一般的情况下, 计算的时间往往不符合实际需求。

如果估计值距离最优点很近, 可以加速算法收敛。反之, 如果离最优点很远, 并且假设在最坏情况之下, 即在最优点的相反的方向, 那么搜索过程将会消耗很大的成本。当然, 在没有任何先验知识的情况下, 很难做出理想的初始猜测。因此, 我们应该同时做出多个方向猜测, 具体地, 应该更加关注相反的方向。如果, 我们想要搜索最优点 x , 搜索相反的方向可以提高靠近最优点的几率, 那么, 算法搜索的第一步将是计算相反的点 \bar{x} [16]。

设 x 在区间 $[a, b]$ 内, 那么 x 的相反值 \bar{x} 可以定义为以下形式:

$$\bar{x} = a + b - x \quad (8)$$

这样的定义也可以应用于更加高维的情况, 假设 $P = \{x_1, x_2, \dots, x_D\}$ 是 D 维搜索空间中的一个点, 而 $x_i (1 \leq i \leq D)$ 在 a_i 与 b_i 的范围内, 相反点 $\bar{P} = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_D\}$ 可以通过以下公式(9)得出:

$$\bar{x}_i = a_i + b_i - x_i \quad (9)$$

由公式(8)和公式(9)可以看出, 随机的猜测点和猜测它的相反值可以同时以更高的概率来找到潜在的更加理想的求解区域, 其中最优解就很有可能位于其中。

在得到一个个体的相反个体后, 同时计算该个体与其相反个体的适应度值, 比较两者的适应度值, 若其相反个体的适应度值优于原个体的适应度值, 原个体将被相反个体所替代, 这样个体将会用更高的概率靠近最优个体, 计算效率和计算精确度会得到明显提高。表 2 给出了 OBLDE-RSO 的伪代码。

Table 2. Pseudocode of OBLDE-RSO algorithm

表 2. OBLDE-RSO 的伪代码

输入: 鼠群 $P_i (i=1, 2, \dots, N)$
 输出: 全局最优位置

1. 初始化 RSO 的参数 A, C, R, F, B 和 pcr
2. 计算每一个个体的适应度值
3. $P_r \leftarrow$ 全局最佳位置
4. **While** 当前迭代次数 $t <$ 设定的最大迭代次数 $Max_{iteration}$ **do**
5. **For** 每一个个体 **do**
6. 通过公式(5)更新当前个体的位置
7. **End for**
8. 更新参数 A, C, R, F 和 B
9. 如果有个体超出了搜索空间, 对其进行修正
10. 评估每一个个体的适应度值
11. 用公式(9)计算出每个个体的相反个体并计算其适应度值
12. 若相反个体的适应度值优于原个体, 则用相反个体代替原个体
13. 如果当前个体适应度值优于历史全局适应度值 P_r , 则将当前位置更新为全局最佳位置
14. $t \leftarrow t + 1$
15. **End while**
16. **Return** 全局最优个体 P_r 的位置
17. **End**

4. 实验结果与分析

本文引入了 IEEE CEC2017 基准测试函数集, 利用该测试函数集测试 OBLDE-RSO 算法。IEEE CEC2017 基准测试函数集由 29 个函数组成, 包括 2 个单峰函数(g1, g3), 7 个简单多峰函数(g4-g10), 10 个混合函数(g11-g20)和 10 个组合函数(g21-g30)。测试函数集的具体细节可参照文献[17]。

在本实验中, 为了进行公平的比较, 操作环境保持不变, 如下:

- 1) 每个函数独立运行 30 次。
- 2) 种群规模为 150, 个体维度为 30。
- 3) 函数的评估总次数为 45,000 次。
- 4) 搜索范围为[-100, 100]。

这些实验都是在一台具有 I7 处理器、windows 10 操作系统和 8 GB 内存的计算机上执行的, 所有算法都是在 Matlab2018b 中进行编码的。

为了证明 OBLDE-RSO 性能的优越性, 我们将基于反向学习的 DE-RSO 混合优化算法 OBLDE-RSO 同鼠群算法 RSO、差分进化算法 DE、狮群算法 LSO 和遗传算法 GA 进行比较。表 3 列出了这些算法的参数。

Table 3. The parameters of the OBLDE-RSO, RSO, DE, LSO and GA

表 3. OBLDE-RSO, RSO, DE, LSO 和 GA 的参数

算法	参数
OBLDE-RSO	$R \in [1,3], C \in [0,2], F \in [0.2,0.8], pcr = 0.9$
RSO	$R \in (1,5), C \in (0,2)$
DE	$F = 0.7, CR = 0.5$
LSO	$\beta = 0.7$
GA	$P_c = 0.9, P_m = 0.1$

表 4 是 OBLDE-RSO, RSO, DE, LSO 和 GA 的对比结果, 结果包括各算法在测试函数集中独立运行 30 次的平均值(Mean)和标准差(Std)。

由表 4 可知, 在大部分情况下 OBLDE-RSO 的性能优于 RSO、DE、LSO、GA。具体来说, 在 29 个测试函数中, OBLDE-RSO 在其中的 27 个对 RSO、DE、LSO、GA 表现出了更加出色的性能。OBLDE-RSO 在单峰函数、简单多峰函数和混合函数上, 在求解的精确度上比其他比较算法具有绝对的优势。在组合函数上, OBLDE-RSO 也具有十分优异的性能。这说明 DE-RSO 混合策略和反向学习策略的使用也可以提高 RSO 的性能。

另外, 在四类测试函数中各挑取一个代表函数(g1, g6, g16, g24)绘制收敛图和盒型图, 以进一步分析实验结果。图 1 和图 2 分别是算法在四个函数中的测试后的收敛图和盒型图。

图 1 中可以直观地展示出 OBLDE-RSO, RSO, LSO, DE 和 GA 的收敛性能。OBLDE-RSO 的收敛趋势线比其他比较算法的收敛速度更快, 同时, OBLDE-RSO 的收敛精度也高于其他比较算法, 这进一步证实了使用两种策略可以提高算法的收敛速度和收敛精度。

在图 2 中, 盒型图展示出 4 次实验得到的实验结果。由图可以看出, 在绝大多数情况下, OBLDE-RSO 的结果优于其他比较算法, 说明 OBLDE-RSO 具有更加优越和稳定的数据结构。

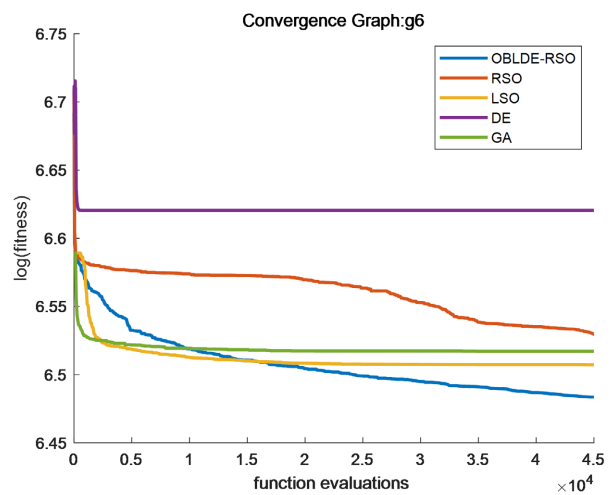
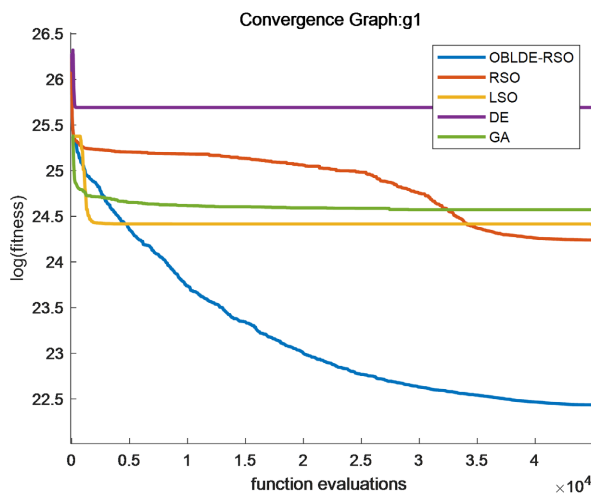
Table 4. Comparison results of OBLDE-RSO, RSO, DE, LSO and GA

表 4. OBLDE-RSO, RSO, DE, LSO 和 GA 的对比结果

	OBLDE-RSO		RSO		LSO		DE		GA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
g1	5.546E+09	2.377E+09	3.366E+10	3.741E+09	3.637E+10	4.945E+09	1.050E+11	1.738E+10	4.697E+10	8.859E+09
g3	4.890E+04	6.714E+03	6.490E+04	8.499E+03	8.781E+04	1.336E+04	3.382E+05	8.377E+04	1.532E+05	1.883E+04
g4	9.056E+02	1.689E+02	1.029E+04	2.105E+03	7.931E+03	2.140E+03	3.524E+04	6.949E+03	8.876E+03	2.159E+03
g5	7.438E+02	3.861E+01	8.821E+02	2.164E+01	8.185E+02	1.736E+01	1.125E+03	4.377E+01	9.034E+02	2.574E+01
g6	6.543E+02	1.156E+01	6.852E+02	5.942E+00	6.702E+02	4.338E+00	7.271E+02	8.859E+00	6.767E+02	7.148E+00
g7	1.093E+03	8.892E+01	1.275E+03	5.222E+01	1.213E+03	5.592E+01	3.176E+03	2.350E+02	2.022E+03	1.542E+02

Continued

g8	1.005E+03	2.934E+01	1.069E+03	3.158E+01	1.065E+03	1.626E+01	1.372E+03	3.513E+01	1.158E+03	2.815E+01
g9	5.956E+03	1.335E+03	9.020E+03	1.666E+03	6.111E+03	7.949E+02	3.278E+04	4.676E+03	1.328E+04	1.658E+03
g10	6.598E+03	5.480E+02	8.139E+03	7.180E+02	8.231E+03	5.671E+02	9.161E+03	3.946E+02	8.616E+03	2.354E+02
g11	1.858E+03	4.538E+02	4.832E+03	1.107E+03	6.748E+03	2.217E+03	2.987E+04	9.911E+03	1.104E+04	2.925E+03
g12	2.989E+08	2.418E+08	1.150E+10	1.826E+09	4.785E+09	1.672E+09	2.057E+10	4.167E+09	5.510E+09	1.069E+09
g13	2.764E+07	4.822E+07	8.703E+09	2.695E+09	2.129E+09	9.345E+08	1.666E+10	6.124E+09	2.609E+09	1.068E+09
g14	4.522E+05	7.855E+05	1.794E+06	1.289E+06	9.252E+05	5.676E+05	1.845E+07	1.140E+07	2.168E+06	1.259E+06
g15	7.855E+05	3.335E+06	5.400E+07	1.960E+08	4.068E+07	3.732E+07	3.562E+09	1.460E+09	3.377E+08	1.804E+08
g16	3.283E+03	3.711E+02	4.084E+03	3.353E+02	4.196E+03	5.663E+02	6.798E+03	5.379E+02	4.224E+03	2.930E+02
g17	2.469E+03	2.237E+02	2.953E+03	2.447E+02	2.772E+03	2.832E+02	6.547E+03	2.632E+03	3.001E+03	2.063E+02
g18	2.262E+06	2.139E+06	5.917E+06	7.186E+06	7.958E+06	8.369E+06	1.758E+08	8.744E+07	1.758E+07	1.081E+07
g19	5.870E+06	1.096E+07	3.922E+08	2.143E+08	1.169E+08	1.361E+08	4.769E+09	2.094E+09	5.623E+08	2.223E+08
g20	2.608E+03	1.216E+02	2.805E+03	1.465E+02	2.658E+03	1.300E+02	3.443E+03	1.994E+02	2.981E+03	1.478E+02
g21	2.513E+03	7.882E+01	2.689E+03	3.460E+01	2.626E+03	3.195E+01	2.877E+03	4.086E+01	2.662E+03	2.859E+01
g22	5.250E+03	2.377E+03	8.488E+03	1.317E+03	7.907E+03	9.811E+02	1.082E+04	3.508E+02	8.568E+03	1.042E+03
g23	2.876E+03	3.795E+01	3.206E+03	6.311E+01	3.212E+03	6.475E+01	3.526E+03	5.380E+01	3.189E+03	4.791E+01
g24	3.062E+03	6.969E+01	3.436E+03	7.115E+01	3.361E+03	1.045E+02	3.578E+03	8.694E+01	3.363E+03	6.121E+01
g25	3.164E+03	9.531E+01	4.542E+03	2.238E+02	4.426E+03	4.430E+02	1.513E+04	2.192E+03	6.195E+03	6.054E+02
g26	6.087E+03	6.267E+02	8.853E+03	5.860E+02	9.294E+03	4.240E+02	1.354E+04	1.056E+03	9.339E+03	5.185E+02
g27	3.336E+03	5.778E+01	3.887E+03	2.361E+02	3.200E+03	8.284E-05	3.959E+03	2.058E+02	3.710E+03	1.106E+02
g28	3.793E+03	6.097E+02	5.892E+03	4.008E+02	3.300E+03	3.037E-03	1.090E+04	1.302E+03	6.088E+03	5.519E+02
g29	4.717E+03	4.120E+02	5.119E+03	4.572E+02	5.791E+03	3.198E+02	8.318E+03	2.699E+03	5.651E+03	2.997E+02
g30	2.463E+07	1.648E+07	6.115E+08	3.702E+08	2.707E+08	1.541E+08	2.431E+09	1.063E+09	3.529E+08	1.678E+08



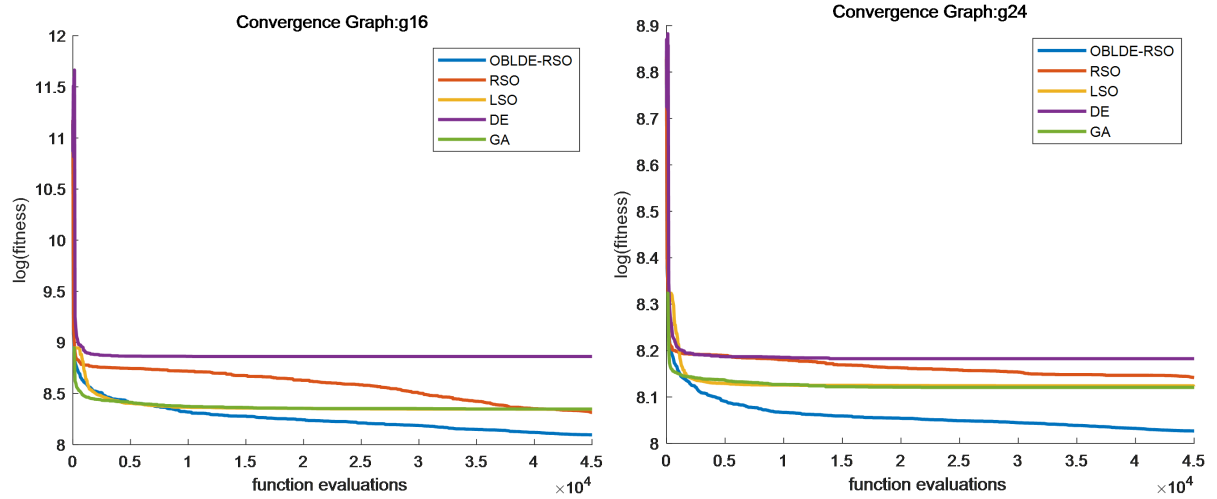


Figure 1. Convergence curves of each test function
图 1. 各算法的收敛曲线对比图

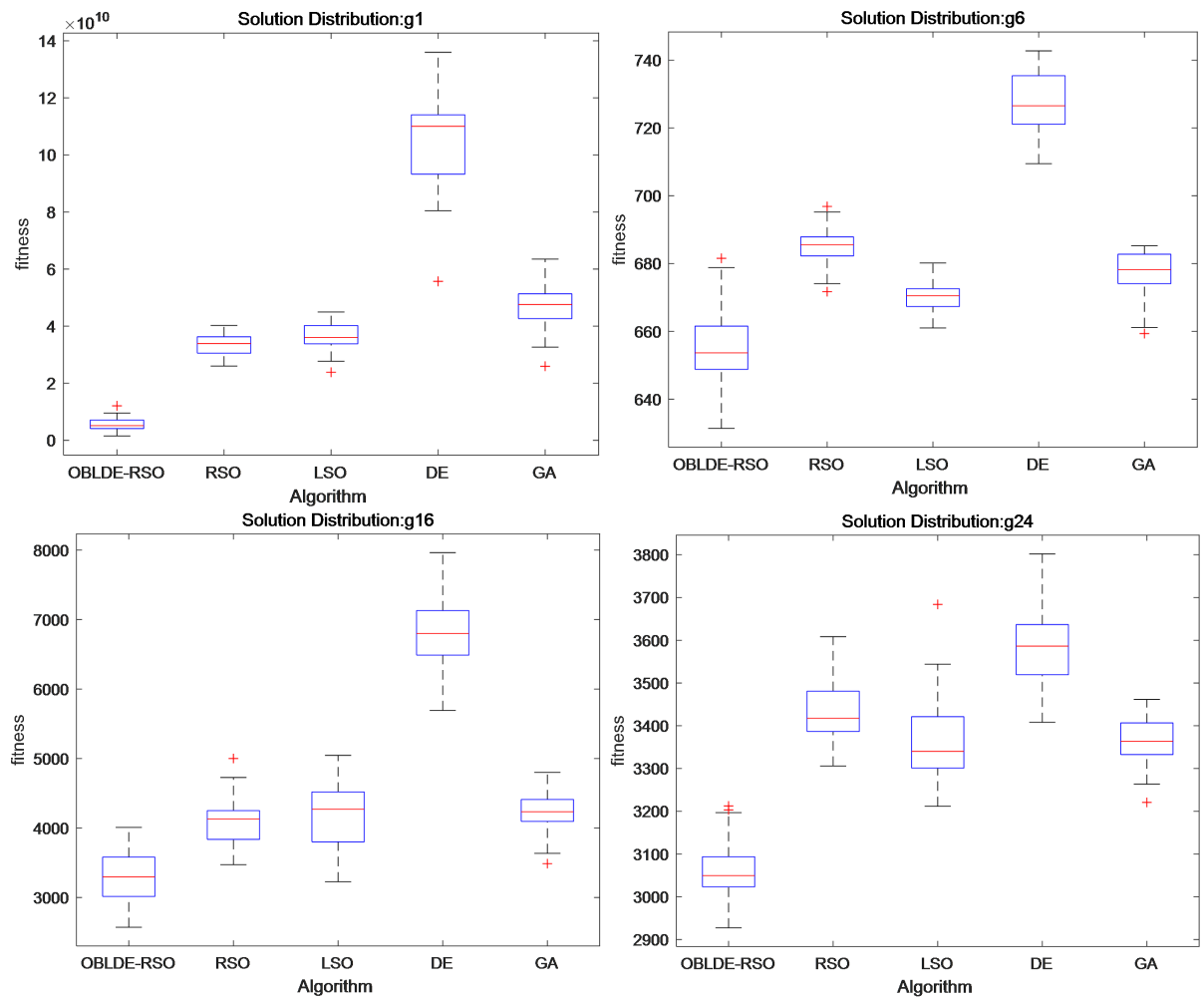


Figure 2. Box chart comparison results of each test function
图 2. 算法的盒图对比结果

5. 结论

本文提出了一种改进的鼠群优化算法——基于反向学习的 DE-RSO 混合优化算法, 算法使用了 DE-RSO 混合策略和反向学习策略以提高算法的性能。DE-RSO 混合策略使用变异机制以保持种群多样性并降低算法陷入局部最优的可能性, 可以有效改善 RSO 过早收敛的缺点。OBL 策略可以针对性地扩大个体的搜索范围, 使个体以更高概率找到潜在的更加理想的求解区域。同时添加一个参数 B 以控制算法在搜索前期和后期的不同搜索任务。通过对比 OBLDE-RSO 和其它比较算法在 IEEE CEC2017 测试函数集上的实验数据和图像, 证明了使用了两种策略后, 算法的局部搜索能力和全局搜索能力都得到了很大的提升。

基金项目

国家自然科学基金面上项目——基于多智能算法融合模型的多级并行叠前分频波形反演研究, 41772123。

参考文献

- [1] Jalili, S., Nallaperuma, S., Keedwell, E., *et al.* (2021) Application of Metaheuristics for Signal Optimisation in Transportation Networks: A Comprehensive Survey. *Swarm and Evolutionary Computation*, **63**, Article ID: 100865. <https://doi.org/10.1016/j.swevo.2021.100865>
- [2] Holland, J.H. (1992) Genetic Algorithms. *Scientific American*, **267**, 66-73. <https://doi.org/10.1038/scientificamerican0792-66>
- [3] Price, K.V., Storn, R.M. and Lampinen, J.A. (2006) Differential Evolution: A Practical Approach to Global Optimization. Springer Science & Business Media, Berlin.
- [4] Papadrakakis, M., Lagaros, N.D. and Tsompanakis, Y. (1998) Structural Optimization Using Evolution Strategies and Neural Networks. *Computer Methods in Applied Mechanics and Engineering*, **156**, 309-333. [https://doi.org/10.1016/S0045-7825\(97\)00215-6](https://doi.org/10.1016/S0045-7825(97)00215-6)
- [5] Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S. (2009) GSA: A Gravitational Search Algorithm. *Information Sciences*, **179**, 2232-2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- [6] Alatas, B. (2011) ACROA: Artificial Chemical Reaction Optimization Algorithm for Global Optimization. *Expert Systems with Applications*, **38**, 13170-13180. <https://doi.org/10.1016/j.eswa.2011.04.126>
- [7] Kaveh, A. (2017) Ray Optimization Algorithm. In: *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer, Cham, 237-280. https://doi.org/10.1007/978-3-319-46173-1_8
- [8] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, 27 November-1 December 1995, 1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>
- [9] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [10] Liu, S., Yang, Y. and Zhou, Y. (2018) A Swarm Intelligence Algorithm-Lion Swarm Optimization. *Pattern Recognition and Artificial Intelligence*, **31**, 431-441.
- [11] Abdulhameed, S. and Rashid, T.A. (2021) Child Drawing Development Optimization Algorithm Based on Child's Cognitive Development. *Arabian Journal for Science and Engineering*. <https://doi.org/10.1007/s13369-021-05928-6>
- [12] Shi, Y. (2011) Brain Storm Optimization Algorithm. *International Conference in Swarm Intelligence*, Chongqing, 12-15 June 2011, 303-309. https://doi.org/10.1007/978-3-642-21515-5_36
- [13] Dhiman, G., Garg, M., Nagar, A., *et al.* (2021) A Novel Algorithm for Global Optimization: Rat Swarm Optimizer. *Journal of Ambient Intelligence and Humanized Computing*, **12**, 8457-8482. <https://doi.org/10.1007/s12652-020-02580-0>
- [14] Vasantharaj, A., Rani, P.S., Huque, S., *et al.* (2021) Automated Brain Imaging Diagnosis and Classification Model Using Rat Swarm Optimization with Deep Learning Based Capsule Network. *International Journal of Image and Graphics*, Article ID: 2240001. <https://doi.org/10.1142/S0219467822400010>
- [15] Kumar, B.V. (2021) Hybrid Metaheuristic Optimization based Feature Subset Selection with Classification Model for Intrusion Detection in Big Data Environment. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, **12**, 2297-2308.

- [16] Tizhoosh, H.R. (2005) Opposition-Based Learning: A New Scheme for Machine Intelligence. *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Vienna, 28-30 November 2005, 695-701. <https://doi.org/10.1109/CIMCA.2005.1631345>
- [17] Yang, Z., Deng, L.B., Wang, Y.C. and Liu, J.F. (2021) Aptenodytes Forsteri Optimization: Algorithm and Applications. *Knowledge-Based Systems*, **232**, Article ID: 107483. <https://doi.org/10.1016/j.knosys.2021.107483b>