

控制与决策

Control and Decision

海狸算法：一种自然启发的元启发式算法

廖想, 周安琪, 刘珂, 郑梓煜, 付波, 李超顺

引用本文:

廖想, 周安琪, 刘珂, 等. 海狸算法: 一种自然启发的元启发式算法[J]. *控制与决策*, 2025, 40(3): 1043-1049.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2024.0362>

您可能感兴趣的其他文章

Articles you may be interested in

面向全局搜索的自适应领导者樽海鞘群算法

Global search-oriented adaptive leader salp swarm algorithm

控制与决策. 2021, 36(9): 2152-2160 <https://doi.org/10.13195/j.kzyjc.2020.0090>

基于弱关联的自适应高维多目标进化算法

A weak association-based adaptive evolutionary algorithm for manyobjective optimization

控制与决策. 2021, 36(8): 1804-1814 <https://doi.org/10.13195/j.kzyjc.2019.1723>

求解非线性方程组的智能优化算法综述

Overview of intelligent optimization algorithms for solving nonlinear equation systems

控制与决策. 2021, 36(4): 769-778 <https://doi.org/10.13195/j.kzyjc.2020.0379>

基于分解的多目标多因子进化算法

A multiobjective multifactorial evolutionary algorithm based on decomposition

控制与决策. 2021, 36(3): 637-644 <https://doi.org/10.13195/j.kzyjc.2019.0525>

融合能量周期性递减与牛顿局部增强的改进HHO算法

[Improved harris hawks optimization coupling energy cycle decline mechanism and Newton local enhancement strategy](#)

控制与决策. 2021, 36(3): 629-636 <https://doi.org/10.13195/j.kzyjc.2019.0810>

海狸算法: 一种自然启发的元启发式算法

廖想^{1†}, 周安琪¹, 刘珂¹, 郑梓煜¹, 付波¹, 李超顺²

(1. 湖北工业大学 电气与电子工程学院, 武汉 430068; 2. 华中科技大学 土木与水利工程学院, 武汉 430074)

摘要: 为了克服传统数值方法在处理复杂优化问题时的局限性, 提高找到全局最优解的效率, 提出一种名为海狸算法 (BA) 的新型元启发式算法, 用于解决全局优化问题. 首先, BA 根据海狸在修建海狸坝时的伐木行为, 将海狸分为质检狸、开发狸和采伐狸 3 种类型, 以模拟其在伐木过程中的群体合作. 其中: 开发狸以随机方式搜索未知木材地, 寻找新的木材资源; 采伐狸则奔袭至质检狸处搜集木材, 并在奔袭过程中寻找木材资源; 而质检狸引领海狸群体朝着木材资源最丰富的方向前进. 然后, 将 BA 在 CEC 2017 测试函数上进行测试, 并与其他 7 种算法进行比较, 研究结果显示, BA 在大部分函数中获得了最佳解, 具有较强的优化能力. 最后, 将 BA 应用于拉伸/压缩弹簧设计、三杆桁架设计等两个具有挑战性的工程问题, 并与其他两种算法进行比较, 结果表明, BA 在这些工程问题中均取得了最佳的优化结果, 相较于其他两种算法表现更为出色.

关键词: 元启发式算法; 海狸算法; 群智能优化; 全局优化; 单目标; 优化算法

中图分类号: TP18

文献标志码: A

DOI: 10.13195/j.kzyjc.2024.0362

引用格式: 廖想, 周安琪, 刘珂, 等. 海狸算法: 一种自然启发的元启发式算法 [J]. 控制与决策, 2025, 40(3): 1043-1049.

Beaver algorithm: A nature-inspired metaheuristic algorithm

LIAO Xiang^{1†}, ZHOU An-qi¹, LIU Ke¹, ZHENG Zi-yu¹, FU Bo¹, LI Chao-shun²

(1. School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China;
2. School of Civil and Hydraulic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: In order to overcome the limitations of traditional numerical methods in dealing with complex optimization problems and improve the efficiency of finding global optimal solutions, this paper proposes a new metaheuristic algorithm called Beaver algorithm (BA) to solve global optimization problems. According to the felling behaviour of Beavers during the construction of Beaver dams, the BA divides Beavers into three types: Inspection Beaver, development Beaver, and cutting Beaver, to simulate their group cooperation in the felling process. The development Beaver searches unknown timber land randomly to find new timber resources. The cutting Beaver rushes to the inspection Beaver to collect wood and is in the process of raiding to find wood resources. The inspection Beaver lead the whole Beaver towards the direction of the most abundant wood resources. The BA is tested on the CEC 2017 test function and compared with seven other algorithms. The results show that the BA obtains the best solution in most functions and has strong optimization ability. In addition, the BA is applied to two challenging engineering problems, such as tension/compression spring design and three-bar truss design, and is compared with the other two algorithms. The results show that the BA achieves the best optimization in engineering problems and performs better than the other two algorithms.

Keywords: metaheuristic algorithms; Beaver algorithm; swarm intelligence optimization; global optimization; single objective; optimization algorithm

0 引言

随着科学技术的不断发展, 优化问题求解的复杂性和难度显著增加, 元启发式算法被提出以解决这些复杂的现实问题. 在问题求解过程中, 元启发式

算法无需计算问题的梯度, 只需要在一些可行解下计算问题的值, 极大地减少了计算复杂度. 在处理不同问题时, 只需要修改待解函数, 而无需对算法的核心部分进行大量修改. 这种简单性和灵活性使得元

收稿日期: 2024-04-03; 录用日期: 2024-07-24.

基金项目: 国家自然科学基金项目 (51809097); 太阳能高效利用及储能运行控制湖北省重点实验室开放基金项目 (HBSEES202312); 新能源及电网装备安全监测湖北省工程研究中心开放基金项目 (HBSKF202125).

†通信作者. E-mail: liaoxiang@hbut.edu.cn.

启发式算法在许多领域得到了广泛应用.

元启发式算法通常分为3类:基于进化的算法、基于物理的算法和基于群体的算法.基于进化的算法是一类模拟自然进化过程的优化算法,这类算法包括遗传算法^[1]、差分进化算法^[2]、进化策略^[3]、遗传规划^[4]、进化规划^[5]等;基于物理的算法是一类模拟物理过程或受物理原理启发的优化算法,这类算法包括引力搜索算法^[6]、模拟退火算法^[7]、黑洞算法^[8]、多宇宙优化器^[9]、带电系统搜索^[10]等.基于群体的算法是一类模拟群体行为和群体智能的优化算法,这类算法包括蚁群算法^[11]、人工蜂群算法^[12]、粒子群算法^[13]、鲸鱼优化算法^[14]、沙猫群优化算法^[15]等.

尽管不断有新的算法被开发出来,但是 no free lunch 理论指出,没有特定的算法可以完美地解决所有优化问题.因此,有必要开发不同的元启发式算法来填补解决特定问题的空白.鉴于此,本文提出一种新的基于群体的元启发式算法,称为海狸算法(BA).通过对 CEC 2017 测试函数的测试以及对两个工程优化问题的研究来评估 BA 的性能,并将其结果与其他元启发式算法进行比较.实验结果表明,BA 的优化性能优于对比算法.

1 海狸优化

海狸是一种半水栖的珍贵哺乳动物,也是体型最大的啮齿动物之一.它们体型肥胖溜圆,毛色主要为棕褐色,夹杂着黑、灰、黄、白等色彩,具有两层毛,外层粗长,内层柔软光亮.海狸常利用前爪涂抹尾部分泌的油脂来防水.它们广泛分布在北半球欧亚大陆森林地带的河流、湖泊和沼泽等水域.

海狸以其伐木行为和建筑能力而闻名.在河流系统中,海狸通常会挖掘河岸洞穴来建造住所.为了维持生活区内水位的稳定,海狸常常用树枝、石块和软泥筑堤坝,以阻挡溪流形成池塘或湖泊,其筑造的堤坝长达数十米.海狸通常挖掘浅沟,也称为运河,从海狸池塘横向延伸.这些结构使得海狸能够更容易获得食物和建筑资源.运河通常发展为密集的网络,对洪泛平原的当地水文地貌做出了重大贡献,在其他平面景观中形成了曲折的流动路径和复杂的地形^[16].

海狸通常以群体形式协同工作,彼此相互合作,传递木材或建材,以实现共同目标.这种群体协作行为可被模拟并转化为一些任务分配和调度算法,以优化群体协作的效果.根据海狸在修建海狸坝时的伐木行为,BA 将海狸分类为质检狸、开发狸和采伐狸3种类型,模拟它们在伐木过程中的群体合作来解决优化问题.

2 数学模型和优化算法

在 BA 中:海狸所获取的木材为问题的解决方案,而木材的数量则对应解决方案的质量.开发狸通过随机搜索和探索未知木材地寻找新资源,增加算法的全局搜索能力,避免陷入局部最优解;采伐狸则在搜集木材的过程中变异当前解,引入局部或全局的随机扰动,创造新的候选解,促进更全面地探索解空间;质检狸则领导海狸群体前往木材资源最丰富的方向,指导搜索朝着全局最优解的方向前进.

2.1 初始化

在多维优化问题中,每只海狸均为一个列矩阵.每个列矩阵为一个问题的解决方案.在一组变量 $(x_{i,1}, x_{i,2}, \dots, x_{i,\text{dim}})$ 中,每个变量必须位于上边界与下边界间.BA 的初始化如下所示:

$$x = [x_1, x_2, \dots, x_N] = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,\text{dim}} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \dots & x_{N,\text{dim}} \end{bmatrix}. \quad (1)$$

其中: x 为初始种群位置; N 为种群数量; dim 为种群维数; $x_{i,j}$ 为个体 i 在 j 维中的位置, $x_{i,j}$ 的值由下式得到:

$$x_{i,j} = x_j^{\min} + \text{rand}(0, 1)(x_j^{\max} - x_j^{\min}). \quad (2)$$

这里: x_j^{\min} 为第 j 维的下界, x_j^{\max} 为第 j 维的上界.

2.2 开发狸

开发狸数量为 $m = \frac{2}{3}N$, 开发狸在整个木材地内随机游走,最大游走次数为 maxWalks , 游走过程的数学公式如下所示:

$$a_1 = \max(x_j^{\min}, x_j - \text{stepsize}), \quad (3)$$

$$a_2 = \min(x_j^{\max}, x_j + \text{stepsize}), \quad (4)$$

$$v_{i,j} = a_1 + \varphi(a_2 - a_1). \quad (5)$$

其中: stepsize 为游走步长, φ 为区间 $[0, 1]$ 内的随机数.在游走过程中,若开发狸所在处木材数量大于原位置木材数量,则开发狸的位置更新为新位置;若开发狸所在处木材数量大于质检狸所在处木材数量,则其更新为质检狸.开发狸重复以上游走行为直至达到最大游走次数 maxWalks .

2.3 采伐狸

采伐狸数量为 $c = \frac{1}{6}N$, 每只采伐狸会根据锦标赛选择一只质检狸奔袭.在 BA 中,每次锦标赛由两只质检狸竞争.在每次循环中,随机选择质检狸并比较它们的适应度,找到适应度最好的质检狸(胜者).若当前锦标赛胜者的适应度高于其他质检狸或锦标赛大小已等于质检狸总数,则返回胜者的索引.采伐

狸最大奔袭次数为maxRaidSteps,奔袭过程的数学公式为

$$b_1 = \max \left(x_j^{\min}, x_j - \sin \frac{\text{fit}_u}{\text{fit}_i} \times \frac{x_j^{\max} - x_j^{\min}}{10} \right), \quad (6)$$

$$b_2 = \min \left(x_j^{\max}, x_j + \sin \frac{\text{fit}_u}{\text{fit}_i} \times \frac{x_j^{\max} - x_j^{\min}}{10} \right), \quad (7)$$

$$w_{i,j} = b_1 + \varphi(b_2 - b_1). \quad (8)$$

其中: fit_i 为采伐狸当前位置的适应度值, fit_u 为采伐狸选择奔袭的质检狸的适应度值. 在奔袭过程中, 若采伐狸所在处的木材数量大于其所奔袭质检狸所在处的木材数量, 则将采伐狸的位置更新为质检狸的位置. 采伐狸重复以上奔袭行为直至达到最大奔袭次数maxRaidSteps, 此后采伐狸会在该地进行局部搜索, 其局部搜索方式与质检狸一样, 将在后文进行详细介绍.

2.4 质检狸

质检狸数量为 $n = \frac{1}{6}N$, 质检狸所在处是木材数量最多的地方, 它们在所在处进行局部搜索, 搜索过程的数学公式为

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}, \quad (9)$$

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id, \text{pbest}}^k - x_{id}^k) + c_2 r_2 (p_{d, \text{gbest}}^k - x_{id}^k). \quad (10)$$

其中: r_1 和 r_2 为区间 $[0, 1]$ 内的随机数, v_{id}^k 为粒子 i 在第 k 次迭代中第 d 维的速度向量, x_{id}^k 为粒子 i 在第 k 次迭代中第 d 维的位置向量, $p_{id, \text{pbest}}^k$ 为粒子 i 在第 k 次迭代中第 d 维的历史最优位置, $p_{d, \text{gbest}}^k$ 为群体在第 k 次迭代中第 d 维的历史最优位置, ω 为惯性因子, c_1 为个体学习因子, c_2 为群体学习因子. 在局部搜索的过程中, 若搜索处木材数量大于质检狸原本所在处木材数量, 则质检狸的位置更新为搜索处.

2.5 算法流程

BA 的具体步骤如下.

step 1: 初始化参数 n 、 m 、 c 、maxWalks、stepsize、maxRaidSteps.

step 2: 由式 (2), 建立初始种群.

step 3: 按照适应度值对海狸进行升序排列. 其中: 适应度值排名前 n 的海狸为质检狸, 适应度值排名于质检狸后的 m 只海狸为开发狸, 剩下的 c 只海狸为采伐狸.

step 4: 开发狸随机游走, 数学公式如式 (3) ~ (5) 所示. 开发狸游走过程中, 若适应度值优于质检狸, 则其更新为质检狸; 否则, 开发狸游走直至达到最大游走次数maxWalks.

step 5: 采伐狸朝质检狸奔袭, 数学公式如式 (6) ~

(8) 所示. 采伐狸奔袭过程中, 若适应度值优于所奔袭的质检狸, 则更新为质检狸; 否则, 采伐狸奔袭直至达到最大奔袭次数maxRaidSteps. 达到最大奔袭次数后, 采伐狸会在该地进行局部搜索.

step 6: 质检狸在所在处进行局部搜索, 数学公式如式 (9) 和 (10) 所示. 在局部搜索过程中, 若搜索处适应度值优于原位置, 则质检狸的位置更新为搜索处.

step 7: 记录全局最优值, 若满足终止条件, 则输出最优解和最优值; 否则, 返回至 step 3.

3 结果与讨论

3.1 基准测试函数

CEC 2017 基准测试函数是一组极具挑战性的测试函数. 该集合包含 2 个单峰函数 (CF1 和 CF3)、10 个多峰函数 (CF4 ~ CF10)、10 个混合函数 (CF11 ~ CF20) 和 10 个复合函数 (CF21 ~ CF30). 由于 CF2 函数的较大不稳定性, 其在大多数研究中已不再用于算法测试.

3.2 实验装置

本文基于 Intel (R) Core (TM) i7-10875 H CPU、2.30 GHz 主频、16 GB 内存以及 Windows 11 (64 位) 的操作系统对所提出算法进行仿真实验. 算法的编程软件为 Matlab R 2020 b. 在 CEC 2017 基准测试函数的实验中, 基准测试函数的上下限被设定为 100 和 -100, 维度为 10. 为了验证 BA 的性能, 本文选择鱼鹰优化算法^[17] (OOA)、沙猫群优化算法 (SCSO)、斑马优化算法^[18] (ZOA)、鹈鹕优化算法^[19] (POA)、非洲秃鹫优化算法^[20] (AVOA)、飞蛾扑火优化算法^[21] (MFO)、灰狼优化算法^[22] (GWO) 与 BA 进行比较. 这些算法的种群大小为 30, 迭代次数为 500. 参数设置如表 1 所示.

表1 参数设置

算法	主要参数	取值
OOA	l	{1, 2}
	r	[0, 1]
ZOA	P_S	0.5
	R	0.01
AVOA	L_1	[0, 1]
	L_2	[0, 1]
	z	[0, 1]
	h	[-2, 2]
BA	maxWalks	2
	stepsize	1.6
	maxRaidSteps	2
SCSO	S_M	2
MFO	t	[-1, 1]
	r	[-2, -1]
GWO	a	[0, 2]
	r_1	[0, 1]
	r_2	[0, 1]
POA	l	{1, 1}
	R	0.2

3.3 CEC 2017 测试函数的统计结果分析

表2为BA与7种比较算法在独立运行30次后所得到的数据统计结果。在CF1、CF3~CF5中:BA取得了最好的平均值和标准差,取得的值远好于其他7种算法;在CF6中:GWO取得了最好的平均值和标准差,虽然BA没有取得最好的平均值和标准差,但是其结果优于OOA、SCSO、ZOA、POA、AVOA;

在CF7中:BA取得了最好的平均值和标准差,表明BA取得的值好且较稳定;在CF8中:BA取得了最好的平均值,ZOA取得了最好的标准差,表明BA取得的值好但是其稳定性有待提高;在CF9~CF23中:BA取得了最好的平均值和标准差,优化性能优于其他7种算法;在CF24、CF25和CF27中:BA取得了最好的平均值,GWO取得了最好的标准差;在

表2 均值标准差

F	metric	BA	OOA	SCSO	ZOA	POA	AVOA	GWO	MFO
CF1	mean	5.255 6e+02	1.063 7e+10	9.368 6e+07	8.267 6e+08	1.888 5e+08	3.151 7e+03	1.510 7e+07	1.230 6e+08
	std	5.374 9e+02	3.129 6e+09	1.746 3e+08	1.077 7e+09	3.262 9e+08	3.119 1e+03	6.292 6e+07	4.309 8e+08
CF3	mean	3.000 0e+02	1.321 4e+04	2.601 5e+03	1.465 4e+03	8.666 2e+02	4.493 1e+02	2.796 1e+03	1.275 3e+04
	std	1.511 3e-13	3.971 2e+03	2.019 6e+03	1.399 0e+03	8.347 6e+02	1.497 1e+02	2.571 0e+03	1.329 8e+04
CF4	mean	4.017 5e+02	1.163 0e+03	4.422 3e+02	4.589 7e+02	4.339 0e+02	4.103 0e+02	4.182 1e+02	4.275 0e+02
	std	1.408 9e+00	3.980 7e+02	4.527 1e+01	5.387 0e+01	4.080 6e+01	1.642 3e+01	2.118 7e+01	3.556 2e+01
CF5	mean	5.145 6e+02	5.795 8e+02	5.352 3e+02	5.384 2e+02	5.400 2e+02	5.481 2e+02	5.224 1e+02	5.319 2e+02
	std	2.783 5e+00	1.946 4e+01	1.450 3e+01	1.335 6e+01	1.423 6e+01	1.471 5e+01	1.344 1e+01	1.282 8e+01
CF6	mean	6.069 0e+02	6.435 2e+02	6.218 2e+02	6.194 1e+02	6.228 6e+02	6.219 5e+02	6.021 1e+02	6.037 1e+02
	std	3.062 9e+00	1.181 5e+01	1.204 0e+01	6.592 5e+00	1.241 1e+01	1.206 3e+01	2.201 1e+00	3.796 1e+00
CF7	mean	7.307 3e+02	8.030 3e+02	7.618 7e+02	7.470 8e+02	7.677 0e+02	7.734 3e+02	7.351 7e+02	7.491 1e+02
	std	8.605 9e+00	2.329 5e+01	1.911 3e+01	1.190 0e+01	1.718 5e+01	2.238 6e+01	1.075 4e+01	2.086 4e+01
CF8	mean	8.131 3e+02	8.521 7e+02	8.286 7e+02	8.179 4e+02	8.228 1e+02	8.289 8e+02	8.158 6e+02	8.292 3e+02
	std	4.159 3e+00	1.040 0e+01	7.864 4e+00	3.652 1e+00	6.466 1e+00	9.455 9e+00	6.353 2e+00	1.120 5e+01
CF9	mean	9.231 1e+02	1.400 6e+03	1.764 6e+03	1.086 4e+03	1.105 9e+03	1.169 4e+03	9.254 4e+02	1.086 0e+03
	std	2.058 6e+01	2.199 2e+02	2.691 6e+02	1.087 9e+02	1.405 3e+02	2.137 5e+02	4.268 5e+01	3.319 6e+02
CF10	mean	1.231 7e+03	2.482 1e+03	2.009 6e+03	1.752 0e+03	1.713 2e+03	1.967 5e+03	1.729 0e+03	1.947 7e+03
	std	1.328 1e+02	2.439 5e+02	3.527 1e+02	2.189 1e+02	1.784 2e+02	2.912 3e+02	3.934 0e+02	2.882 7e+02
CF11	mean	1.111 9e+03	3.368 7e+03	1.169 6e+03	1.181 9e+03	1.153 7e+03	1.147 2e+03	1.143 7e+03	1.280 9e+03
	std	5.719 2e+00	3.102 3e+03	5.505 0e+01	7.489 1e+01	3.747 8e+01	2.973 5e+01	3.720 1e+01	3.623 3e+02
CF12	mean	6.114 5e+03	2.934 4e+08	1.114 0e+06	1.681 4e+06	4.461 0e+05	2.083 1e+06	5.486 6e+05	2.394 3e+06
	std	3.496 8e+03	2.568 8e+08	1.460 5e+06	1.440 1e+06	9.308 9e+05	2.777 1e+06	6.687 9e+05	4.423 6e+06
CF13	mean	1.647 5e+03	1.238 7e+06	1.475 2e+04	1.233 6e+04	2.329 1e+03	1.179 2e+04	1.506 1e+04	1.298 7e+04
	std	1.403 3e+02	3.425 0e+06	1.015 7e+04	6.466 3e+03	8.623 3e+02	9.228 8e+03	1.065 3e+04	1.249 3e+04
CF14	mean	1.449 3e+03	2.312 6e+03	3.437 1e+03	4.006 5e+03	1.472 7e+03	2.329 8e+03	3.531 0e+03	4.488 3e+03
	std	1.163 9e+01	1.144 8e+03	2.107 8e+03	3.162 7e+03	3.852 7e+01	1.231 8e+03	2.076 5e+03	5.229 2e+03
CF15	mean	1.563 0e+03	1.250 0e+04	4.381 3e+03	5.157 1e+03	1.645 1e+03	6.076 6e+03	7.740 0e+03	9.322 5e+03
	std	2.387 2e+01	6.517 1e+03	1.993 5e+03	3.434 5e+03	8.146 9e+01	4.921 5e+03	5.900 4e+03	9.803 3e+03
CF16	mean	1.607 6e+03	1.992 3e+03	1.872 0e+03	1.832 9e+03	1.800 7e+03	1.910 6e+03	1.768 6e+03	1.755 4e+03
	std	6.236 5e+00	1.342 3e+02	1.185 0e+02	1.035 0e+02	1.270 6e+02	1.601 6e+02	1.382 8e+02	1.227 1e+02
CF17	mean	1.738 9e+03	1.812 8e+03	1.777 9e+03	1.759 0e+03	1.758 8e+03	1.784 0e+03	1.770 4e+03	1.790 6e+03
	std	7.999 3e+00	4.131 0e+01	2.755 5e+01	1.184 3e+01	1.509 6e+01	4.280 0e+01	4.387 3e+01	6.555 1e+01
CF18	mean	1.960 8e+03	6.270 9e+06	2.251 7e+04	8.600 4e+03	3.894 5e+03	1.812 5e+04	2.641 2e+04	2.624 7e+04
	std	8.754 1e+01	1.891 6e+07	1.201 5e+04	5.811 1e+03	5.066 0e+03	1.185 9e+04	1.510 8e+04	1.528 6e+04
CF19	mean	1.929 5e+03	4.597 2e+04	1.787 7e+04	9.018 8e+03	1.947 4e+03	9.908 0e+03	1.992 7e+04	9.336 0e+03
	std	1.108 9e+01	6.236 5e+04	4.442 1e+04	2.034 3e+04	4.162 0e+01	9.305 5e+03	4.922 0e+04	9.851 2e+03
CF20	mean	2.044 6e+03	2.188 7e+03	2.129 1e+03	2.100 3e+03	2.079 6e+03	2.131 6e+03	2.081 5e+03	2.086 2e+03
	std	1.251 1e+01	7.453 0e+01	6.694 8e+01	5.460 4e+01	3.838 0e+01	7.523 4e+01	4.986 0e+01	6.388 9e+01
CF21	mean	2.187 9e+03	2.343 7e+03	2.295 4e+03	2.312 2e+03	2.263 2e+03	2.303 3e+03	2.317 1e+03	2.309 7e+03
	std	4.665 5e+00	5.281 0e+01	7.039 6e+01	3.425 2e+01	6.492 2e+01	6.619 1e+01	8.938 0e+00	4.789 9e+01
CF22	mean	2.285 4e+03	2.926 7e+03	2.330 7e+03	2.346 4e+03	2.320 8e+03	2.341 3e+03	2.314 6e+03	2.324 1e+03
	std	2.955 8e+01	3.091 9e+02	3.241 2e+01	3.963 5e+01	4.712 1e+01	1.687 1e+02	2.991 5e+01	7.756 2e+01
CF23	mean	2.419 0e+03	2.712 4e+03	2.645 0e+03	2.674 2e+03	2.650 2e+03	2.644 5e+03	2.624 8e+03	2.631 3e+03
	std	9.519 6e+00	2.765 9e+01	1.886 2e+01	2.801 9e+01	1.812 3e+01	2.086 5e+01	1.353 5e+01	1.265 6e+01
CF24	mean	2.594 5e+03	2.919 7e+03	2.766 8e+03	2.755 9e+03	2.656 1e+03	2.765 9e+03	2.753 3e+03	2.758 7e+03
	std	1.300 9e+02	9.757 8e+01	4.239 6e+01	9.602 5e+01	1.353 2e+02	5.374 6e+01	1.573 3e+01	5.035 1e+01
CF25	mean	2.892 8e+03	3.513 5e+03	2.938 5e+03	2.963 7e+03	2.937 0e+03	2.926 7e+03	2.937 7e+03	2.950 4e+03
	std	7.966 0e+01	3.007 9e+02	2.680 0e+01	4.423 0e+01	3.075 3e+01	2.620 0e+01	1.480 8e+01	2.644 7e+01
CF26	mean	2.824 3e+03	3.957 5e+03	3.120 0e+03	3.510 1e+03	3.061 6e+03	3.292 4e+03	2.983 7e+03	3.029 5e+03
	std	6.234 5e+01	4.696 6e+02	1.884 8e+02	3.909 0e+02	2.938 9e+02	4.581 3e+02	1.829 2e+02	2.079 8e+02
CF27	mean	3.063 8e+03	3.297 6e+03	3.111 8e+03	3.187 4e+03	3.108 0e+03	3.115 3e+03	3.096 6e+03	3.098 8e+03
	std	1.258 7e+01	7.137 9e+01	2.724 8e+01	3.900 1e+01	2.613 9e+01	3.154 1e+01	6.643 5e+00	1.570 5e+01
CF28	mean	3.129 5e+03	3.820 1e+03	3.324 7e+03	3.394 9e+03	3.288 6e+03	3.337 6e+03	3.376 2e+03	3.357 0e+03
	std	3.965 0e+01	7.636 6e+01	9.976 0e+01	1.706 5e+02	1.227 9e+02	1.250 8e+02	8.869 5e+01	1.259 4e+02
CF29	mean	3.138 4e+03	3.368 3e+03	3.258 0e+03	3.234 3e+03	3.222 5e+03	3.306 8e+03	3.216 6e+03	3.255 0e+03
	std	7.285 4e+00	8.424 8e+01	7.084 0e+01	3.841 5e+01	5.803 9e+01	8.877 8e+01	5.252 9e+01	5.659 6e+01
CF30	mean	5.642 5e+03	1.873 6e+07	1.263 9e+06	1.263 4e+06	2.117 3e+05	2.751 0e+05	1.189 6e+06	5.695 9e+05
	std	1.827 5e+03	1.371 8e+07	1.587 8e+06	1.657 3e+06	3.979 3e+05	2.908 3e+05	1.273 6e+06	5.277 2e+05

CF26、CF28 ~ CF30 中: BA 取得了最好的平均值和标准差, 表明 BA 取得的值好且稳定性高.

总体而言, BA 具有良好的开发能力和全局探索能力, 以及较强的避免陷入局部最优解的能力, 比其他 7 种算法表现更为优越.

3.4 CEC 2017 测试函数的收敛曲线分析

图 1 为 BA 和 7 种比较算法在 CEC 2017 测试函数上的收敛曲线. 在 CF1、CF3 ~ CF5 中: BA 可快速收敛并获得最好的适应度值, 优于其他对比算法; 在 CF6 中: 虽然 BA 没有取得最好的适应度值, 但是其结果优于 OOA、POA、AVOA、SCSO、ZOA; 在 CF7 和 CF8 中: BA 的收敛速度最快且取得的适

应度值最好, OOA 取得的适应度值最差; 在 CF9 中: BA 取得的适应度值仅次于 MFO, 而 OOA、AVOA 收敛速度慢且过早陷入局部最优; 在 CF10 ~ CF16、CF18、CF21 ~ CF24 中: BA 收敛速度最快, 取得的适应度值最好; 在 CF17、CF20 中: MFO 取得了最好的适应度值, BA 取得的适应度值仅次于 MFO 且收敛速度较快; 在 CF19 中: POA、BA 取得的适应度值最好, BA 的收敛速度大于 POA 的收敛速度; 在 CF25 中: BA 取得的适应度值仅次于 POA; 在 CF26 和 CF27、CF29 和 CF30 中: BA 可快速收敛并获得最好的适应度值, 优于其他对比算法; 在 CF28 中: BA 取得的适应度值仅次于 AVOA.

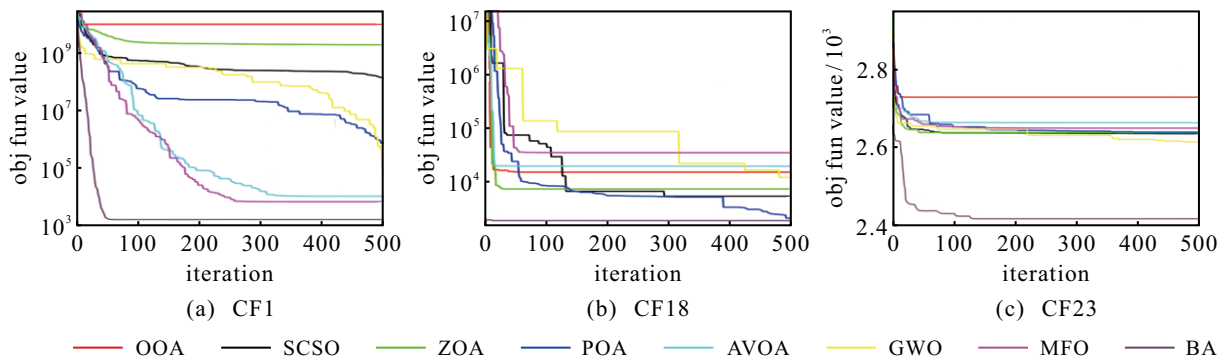


图1 迭代曲线

总体而言, BA 在大部分测试函数上取得了最好的结果. 相较于其他 7 种算法, BA 的性能更加优越.

3.5 CEC 2017 箱型图结果分析

图 2 为 BA 与 7 种比较算法独立运行 30 次所获得的箱型图. 在 CF1、CF3 ~ CF5 中: BA 的箱型图低于其他算法, 且箱型图较窄, 表明 BA 在 CF1、CF3 ~ CF5 中取得的适应度值和稳定性均最好; 在

CF6 中: GWO 取得的适应度值和稳定性最好, BA 取得的适应度值和稳定性优于 OOA、SCSO、ZOA、POA、AVOA; 在 CF7 ~ CF21、CF23 中: BA 取得的适应度值最好且最稳定; 在 CF22、CF24 中: BA 取得的适应度值最好但是框图较宽, 稳定性有待加强; 在 CF25 ~ CF30 中: BA 取得的适应度值最好且较稳定.

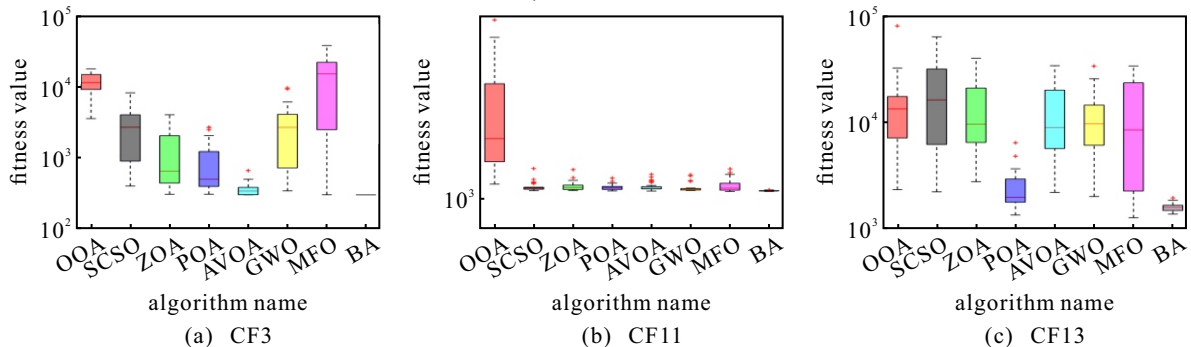


图2 箱型图

总体而言, BA 在大部分测试函数中取得了最好的结果且稳定性较好. 相较于其他 7 种算法, BA 的优化效果更好.

3.6 约束工程设计问题

在本节中, 本文将 BA 应用于拉伸/压缩弹簧设计、三杆桁架设计等工程问题, 并将 BA 得到的结果与鱼鹰优化算法、水流优化器^[23] (water flow optimizer,

WFO) 的结果进行对比分析, 以验证其解决实际工程问题的效果.

3.6.1 拉伸/压缩弹簧设计

在这项研究中, 第 1 个工程优化问题是拉伸/压缩弹簧设计. 该工程问题的优化目标是在一定限制下找到拉伸/压缩弹簧的最小质量. 变量包括线圈直径 d 、平均线圈直径 D 以及有效线圈数 N , 具体如图 3

所示. 约束条件包括最小偏差 (g_1)、剪切应力 (g_2)、冲击频率 (g_3) 和外径极限 (g_4). 问题的数学模型为

$$\begin{aligned} \text{Variablex} &= [d, D, P], \\ \min f(x) &= (x_3 + 2)x_2x_1^2, \\ g_1(x) &= 1 - \frac{x_3x_2^3}{71\,785x_1^4} \leq 0, \\ g_2(x) &= \frac{4x_2^2 - x_1x_2}{12\,566(x_2x_1^3 - x_1^4)} + \frac{1}{5\,108x_1^2} - 1 \leq 0, \\ g_3(x) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\ g_4(x) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0, \\ 0.05 \leq x_1 \leq 2, \quad 0.25 \leq x_2 \leq 1.3, \\ 2 \leq x_3 \leq 15. \end{aligned} \tag{11}$$

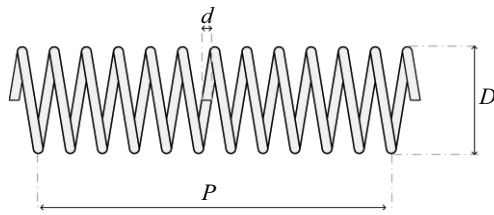


图3 拉伸/压缩弹簧设计

表3 为拉伸/压缩弹簧设计问题最优结果对比. 对比结果表明: BA 取得的结果最好, 明显优于对比算法, 其解决拉伸/压缩弹簧设计问题具有良好的效果.

表3 拉伸/压缩弹簧设计问题最优结果对比

algorithm	d	D	V	best weight
BA	0.052 0	0.359 8	11.569 4	0.013 2
OOA	0.056 2	0.472 1	7.026 8	0.013 5
WFO	0.078 7	1.002 6	2.840 4	0.030 0

3.6.2 三杆桁架设计

在这项研究中, 第2个工程优化问题是三杆桁架设计. 该工程问题的优化目标是通过调整横截面积 (x_1 和 x_2) 来最小化三杆桁架的体积, 具体如图4所示. 问题的数学模型如下所示:

$$\begin{aligned} \text{Variablex} &= [x_1, x_2], \\ \min f(x) &= (2\sqrt{2} \times x_1 + x_2) \times l; \\ g_1(x) &= \frac{\sqrt{2} \times x_1 + x_2}{\sqrt{2} \times x_1^2 + 2x_1x_2} \times P - \sigma \leq 0; \\ g_2(x) &= \frac{x_2}{\sqrt{2} \times x_1^2 + 2x_1x_2} \times P - \sigma \leq 0; \\ g_3(x) &= \frac{1}{\sqrt{2} \times x_2 + x_1} \times P - \sigma \leq 0; \\ l &= 100 \text{ cm}; \quad P = 2 \text{ kN/cm}^2; \quad \sigma = 2 \text{ kN/cm}^2; \\ 0 \leq x_i \leq 1, \quad i &= 1, 2. \end{aligned} \tag{12}$$

表4 为三杆桁架设计问题最优结果对比. 对比结果表明: BA 取得的结果最好, 明显优于对比算法,

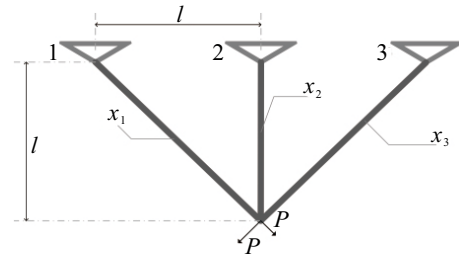


图4 三杆桁架设计

表4 三杆桁架设计问题最优结果对比

algorithm	x_1	x_2	best volume
BA	0.785 3	0.417 9	263.909 2
OOA	0.793 5	0.395 7	263.995 8
WFO	0.794 3	0.394 0	264.075 2

其解决三杆桁架设计问题具有良好的效果.

4 结论和未来工作

本文介绍了一种新的基于群体的元启发式算法, 即海狸算法 (BA). 该算法的灵感来源于海狸的伐木行为, 海狸根据各自的分工展开不同的活动, 通过共享和交流群体信息来寻找问题的最优解. 通过对 CEC 2017 基准测试函数和两个工程优化问题的研究, 评估了 BA 的性能. 在 CEC 2017 基准测试函数的实验中, 将 BA 与其他 7 种算法的结果进行了比较分析. 实验结果表明, BA 在所有算法中表现最为成功, 在大部分测试函数中获得了最佳解. 在工程优化问题的实验中, 将 BA 与其他两种算法的结果进行了比较分析. 实验结果表明, BA 在工程优化问题中均取得了最好的结果, 优化效果优于对比算法.

未来的工作可以开发 BA 的二进制和多目标版本, 以解决离散问题和多目标优化问题. 在应用方面, BA 可用于解决轨迹规划、目标分配等问题.

参考文献 (References)

- [1] 朱光宇, 张德颂. 基于强化学习的遗传算法求解一种新的钻削路径优化问题[J]. 控制与决策, 2024, 39(2): 697-704.
(Zhu G Y, Zhang D S. Genetic algorithm based on reinforcement learning for a novel drilling path optimization problem[J]. Control and Decision, 2024, 39(2): 697-704.)
- [2] 孙倩, 王磊, 徐庆征, 等. 一种基于超粒子引导的自适应知识迁移多任务差分进化算法[J]. 控制与决策, 2024, 39(1): 26-38.
(Sun Q, Wang L, Xu Q Z, et al. A super-particle guided multifactorial differential evolution algorithm with adaptive knowledge transfer[J]. Control and Decision, 2024, 39(1): 26-38.)
- [3] Zhao J, Han C Z, Wei B. Binary particle swarm optimization with multiple evolutionary strategies[J]. Science China Information Sciences, 2012, 55(11): 2485-2494.

- [4] Gandomi A H, Alavi A H. Multi-stage genetic programming: A new strategy to nonlinear system modeling[J]. *Information Sciences*, 2011, 181(23): 5227-5239.
- [5] Mallipeddi R, Mallipeddi S, Suganthan P N. Ensemble strategies with adaptive evolutionary programming[J]. *Information Sciences*, 2010, 180(9): 1571-1581.
- [6] Alirezanejad M, Enayatifar R, Motameni H, et al. GSA-LA: Gravitational search algorithm based on learning automata[J]. *Journal of Experimental & Theoretical Artificial Intelligence*, 2021, 33(1): 109-125.
- [7] Xu F M, Wang S H. A hybrid simulated annealing thresholding algorithm for compressed sensing[J]. *Signal Processing*, 2013, 93(6): 1577-1585.
- [8] Hatamlou A. Black hole: A new heuristic optimization approach for data clustering[J]. *Information Sciences*, 2013, 222: 175-184.
- [9] Mirjalili S, Mirjalili S M, Hatamlou A. Multi-verse optimizer: A nature-inspired algorithm for global optimization[J]. *Neural Computing and Applications*, 2016, 27(2): 495-513.
- [10] Kaveh A, Talatahari S. A novel heuristic optimization method: Charged system search[J]. *Acta Mechanica*, 2010, 213(3): 267-289.
- [11] 毛文平, 李帅永, 谢现乐, 等. 基于自适应机制改进蚁群算法的移动机器人全局路径规划[J]. *控制与决策*, 2023, 38(9): 2520-2528.
(Mao W P, Li S Y, Xie X L, et al. Global path planning of mobile robot based on adaptive mechanism improved ant colony algorithm[J]. *Control and Decision*, 2023, 38(9): 2520-2528.)
- [12] 申晓宁, 姚铨滨, 徐继勇, 等. 考虑新技能学习机制的软件项目调度人工蜂群算法[J]. *控制与决策*, 2023, 38(3): 790-796.
(Shen X N, Yao C B, Xu J Y, et al. Artificial bee colony algorithm for software project scheduling considering new skills learning mechanism[J]. *Control and Decision*, 2023, 38(3): 790-796.)
- [13] 程美英, 钱乾, 倪志伟. 基于种群多样性控制的多级信息迁移多任务优化粒子群算法[J]. *控制与决策*, 2024, 39(3): 728-738.
(Cheng M Y, Qian Q, Ni Z W. Multi-level information transfer multi-task PSO based on population diversity control[J]. *Control and Decision*, 2024, 39(3): 728-738.)
- [14] 王延峰, 廖荣航, 梁恩豪, 等. 基于围攻机制的改进鲸鱼优化算法[J]. *控制与决策*, 2023, 38(10): 2773-2782.
(Wang Y F, Liao R H, Liang E H, et al. Improved whale optimization algorithm based on siege mechanism[J]. *Control and Decision*, 2023, 38(10): 2773-2782.)
- [15] Seyyedabbasi A, Kiani F. Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems[J]. *Engineering with Computers*, 2023, 39(4): 2627-2651.
- [16] Hood G A, Larson D G. Ecological engineering and aquatic connectivity: A new perspective from beaver-modified wetlands[J]. *Freshwater Biology*, 2015, 60(1): 198-208.
- [17] Wen X D, Liu X D, Yu C H, et al. IOOA: A multi-strategy fusion improved osprey optimization algorithm for global optimization[J]. *Electronic Research Archive*, 2024, 32(3): 2033-2074.
- [18] Trojovská E, Dehghani M, Trojovský P. Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm[J]. *IEEE Access*, 2022, 10: 49445-49473.
- [19] Song H M, Xing C, Wang J S, et al. Improved pelican optimization algorithm with chaotic interference factor and elementary mathematical function[J]. *Soft Computing*, 2023, 27(15): 10607-10646.
- [20] Xi M Y, Song Q X, Xu M, et al. Binary African vultures optimization algorithm for various optimization problems[J]. *International Journal of Machine Learning and Cybernetics*, 2023, 14(4): 1333-1364.
- [21] Pandey A, Rajan A, Nandi A, et al. Lifetime enhancement of sensor networks by the moth flame optimization[J]. *Wireless Personal Communications*, 2021, 118(4): 2807-2820.
- [22] Dereli S. A new modified grey wolf optimization algorithm proposal for a fundamental engineering problem in robotics[J]. *Neural Computing and Applications*, 2021, 33(21): 14119-14131.
- [23] Luo K P. Water flow optimizer: A nature-inspired evolutionary algorithm for global optimization[J]. *IEEE Transactions on Cybernetics*, 2022, 52(8): 7753-7764.

作者简介

廖想 (1984-), 男, 副教授, 博士, 主要研究方向为高维多目标优化方法、综合能源系统数字孪生技术, E-mail: liaoxiang@hbut.edu.cn;

周安琪 (1999-), 女, 硕士生, 主要研究方向为综合能源系统优化控制方法, E-mail: 102210388@hbut.edu.cn;

刘珂 (1997-), 男, 硕士生, 主要研究方向为综合能源系统优化控制方法, E-mail: 102200231@hbut.edu.cn;

郑梓煜 (1998-), 男, 硕士生, 主要研究方向为综合能源系统优化控制方法, E-mail: 102200224@hbut.edu.cn;

付波 (1973-), 男, 教授, 博士, 主要研究方向为综合能源系统、模式识别, E-mail: fubofanxx@mail.hbut.edu.cn;

李超顺 (1983-), 男, 教授, 博士, 博士生导师, 主要研究方向为水电能源优化运行、水风光不确定性预报与联合优化运行, E-mail: cslu@hust.edu.cn.