

多策略改进的蜣螂优化算法及其应用

郭 琴¹, 郑巧仙²⁺

1. 湖北大学 计算机与信息工程学院, 武汉 430062

2. 湖北大学 网络空间安全学院, 武汉 430062

+ 通信作者 E-mail: zqxlm1978@163.com

摘要: 蜣螂优化算法(DBO)是近年提出的智能优化算法,与其他优化算法一样,DBO也存在收敛精度低、易陷入局部最优等缺点。针对DBO的这些局限性,提出一种多策略改进的蜣螂优化算法(MIDBO)。首先,改进雏球和偷窃蜣螂对局部最优解和全局最优解的接受程度,使其根据自身搜索能力动态变化,既提升了种群质量又保持了适应度高的个体的良好搜索能力;其次,融合麻雀搜索算法中的追随者位置更新机制对算法进行扰动,并用贪婪策略更新位置,提升了算法的收敛精度;最后,当算法陷入停滞时引入柯西高斯变异策略,提高了算法跳出局部最优解的能力。仿真实验基于20个基准测试函数和CEC2019测试函数,验证了3种改进策略的有效性,将所改进算法和对比较算法的优化结果进行收敛性分析和Wilcoxon秩和检验,证明了MIDBO具有良好的寻优性能和鲁棒性。将MIDBO运用在汽车碰撞优化问题的求解上,进一步验证了MIDBO在求解实际工程问题中的有效性和可靠性。

关键词: 蜣螂优化算法;局部最优解;麻雀搜索算法;柯西高斯变异;汽车碰撞优化问题;Wilcoxon秩和检验

文献标志码: A **中图分类号:** TP301.6

Multi-strategy Improved Dung Beetle Optimizer and Its Application

GUO Qin¹, ZHENG Qiaoxian²⁺

1. School of Computer and Information Engineering, Hubei University, Wuhan 430062, China

2. School of Cyber Science and Technology, Hubei University, Wuhan 430062, China

Abstract: Dung beetle optimizer (DBO) is an intelligent optimization algorithm proposed in recent years. Like other optimization algorithms, DBO also has disadvantages such as low convergence accuracy and easy to fall into local optimum. A multi-strategy improved dung beetle optimizer (MIDBO) is proposed. Firstly, it improves acceptance of local and global optimal solutions by brood balls and thieves, so that the beetles can dynamically change according to their own searching ability, which not only improves the population quality but also maintains the good searching ability of individuals with high fitness. Secondly, the follower position updating mechanism in the sparrow search algorithm is integrated to disturb the algorithm, and the greedy strategy is used to update the location, which improves the convergence accuracy of the algorithm. Finally, when the algorithm stagnates, Cauchy Gaussian variation strategy is introduced to improve the ability of the algorithm to jump out of the local optimal solution. Based on 20 benchmark test functions and CEC2019 test function, the simulation experiment verifies the effectiveness of the three improved strategies. The convergence analysis of the optimization results of the improved algorithm and the comparison algorithms and Wilcoxon rank sum test prove that MIDBO has good optimization performance and robustness. The validity and reliability of MIDBO in solving practical engineering problems are further verified by applying

基金项目: 国家自然科学基金(61803149)。

This work was supported by the National Natural Science Foundation of China (61803149).

收稿日期: 2023-08-06 **修回日期:** 2023-11-16

MIDBO to the solution of automobile collision optimization problems.

Key words: dung beetle optimization algorithm; local optimal solution; sparrow search algorithm; Cauchy Gaussian variation; car collision optimization problems; Wilcoxon rank sum test

由于工程领域的优化问题日益复杂,传统的精确算法,例如动态规划法^[1]、分支定界法^[2],虽然能求到问题的最优解,但其求解复杂度随着问题规模的扩大而呈指数级增长。启发式算法与传统精确算法比,具有结构简单、易实现且高效等特点,群智能优化算法是一类启发式算法,在求解复杂优化问题时能提供可靠的解决方案,已广泛运用于可再生能源^[3]、无人机路径规划^[4]等领域。近年来,学者们通过观察、模拟自然界中一些事物或生物的生活习性提出各种群智能优化算法,例如鲸鱼优化算法(whale optimization algorithm, WOA)^[5]、蜣螂优化算法(dung beetle optimizer, DBO)^[6]、麻雀搜索算法(sparrow search algorithm, SSA)^[7]、爬行动物搜索算法(reptile search algorithm, RSA)^[8]、细菌觅食算法(bacterial foraging optimization algorithm, BFOA)^[9]、布谷鸟搜索算法(cuckoo search algorithm, CS)^[10]。然而,这些群智能优化算法都存在收敛精度低、收敛速度慢等缺点。为此,各种改进算法应运而生:文献[11]提出一种融合透镜成像反向学习的精英池侏儒猫鼬优化算法(improved dwarf mongoose optimization, IDMO),使用透镜成像反向学习策略,避免算法在迭代过程中陷入局部最优,在阿尔法组觅食时引入精英池策略,提高了算法的收敛精度;文献[12]利用Cat映射初始化种群,增加种群多样性,通过改进缩放因子动态调整发现者数量和意识到风险的麻雀数量,再根据个体适应度值进行柯西变异或Tent混沌扰动,提升了算法的局部逃逸能力;文献[13]提出一种多策略融合的改进麻雀搜索算法,采用精英混沌反向学习策略生成初始种群,并结合鸡群算法的随机跟随策略,优化麻雀搜索算法中跟随者的位置更新过程。

蜣螂优化算法(DBO)是Xue等于2022年提出的一种新型群智能优化算法^[6]。现已有学者将此算法用在实际问题中,文献[14]将改进的蜣螂优化算法用在常见的3个工程设计问题中。文献[15]将蜣螂优化算法用于空气质量预测模型中寻找超参数,避免了超参数设置的盲目性,实验结果表明融入了蜣螂优化算法的模型优于其他对比模型。文献[16]提出一种改进的蜣螂优化算法,使用Beta分布动态生成反射解,以更好地探索搜索空间并让个体跳出局部最优,

并引入Levy分布来处理越界粒子,最后,采用两种不同的交叉算子来改进偷窃蜣螂的位置更新,实验表明该改进算法运用在三维路径规划中比其他启发式算法的效果更好。

与其他群智能算法不同的是,DBO的位置更新是并行的,该算法受启发于蜣螂的滚球、繁育、觅食、偷窃等4种行为,将种群分为滚球蜣螂、繁育蜣螂、小蜣螂、偷窃蜣螂,依据个体的不同类别,采用4种不同的位置更新方式。与其他群智能优化算法一样,DBO也存在收敛精度低、收敛速度慢、易陷入局部最优等缺陷^[15]。本文对此做出如下改进:首先根据雏球和偷窃蜣螂的个体自身适应度改进雏球对局部最优解的接受程度和偷窃蜣螂对全局最优解的接受程度,适应度较低的个体对最优解的接受程度高,令适应度低的个体大跨步向最优解靠近,可提升种群质量。适应度较高的个体对最优解的接受程度低,可令该个体在小范围内继续搜索,保持自身良好的搜索能力。其次,由于麻雀搜索算法具有全局搜索能力强的优点^[7],在种群完成一次迭代后,采用麻雀搜索算法中的追随者位置更新机制对种群进行一定的扰动,并使用贪婪策略更新种群位置,提高了算法的收敛精度。最后,若算法5次迭代后全局最优解没有改变,则判定算法陷入停滞,使用柯西高斯变异助使算法跳出局部最优解。将改进后的算法进行基准函数测试对比实验,与其他智能优化算法相对比,验证了本文改进策略的有效性,此外将改进后的算法运用在汽车碰撞优化问题中,证明了该算法的工程适用性。

1 蜣螂优化算法

蜣螂优化算法是受蜣螂滚动粪球、繁育、觅食、偷窃等行为的启发而提出的,这4种行为分别对应滚球蜣螂、繁育蜣螂、小蜣螂、偷窃蜣螂,4种蜣螂采用不同的位置更新方式,共同寻找问题的最优解,4种蜣螂在种群中所占的比例分别为20%、20%、25%、35%。

(1) 滚球蜣螂

自然界中蜣螂会一直滚动粪便,把它做成一个粪球,滚动粪球需要光引导其导航,以保证蜣螂沿着直线滚动粪球,但蜣螂会有一定的概率遇到障碍物,

其概率如式(1)所示,没有遇到障碍物滚动粪球时位置更新如式(2)所示,遇到障碍物时需要通过在粪球上跳舞来重新定位方向,以此得到新的滚动路线,跳舞重定位方向时位置更新如式(3)所示。

$$\alpha = \begin{cases} 1, \eta > \lambda \\ -1, \eta < \lambda \end{cases} \quad (1)$$

$$X_i(t+1) = X_i(t) + \alpha \times k \times X_i(t-1) + b \times \Delta X \quad (2)$$

$$X_i(t+1) = X_i(t) + \tan \theta |X_i(t) - X_i(t-1)| \quad (3)$$

式(1)中 η 是(0,1)之间的随机数, λ 表示蛭螂遇到障碍物的概率, $\lambda \in (0,1)$ 。式(2)中, k 是偏离系数, $k \in (0,0.02]$, b 是常量, $b \in (0,1)$, ΔX 表示光的强度, $\Delta X = |X_i - X^w|$, 其中 X^w 代表种群中的最差位置。式(3)中, $\theta \in [0, \pi]$, 表示偏离系数, 当 $\theta \in [0, \pi/2, \pi]$ 时, $\tan \theta$ 无意义, 位置不更新。

(2) 繁育蛭螂

蛭螂会把粪球滚动到一个安全的位置隐藏起来, 以便为雌性蛭螂产卵提供一个安全环境和保护后代, 对安全区域的定义如下式:

$$\begin{cases} Lb^* = \max(X^* \times (1-R), Lb) \\ Ub^* = \min(X^* \times (1+R), Ub) \end{cases} \quad (4)$$

其中, X^* 表示本次迭代群体最优位置(局部最优位置) $R = t/T$, Lb 、 Ub 分别表示可行区域的下界和上界; Ub^* 、 Lb^* 为安全区域的上界和下界。

寻找好安全区域后, 雌性蛭螂会在安全区域内产卵, 每只雌性蛭螂会产生一个雏球, 雏球的位置更新如式(5)所示, 当雏球位置超出安全区域时, 做式(6)处理。

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \quad (5)$$

$$B_i = \begin{cases} Lb^*, B_i < Lb^* \\ Ub^*, B_i > Ub^* \end{cases} \quad (6)$$

式(5)中, b_1 和 b_2 为 $1 \times D$ 的独立随机向量, D 为优化问题的维数。

(3) 小蛭螂

雏球会长成小蛭螂, 从地里爬出来觅食, 小蛭螂觅食位置不是任意的, 会选择一个最佳觅食区域觅食, 最佳觅食区域寻找到食物的概率比较大, 最佳觅食区域的范围定义如式(7)所示; 确定最佳觅食区域后, 小蛭螂进行觅食时位置更新如式(8)所示。

$$\begin{cases} Lb^b = \max(X^b \times (1-R), Lb) \\ Ub^b = \min(X^b \times (1+R), Ub) \end{cases} \quad (7)$$

$$X_i(t+1) = X_i(t) + C_1 \times (X_i(t) - Lb^b) + C_2 \times (X_i(t) - Ub^b) \quad (8)$$

式(7)中, X^b 为全局最优位置, Lb^b 、 Ub^b 为最佳

觅食区域的下界和上界; 式(8)中, C_1 为 $1 \times D$ 、服从正态分布的随机数向量, C_2 为取值范围为(0,1)的随机向量。

(4) 偷窃蛭螂

偷窃蛭螂会偷取其他蛭螂的粪球, 偷窃蛭螂并不是在任意位置行窃, 而是在最佳偷窃地点, 也就是全局最优位置附近偷窃, 偷窃蛭螂的位置更新如下:

$$X_i(t+1) = X^b + S \times g \times (|X_i(t) - X^*| + |X_i(t) - X^b|) \quad (9)$$

式中, g 是大小为 $1 \times D$ 且服从正态分布的随机向量, S 表示一个常量。

2 多策略改进的蛭螂优化算法

2.1 改进雏球和偷窃蛭螂对最优解的接受程度

在基本蛭螂优化算法中, 雏球根据局部最优位置进行位置更新, 如式(5)所示, 偷窃蛭螂根据全局最优位置更新位置, 如式(9)所示, 由此可知, 雏球和偷窃蛭螂会不停地向局部最优位置和全局最优位置靠近, 但在算法前期, 应尽量扩大群体的搜索范围, 快速向局部最优位置或全局最优位置靠近虽然能使算法快速收敛, 但也有一定的概率令算法陷入停滞, 导致无法收敛到全局最优解。因此, 设计一种根据蛭螂搜索能力的强弱(适应度大小)动态改变雏球和偷窃蛭螂对局部最优位置或全局最优位置的接受程度的策略。具体方法如下: 将蛭螂群体根据搜索能力(适应度值优劣)排序, 记第 i 只蛭螂在第 t 次迭代时的搜索能力排名为 $rank_i(t)$ 位, $i, k = 1, 2, \dots, pop$, pop 为种群规模, 例如 $rank_i(t) = 1$ 代表在第 t 次迭代时, 第 i 只蛭螂的搜索能力在种群中排第1位(适应度最优), $rank_i(t) = pop$ 代表在第 t 次迭代时, 第 i 只蛭螂的适应度在种群中排第 pop 位(适应度最差), 雏球和偷窃蛭螂根据自身搜索能力的更新位置方式如下:

$$B_i(t+1) = \frac{rank_i(t) - 1}{pop - 1} \times X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \quad (10)$$

$$X_i(t+1) = \frac{rank_i(t) - 1}{pop - 1} \times X^b + S \times g \times (|X_i(t) - X^*| + |X_i(t) - X^b|) \quad (11)$$

由式(10)、式(11)可知, 搜索能力越强的个体更新位置时受局部最优位置或全局最优位置的影响越小, 这可以令搜索能力强的个体继续保持自身良好的搜索能力, 在小范围内继续搜索, 防止步长过大跳过更优解; 相反, 搜索能力弱的个体更新位置时会较大跨步地向局部最优位置和全局最优位置靠近, 可

以提升种群整体质量。

2.2 融合麻雀搜索算法追随机制的扰动策略

2.2.1 麻雀搜索算法追随者位置更新机制

麻雀搜索算法(SSA)是受麻雀觅食行为和反捕食行为启发而提出的新型群智能算法,麻雀根据个体适应度的不同分为发现者、追随者和侦察者。发现者适应度值高,占群体数量的10%~20%,为追随者提供觅食区域与方向。在麻雀种群中,追随者负责跟随发现者寻找食物,追随者的位置更新如下式^[7]:

$$X_i(t+1) = \begin{cases} Q \times \exp\left(\frac{X_w - X_i(t)}{i^2}\right), & i > \frac{pop}{2} \\ X_p(t+1) + |X_i(t) - X_p(t+1)| \times A^+ \times L, & i \leq \frac{pop}{2} \end{cases} \quad (12)$$

其中, Q 是服从正态分布的随机数, $X_p(t+1)$ 表示目前发现者的最优位置, X_w 表示全局最差位置, A 为 $1 \times d$ 的矩阵,矩阵中元素为1或-1, $A^+ = A^T(AA^T)^{-1}$, A^T 为 A 的转置矩阵, L 表示一个元素全为1的 $1 \times d$ 的矩阵。当 $i > pop/2$ 时,表明第 i 个追随者没有获得食物,适应度较低,处于饥饿状态,此时需要飞往其他地方寻找食物。

2.2.2 基于追随者位置更新的扰动策略

SSA具有收敛精度高、收敛速度快、鲁棒性强等特点,在函数优化问题方面,优于粒子群算法、灰狼优化算法等群智能算法^[7]。因此,将麻雀搜索算法中追随者的位置更新机制作为一种扰动策略引入蜣螂优化算法,定义麻雀搜索算法中的发现者最优位置对应蜣螂优化算法中的全局最优位置。若扰动性太强,群体容易陷入混乱无序状态;若扰动性太弱,扰动策略将不能发挥作用;因此,根据迭代深度自适应改变扰动概率,并采用贪婪策略决定是否保留此次扰动,扰动概率如式(13)所示:

$$pv = \frac{2 \left(1 - \left(\frac{t}{T} \right)^{\frac{3}{2}} \right)}{3} \quad (13)$$

扰动概率曲线图如图1所示,由图1可知,在迭代前期,扰动概率较大,可充分发挥麻雀追随机制扰动的作用,达到提升种群质量、提高算法寻优性能的作用;在迭代后期,扰动概率逐渐减小,起到加快算法收敛速度的效果。具体扰动步骤如下:(1)根据式(13)计算第 i 个个体的扰动概率 pv 。(2)生成随机数 $r \in (0, 1)$,若 $r < pv$,进行步骤(3),否则第 i 个个体不进行扰动。(3)生成随机数 $r_1 \in (0, 1)$,若 $r_1 < ST$,根据式

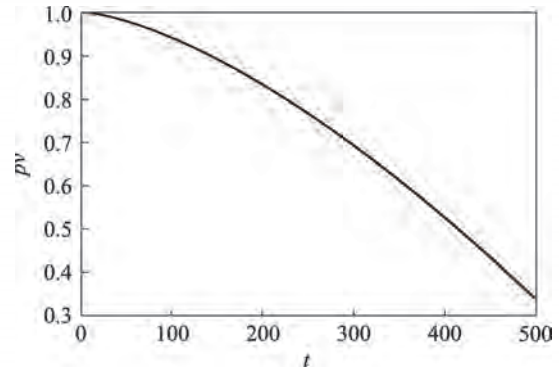


图1 扰动概率曲线图

Fig.1 Disturbance probability curve

(12)中的第二个公式生成新解,否则根据式(12)中的第一个公式生成新解,经过大量实验证明 $ST=0.7$ 时效果最好,因此本文取 $ST=0.7$,这意味着群体中有70%的个体往群体最优位置方向变异,30%个体朝比自身更差的位置变异。(4)计算新解的适应度值,若优于扰动前适应度值,用新解代替扰动前的解,否则,该次扰动无效。

2.3 柯西高斯变异

引入麻雀搜索算法追随机制的扰动主要作用是使群体能更充分地遍历解空间,提升种群质量,类似于混沌扰动;在多峰值情况下,蜣螂优化算法和其他群智能优化一样,容易陷入局部最优解,柯西高斯变异是为应对多峰值情况而提出的辅助策略,加大算法跳出局部最优解的概率^[13],提升算法的寻优精度。在每次麻雀搜索算法追随机制扰动后,判断群体是否早熟,判断方式为:若群体经过5次迭代后最优解不变,则判定算法陷入停滞(早熟),对整个群体采用柯西高斯变异策略,令群体跳出局部最优解,柯西高斯变异方法如下式:

$$X_{new_i}(t) = X_i(t) \times [1 + \beta_1 \text{Cauchy}(0, 1) + \beta_2 \text{Gauss}(0, 1)] \quad (14)$$

式中, $\text{Cauchy}(0, 1)$ 为服从柯西分布的随机数, $\text{Gauss}(0, 1)$ 为服从高斯分布的随机数; $X_{new_i}(t)$ 为变异后的位置, $\beta_1 = 1 - t^2/T^2$, $\beta_2 = t^2/T^2$,柯西分布随机数取值范围很广,因此对个体位置干扰能力强,可增强算法的全局搜索能力,而高斯分布较密集,大概率取值在 $[-2, 2]$ 左右,对个体位置干扰性较小,可以增强算法的局部开发能力, β_1 由1逐渐减小到0, β_2 由0逐渐增加到1;说明迭代前期,柯西变异发挥作用更大,扩大了群体的搜索范围,在算法后期,柯西变异作用逐渐减弱,高斯变异发挥主要作用,让个体在小范围内搜索,增强群体的局部搜索能力,让算法快速收敛。

2.4 算法复杂度分析

DBO算法的复杂度主要取决于以下3个过程:初始化阶段、适应度评估、种群位置更新。定义最大迭代次数 T 、种群规模 N 、问题维度 D ;初始化种群复杂度为 $O(N \times D)$,适应度评估复杂度为 $O(N \times T)$,种群位置更新包括4种蛭螂的位置更新,分别为滚球蛭螂、繁育蛭螂、小蛭螂、偷窃蛭螂的位置更新,复杂度为 $O(N_i \times T \times D)$, N_i 表示第 i 种类型蛭螂的数量,因此DBO算法的复杂度为 $O(\max(N_i) \times T \times D)$ 。

MIDBO (multi-strategy improved dung beetle optimizer) 在 DBO 基础上加了三种策略。第一种策略为改进雏球和偷窃蛭螂位置更新方式,没有增加额外的位置更新和适应度评估,因此没有增加复杂度。第二种策略和第三种策略分别为融合麻雀搜索算法追随机制的扰动策略和柯西高斯变异,这两种策略的位置更新复杂度最大为 $O(N \times T \times D)$,适应度评估最大复杂度为 $O(N \times T)$,因此MIDBO的复杂度为 $O(N \times T \times D)$,由此可知MIDBO与DBO复杂度相同。

2.5 算法流程

MIDBO伪代码如下,具体实现流程图如图2所示。

初始化 T 、 pop 、 α 等参数;初始化种群,评估其适应度值,初始化局部最优位置 X_{lbest} ,局部最优适应度 $f(X_{lbest})$,全局最优位置 X_{gbest} ,全局最优适应度 $f(X_{gbest})$;计算滚球蛭螂规模 N_1 ,繁育蛭螂规模 N_2 ,小蛭螂规模 N_3 ,偷窃蛭螂规模 N_4 ;

```

for t = 1 → T
    for i = 1 → N1
        按照式(1)~式(3)更新滚球蛭螂位置及其适应度;
    end for
    for i = 1 → N2
        按照式(4)、式(10)、式(6)更新雏球位置及其适应度;
    end for
    for i = 1 → N3
        按照式(7)、式(8)更新小蛭螂位置及其适应度;
    end for
    for i = 1 → N4
        按照式(11)更新偷窃蛭螂位置及其适应度;
    end for
    更新 Xlbest、f(Xlbest)、Xgbest、f(Xgbest)
    for i = 1 → pop
        生成随机数 r ∈ (0, 1), 根据式(13)计算扰动概率 pv;
        if r < pv

```

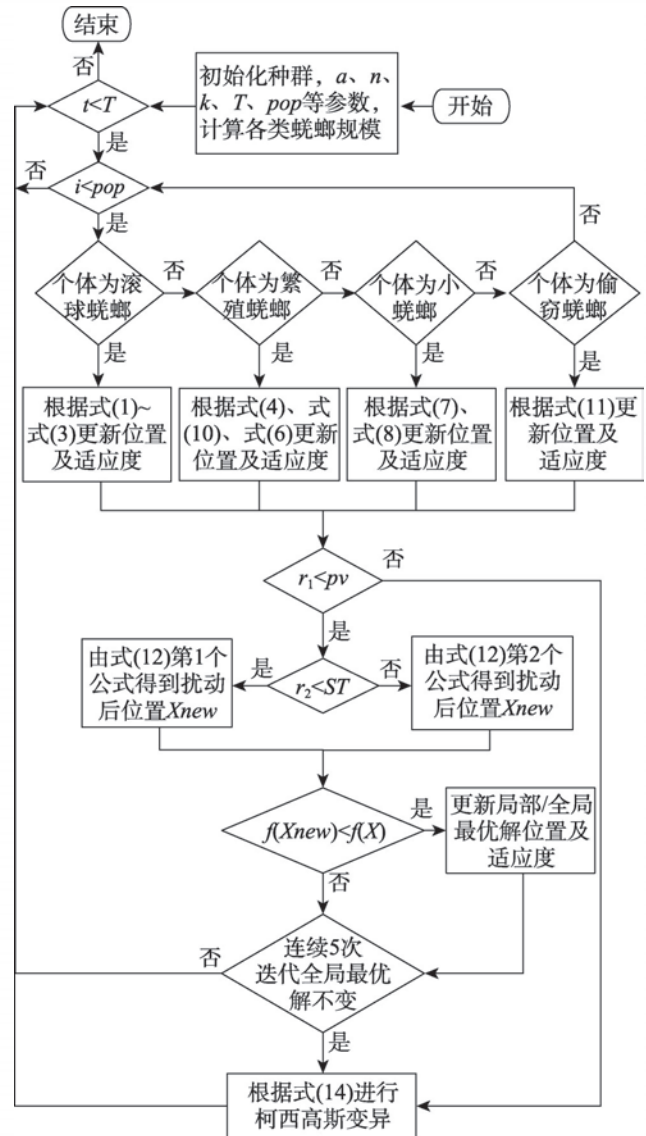


图2 MIDBO算法流程图
Fig.2 Flowchart of MIDBO

```

生成随机数 r1 ∈ (0, 1);
if r1 > ST
    按照式(12)中第一个式子得出扰动后位置 Xnew;
else
    按照式(12)中第二个式子得出扰动后位置 Xnew;
end if
if f(Xnew) < f(X)
    更新 Xlbest、f(Xlbest)、Xgbest、f(Xgbest);
end
end if
end for
if f(Xgbest) 5次没有改变

```

根据式(14)对群体进行柯西高斯变异扰动;
更新 X_{lbest} 、 $f(X_{lbest})$ 、 X_{gbest} 、 $f(X_{gbest})$;
end if
end for

3 仿真实验及结果分析

3.1 测试函数说明

采用表1的20个标准测试函数进行性能测试,其中 $F1\sim F7$ 为高维单峰值函数,只有一个全局最优解,可检验算法的开发能力; $F8\sim F13$ 为高维多峰值函数,有一个全局最优解和多个局部最优解,可检验算法的全局搜索能力和局部逃逸能力; $F14\sim F20$ 为固定维度多峰函数,可以验证算法探索和开发能力之间的平衡;通过3种不同类型的测试函数,充分验证 MIDBO 算法的寻优性能;为确保公平性,所有算法的实验仿真均使用相同的软硬件平台,在 Matlab 2021b 中实现;所有算法统一设定最大迭代次数 $T=500$,种群数量 $pop=30$,各算法独立运行30次,取30次寻优结果的最好值(Best)、平均值(Mean)、标准差(Std)作为评价指标。

3.2 改进策略有效性分析

MIDBO 在 DBO 基础上增加了3种改进方式,将 DBO 与这3种改进方式分别融合得到3种不同的算法:

(1)在 DBO 中改进雉球和偷窃蜣螂对最优解的接受程度得到 DBO1;

(2)在 DBO 中加入融合麻雀搜索算法追随者位置更新机制的扰动策略得到 DBO2;

(3)在 DBO 中加入柯西高斯变异策略得到 DBO3。

将 MIDBO 与 DBO、DBO1、DBO2、DBO3 以及文献[14]提出的 MSADBO(dung beetle optimization algorithm guided by improved sine algorithm)等5种算

法在表1的20个测试函数进行实验。各算法独立运行30次寻优结果的最好值(Best)、平均值(Mean)、标准差(Std)如表2所示,其中 Best 反映了算法的精确性,Mean 和 Std 反映了算法的稳定性。

表2中加粗数据表示比 DBO 平均寻优精度低的值,由表2可知,与 DBO 相比,DBO1、DBO2、DBO3 分别在10、15、12个函数上的平均寻优精度有所提升,而 MIDBO 在18个函数较 DBO 有所提升,2个与 DBO 相当。与 DBO 相比,DBO1、DBO2、DBO3 的3个评价指标都有很大程度的提高,其中 DBO1 在函数 $F1$ 、 $F3$ 上的最优收敛值达到了理论最优解,说明改进雉球和偷窃蜣螂对最优解的接受程度可以提升算法的收敛精度;与 DBO 相比,DBO2 在 $F1\sim F13$ 这13个测试函数上的最优收敛值、平均收敛精度都有大幅度提高,尤其在多峰函数 $F12$ 、 $F13$ 上,标准差只在 $F8$ 上略差于 DBO,其余都比 DBO 标准差更小,说明在 DBO 中引入麻雀搜索算法追随机制可以结合 SSA 全局搜索能力强的优势,并能提高算法的稳定性;DBO3 在函数 $F8$ 上的最优收敛值达到了理论最优解,平均收敛精度也与理论最优解很接近, $F8$ 具有很多极小值,在(420.968 7, ..., 420.968 7)取得最优值,优化条件较高,很多算法都难以找到其全局最优解,说明 DBO 融合柯西高斯变异可以提高 DBO 的寻优性能,增大算法跳出局部最优解的概率。对于固定维度函数 $F14\sim F20$,从 $F14$ 、 $F18\sim F20$ 可以得出,融入了3种策略的 MIDBO 在平均收敛精度上都达到了理论最优解,且标准差比 DBO 和3种融合一个策略的小;对于函数 $F15$ 、 $F17$,虽然 MIDBO 没有达到理论最优解,但与基本 DBO 相比,与理论最优解更加接近;说明融入3种策略的 MIDBO 能很好地平衡算法的勘探和开发能力。与 DBO 相比,融合了3种改

表1 测试函数
Table 1 Test functions

函数	名称	维度	范围	最优值	函数	名称	维度	范围	最优值
$F1$	Sphere	30/100	[-100,100]	0	$F11$	Griewank	30/100	[-600,600]	0
$F2$	Schwefel2.22	30/100	[-10,10]	0	$F12$	Penalized	30/100	[-50,50]	0
$F3$	Schwefel1.2	30/100	[-100,100]	0	$F13$	Penalized2	30/100	[-50,50]	0
$F4$	Schwefel2.21	30/100	[-100,100]	0	$F14$	Foxholes	2	[-65,65]	0.998 0
$F5$	Rosenbrock	30/100	[-30,30]	0	$F15$	Kowalik's	4	[-5,5]	0.000 3
$F6$	STEP	30/100	[-100,100]	0	$F16$	Six-Hump Camelback	2	[-5,5]	-1.310 6
$F7$	Quartic	30/100	[-1.28,1.28]	0	$F17$	Hartman6	6	[0,1]	-3.322 0
$F8$	Schwefel	30/100	[-500,500]	-12 569.5	$F18$	Shekel5	4	[0,10]	-10.153 2
$F9$	Rastrigin	30/100	[-5.12,5.12]	0	$F19$	Shekel7	4	[0,10]	-10.402 9
$F10$	Ackley	30/100	[-32,32]	0	$F20$	Shekel10	4	[0,10]	-10.536 4

表2 不同策略寻优结果

Table 2 Optimization results of different strategies

函数	指标	DBO	DBO1	DBO2	DBO3	MSADBO	MIDBO
F1	Best	1.04E-168	0	2.16E-282	2.23E-164	0	0
	Mean	3.16E-108	6.12E-110	4.15E-190	4.34E-100	0	0
	Std	1.73E-107	3.35E-109	0	2.38E-99	0	0
F2	Best	1.05E-84	1.63E-110	9.95E-133	6.61E-82	0	0
	Mean	9.34E-54	1.37E-57	4.95E-93	1.67E-58	5.95E-292	0
	Std	5.01E-53	7.50E-57	2.70E-92	9.13E-58	0	0
F3	Best	5.40E-151	0	1.04E-259	6.45E-146	0	0
	Mean	7.99E-69	4.46E-106	1.19E-149	1.31E-81	0	0
	Std	4.38E-68	2.44E-105	6.52E-149	7.16E-81	0	0
F4	Best	3.06E-78	2.05E-174	1.22E-171	6.77E-78	0	0
	Mean	1.09E-54	9.41E-70	1.34E-83	1.04E-53	2.49E-278	0
	Std	3.69E-54	4.86E-69	7.28E-83	5.33E-53	0	0
F5	Best	2.55E+01	2.79E+01	0	2.54E+01	2.49E+01	0
	Mean	2.58E+01	2.84E+01	2.15E-08	2.60E+01	2.54E+01	1.42E-05
	Std	3.11E-01	2.53E-01	6.14E-08	3.41E-01	2.42E-01	5.90E-05
F6	Best	9.63E-06	1.02E+00	0	6.25E-06	2.41E-03	0
	Mean	9.73E-04	1.59E+00	7.31E-17	8.62E-03	4.25E-02	6.43E-08
	Std	2.04E-03	3.32E-01	1.87E-16	4.52E-02	8.97E-02	1.88E-07
F7	Best	5.89E-05	4.64E-05	2.41E-05	6.99E-05	1.25E-05	8.70E-06
	Mean	1.26E-03	9.53E-04	3.16E-04	7.45E-04	9.61E-05	3.59E-04
	Std	9.07E-04	6.23E-04	2.29E-04	5.53E-04	6.78E-05	3.34E-04
F8	Best	-1.25E+04	-1.16E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04
	Mean	-8.05E+03	-8.27E+03	-1.10E+04	-1.24E+04	-1.13E+04	-1.26E+04
	Std	1.58E+03	1.92E+03	1.79E+03	3.72E+02	8.63E+02	2.06E-03
F9	Best	0	0	0	0	0	0
	Mean	5.31E-01	0	0	0	0	0
	Std	2.04E+00	0	0	0	0	0
F10	Best	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Mean	8.88E-16	1.36E-15	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Std	0	1.23E-15	0	0	0	0
F11	Best	0	0	0	0	0	0
	Mean	0	0	0	0	0	0
	Std	0	0	0	0	0	0
F12	Best	2.27E-07	6.15E-02	1.57E-32	2.39E-07	7.98E-05	1.57E-32
	Mean	4.66E-03	9.03E-02	3.15E-15	1.21E-04	1.46E-03	4.50E-08
	Std	1.90E-02	2.25E-02	1.29E-14	5.84E-04	2.98E-03	2.29E-07
F13	Best	4.04E-04	1.73E+00	6.38E-22	5.40E-05	2.43E-03	1.35E-32
	Mean	4.76E-01	2.41E+00	1.03E-14	1.74E-01	2.87E-01	1.82E-08
	Std	3.80E-01	2.33E-01	3.26E-14	3.56E-01	4.00E-01	2.74E-08
F14	Best	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	Mean	1.39E+00	1.99E+00	1.55E+00	2.18E+00	1.13E+00	9.98E-01
	Std	8.07E-01	9.75E-01	2.16E+00	2.21E+00	5.03E-01	1.52E-08
F15	Best	3.07E-04	3.21E-04	3.07E-04	3.07E-04	3.08E-04	3.08E-04
	Mean	7.70E-04	5.48E-04	3.08E-04	5.15E-04	3.73E-04	3.21E-04
	Std	3.00E-04	1.46E-04	7.55E-07	3.00E-04	1.21E-04	1.24E-05
F16	Best	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Mean	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Std	6.32E-16	1.90E-05	2.21E-06	6.05E-16	3.81E-09	5.75E-05
F17	Best	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00
	Mean	-3.24E+00	-3.20E+00	-3.27E+00	-3.27E+00	-3.32E+00	-3.25E+00
	Std	8.21E-02	1.13E-01	5.92E-02	8.73E-02	6.05E-02	8.06E-02
F18	Best	-1.02E+01	-9.79E+00	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01
	Mean	-6.81E+00	-7.22E+00	-8.11E+00	-8.44E+00	-1.01E+01	-1.02E+01
	Std	2.41E+00	1.86E+00	2.54E+00	2.22E+00	5.23E-02	7.44E-03
F19	Best	-1.04E+01	-9.94E+00	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01
	Mean	-8.03E+00	-7.87E+00	-7.26E+00	-8.12E+00	-1.04E+01	-1.04E+01
	Std	2.79E+00	1.89E+00	2.62E+00	2.65E+00	7.12E-03	2.82E-03
F20	Best	-1.05E+01	-1.03E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01
	Mean	-8.04E+00	-8.09E+00	-9.64E+00	-9.97E+00	-1.05E+01	-1.05E+01
	Std	2.93E+00	2.10E+00	2.05E+00	1.65E+00	8.93E-03	5.59E-03

进方法的MIDBO在 $F1\sim F20$ 上的最优收敛值、平均收敛精度、标准差都比DBO有所提升,在 $F1\sim F4$ 、 $F8$ 、 $F14$ 、 $F18$ 的平均收敛精度都达到了理论最优解,未出现负优化的现象,说明3种策略的融合是有效的。

MIDBO与MSADBO的比较:MIDBO在 $F1\sim F4$ 上的平均收敛精度都收敛到了理论最优解,而MSADBO只在 $F1$ 、 $F3$ 上的平均寻优精度收敛到理论最优解,MSADBO与其他大部分优化算法一样,在较难求最优解的函数 $F5$ 上陷入了局部最优解,而MIDBO的最优收敛值达到了理论最优解0,平均收敛精度也达到了 $1.0E-5$,远超MSADBO;在 $F9\sim F11$ 上,两种算法的寻优性能一样,在多峰值函数 $F12$ 、 $F13$ 上,MIDBO表现出了优越的寻优性能,最好寻优结果达到了 $1.0E-32$,表明MIDBO有很强的全局搜索能力和局部逃逸能力;在固定维度函数 $F14\sim F20$ 上,MIDBO和MSADBO的最优收敛值一样,但在函数 $F14$ 、 $F18\sim F20$ 上,MIDBO的平均收敛精度和标准差都比MSADBO更优,函数 $F15$ 上MIDBO平均收敛精度也比MSADBO要高,函数 $F16$ 上两种算法相差无几。MIDBO只有在函数 $F7$ 、 $F17$ 上的表现略优于MSADBO,在 $F9\sim F11$ 上的表现与MSADBO无差异,MIDBO在其他函数上的寻优性能都强于MSADBO。

3.3 与其他群智能算法比较

为验证MIDBO算法的有效性,将MIDBO和其他群智能优化算法进行比较,对比算法包括经典且运用广泛的粒子群优化算法(particle swarm optimization, PSO)^[17]和差分进化算法(differential evolution, DE)^[18],新型群智能优化算法包括天鹰优化算法(aquila optimizer, AO)^[19]、海鸥优化算法(seagull optimization algorithm, SOA)^[20]、被囊体种群优化算法(tunicate swarm algorithm, TSA)^[21]、鲸鱼优化算法(whale optimization algorithm, WOA)^[5]、麻雀搜索算法(sparrow search algorithm, SSA)^[7]、灰狼优化算法(grey wolf optimization, GWO)^[22]和基本蜣螂优化算法(DBO)。所有算法最大迭代次数 $T=500$,种群数量 $pop=30$,其他参数严格按照原文设置。各算法在函数 $F1\sim F13$ 30维、100维的优化结果如表3、表4所示,在固定维度函数 $F14\sim F20$ 上的优化结果如表5所示。计算时间也是衡量算法的性能指标,表6列出了各算法30次求解20个测试函数的平均耗时,其中函数 $F1\sim F13$ 维度为30。

由表3可以得出, $F1\sim F13$ 在30维度时,对于单峰函数 $F1\sim F7$,在 $F1\sim F4$ 上,MIDBO最优收敛值、平均收敛精度、标准差都为0,遥遥领先于其他几个对比算法,充分体现了MIDBO的局部搜索能力强且稳定性高;在函数 $F5$ 上,PSO、DE、SOA、TSA、WOA、GWO、DBO等算法都陷入了局部最优,而MIDBO的最优收敛值达到了理论最优解0,平均收敛精度和标准差也排名第一;MIDBO在函数 $F6$ 上的最优收敛值也达到了理论最优解0,平均收敛精度仅次于SSA,排名第二;在函数 $F7$ 上,MIDBO的平均收敛精度稍逊于AO,排名第二,但最优收敛值和标准差排名第一。对于多峰函数 $F8\sim F13$,其中函数 $F8$ 最为复杂,具有很多个极小值,只有MIDBO平均收敛精度达到了理论最优解,证明了融入麻雀搜索算法追随者位置更新机制和柯西变异后可以提升DBO的全局搜索能力和局部逃逸能力;对于函数 $F9\sim F11$,AO、SSA、MIDBO都表现良好,三项指标相同;对于函数 $F12$ 、 $F13$,MIDBO最优收敛值分别达到了 $1.57E-32$ 、 $1.35E-32$,寻优精度比搜索性能良好的SSA高十几个数量级,平均收敛精度和标准差仅次于SSA,排名第二。

由表4可以得出,当测试函数 $F1\sim F13$ 维度为100时,MIDBO依旧保持优越的寻优性能,在函数 $F1\sim F6$ 、 $F8\sim F13$ 上3个指标均排名第一,在 $F7$ 上的平均收敛精度稍差于AO,排名第二。在30维度下,SSA在函数 $F6$ 、 $F12$ 、 $F13$ 上的平均收敛精度高于MIDBO,但在100维度下,MIDBO在这3个函数上的平均收敛精度超越SSA,排名第一;说明随着函数维度的增加,SSA的收敛精度会逐渐下降,而MIDBO依旧保持良好的寻优性能,证明了MIDBO的稳定性。

由表5可以得出,对于固定维度多峰值函数 $F14\sim F20$:在函数 $F16$ 上,除PSO外,其余算法性能相当;在函数 $F17$ 上,MIDBO的平均收敛精度仅次于DE;在函数 $F14$ 、 $F15$ 、 $F18\sim F20$ 上,MIDBO的平均收敛精度都排名第一,标准差也是最小的。体现了MIDBO具有良好的稳定性并能很好地平衡算法的开发能力和探索能力。

由表6可知,MIDBO耗时高于其他算法,这是由于融入了麻雀算法追随者机制扰动策略和柯西高斯变异策略后,适应度评估和位置更新次数增加,导致运行时间变长。MIDBO虽然增加了耗时,但寻优性能总体上优于其余对比算法,所增加的耗时也在可接受范围内。

表3 各算法在函数 $F1 \sim F13$ 上的优化结果($dim=30$)Table 3 Optimization results of each algorithm on functions $F1$ to $F13$ ($dim=30$)

函数	指标	PSO	DE	AO	SOA	TSA	WOA	SSA	GWO	DBO	MIDBO
$F1$	Best	1.46E+03	2.40E-04	2.50E-161	7.23E-15	5.75E-24	1.43E-88	3.19E-145	1.91E-29	7.60E-163	0
	Mean	8.57E+03	4.52E-04	2.71E-102	5.40E-12	1.54E-21	1.15E-73	5.90E-59	1.07E-27	4.51E-96	0
	Std	3.02E+03	1.58E-04	1.49E-101	1.28E-11	5.07E-21	5.11E-73	3.22E-58	1.70E-27	2.47E-95	0
$F2$	Best	1.44E+01	1.31E-03	7.65E-80	4.30E-09	6.61E-15	4.34E-59	1.66E-88	1.38E-17	7.20E-85	0
	Mean	2.85E+01	2.14E-03	2.47E-57	2.34E-08	1.22E-13	4.87E-51	4.25E-31	7.62E-17	3.85E-59	0
	Std	8.07E+00	5.83E-04	1.35E-56	2.08E-08	1.81E-13	2.20E-50	1.90E-30	6.66E-17	2.06E-58	0
$F3$	Best	3.49E+03	2.09E+04	2.03E-159	2.10E-07	3.90E-08	3.02E+04	1.91E-269	4.29E-08	1.80E-140	0
	Mean	8.53E+03	3.23E+04	6.71E-104	4.42E-05	9.36E-05	4.81E+04	2.02E-28	2.66E-05	1.64E-73	0
	Std	3.18E+03	4.69E+03	3.68E-103	1.06E-04	1.95E-04	1.15E+04	7.31E-28	7.36E-05	8.98E-73	0
$F4$	Best	3.78E+01	1.04E+01	9.89E-81	1.57E-04	2.01E-02	1.77E-02	5.96E-91	6.76E-08	9.19E-75	0
	Mean	5.06E+01	1.29E+01	2.27E-52	4.92E-03	4.84E-01	5.02E+01	5.87E-29	8.14E-07	8.14E-54	0
	Std	8.07E+00	1.99E+00	1.25E-51	8.47E-03	6.70E-01	2.82E+01	3.03E-28	6.10E-07	3.82E-53	0
$F5$	Best	2.29E+06	7.06E+01	1.19E-05	2.72E+01	2.71E+01	2.71E+01	3.78E-09	2.56E+01	2.53E+01	0
	Mean	1.34E+07	1.53E+02	1.45E-03	2.81E+01	2.84E+01	2.79E+01	6.13E-05	2.70E+01	2.58E+01	1.47E-05
	Std	8.54E+06	4.21E+01	2.83E-03	5.29E-01	5.92E-01	4.82E-01	1.43E-04	7.76E-01	2.01E-01	7.41E-05
$F6$	Best	2.49E+03	2.38E-04	1.44E-07	2.39E+00	2.57E+00	8.02E-02	1.57E-14	4.26E-05	1.31E-05	0
	Mean	9.70E+03	5.19E-04	3.01E-05	3.24E+00	3.81E+00	4.15E-01	1.57E-11	7.00E-01	9.81E-03	2.32E-08
	Std	3.14E+03	2.23E-04	4.31E-05	3.41E-01	6.32E-01	1.94E-01	4.06E-11	4.33E-01	4.97E-02	1.11E-07
$F7$	Best	2.20E+00	2.72E-02	9.03E-06	4.72E-04	3.33E-03	2.28E-04	1.29E-04	5.53E-04	8.90E-05	3.75E-06
	Mean	5.77E+00	5.84E-02	1.43E-04	2.60E-03	9.90E-03	3.39E-03	1.39E-03	1.81E-03	1.30E-03	2.95E-04
	Std	2.12E+00	1.48E-02	2.35E-04	2.33E-03	4.51E-03	4.68E-03	1.01E-03	1.24E-03	1.35E-03	2.17E-04
$F8$	Best	-2.26E+03	-1.11E+04	-4.19E+03	-5.99E+03	-7.15E+03	-1.26E+04	-9.70E+03	-7.30E+03	-1.23E+04	-1.26E+04
	Mean	-1.82E+03	-9.97E+03	-3.39E+03	-4.98E+03	-5.99E+03	-1.02E+04	-8.55E+03	-6.14E+03	-8.57E+03	-1.26E+04
	Std	2.74E+02	6.79E+02	8.78E+02	4.79E+02	5.57E+02	1.79E+03	6.52E+02	6.79E+02	1.82E+03	5.14E+00
$F9$	Best	6.20E+01	7.05E+01	0	1.71E-13	1.10E+02	0	0	5.68E-14	0	0
	Mean	9.78E+01	8.65E+01	0	1.43E+00	1.82E+02	1.89E-15	0	2.04E+00	2.29E+00	0
	Std	1.37E+01	8.17E+00	0	2.64E+00	4.12E+01	1.04E-14	0	3.12E+00	1.11E+01	0
$F10$	Best	1.65E+01	3.15E-03	8.88E-16	2.00E+01	2.67E-12	8.88E-16	8.88E-16	7.19E-14	8.88E-16	8.88E-16
	Mean	1.83E+01	5.35E-03	8.88E-16	2.00E+01	1.84E+00	3.73E-15	8.88E-16	1.05E-13	8.88E-16	8.88E-16
	Std	9.05E-01	1.25E-03	0	1.21E-03	1.54E+00	2.17E-15	0	1.74E-14	0	0
$F11$	Best	4.46E+01	5.65E-04	0	8.05E-14	0	0	0	0	0	0
	Mean	8.16E+01	5.51E-03	0	2.54E-02	1.19E-02	3.85E-03	0	3.63E-03	0	0
	Std	2.63E+01	5.59E-03	0	5.07E-02	2.12E-02	2.11E-02	0	8.48E-03	0	0
$F12$	Best	4.24E+05	8.93E-06	3.52E-08	1.53E-01	5.61E-01	4.52E-03	1.16E-15	6.81E-03	2.82E-07	1.57E-32
	Mean	2.51E+07	5.63E-05	4.85E-06	3.59E-01	7.10E+00	1.87E-02	6.07E-13	4.55E-02	2.37E-04	2.03E-09
	Std	2.55E+07	3.79E-05	7.95E-06	1.99E-01	3.52E+00	9.37E-03	1.65E-12	2.42E-02	1.17E-03	6.65E-09
$F13$	Best	7.48E+06	6.69E-05	5.26E-10	1.54E+00	1.91E+00	6.81E-02	1.42E-14	1.94E-01	1.10E-02	1.35E-32
	Mean	7.08E+07	2.64E-04	2.95E-05	2.05E+00	3.05E+00	4.62E-01	1.25E-11	6.45E-01	7.62E-01	3.57E-08
	Std	5.20E+07	1.25E-04	6.28E-05	1.99E-01	7.54E-01	2.20E-01	2.90E-11	2.06E-01	5.96E-01	1.08E-07

3.4 收敛曲线分析

为了更直观地反映各算法的收敛速度、局部逃逸能力,给出20个测试函数的迭代曲线图和20个测试函数独立运行30次各算法的平均迭代次数($F1 \sim F13$ 维度为30),分别如图3和表7所示。其中函数 $F1$ 、 $F4$ 、 $F9$ 的迭代曲线图中没有MIDBO的收敛曲线是因为MIDBO在第一次迭代就收敛到了理论最优解0。结合表7和图3可知:对于测试函数

$F1 \sim F4$, MIDBO收敛精度最高且收敛到了理论最优解,在函数 $F1$ 、 $F3$ 上迭代次数最少,在函数 $F2$ 、 $F4$ 上的迭代次数仅次于PSO,这是由于PSO过早陷入停滞;AO在 $F1 \sim F4$ 上都陷入了局部最优解且没有逃逸出来,DBO相较于其他算法收敛速度较快,但不能达到理论最优解;由函数 $F5$ 、 $F7$ 、 $F8$ 迭代曲线可知,相比于其他算法,MIDBO能连续几次跳出局部最优解到达更优解,这体现了MIDBO有很好的局部逃

表4 各算法在函数 $F1 \sim F13$ 上的优化结果($dim=100$)Table 4 Optimization results of each algorithm on functions $F1$ to $F13$ ($dim=100$)

函数	指标	PSO	DE	AO	SOA	TSA	WOA	SSA	GWO	DBO	MIDBO
F1	Best	1.92E+05	1.36E+03	4.45E-159	5.91E-07	1.39E-11	1.38E-79	0	1.65E-13	2.01E-178	0
	Mean	2.29E+05	1.79E+03	5.26E-100	2.09E-05	3.99E-10	2.63E-70	5.00E-54	1.29E-12	5.57E-128	0
	Std	1.66E+04	2.58E+02	2.88E-99	1.89E-05	5.16E-10	1.37E-69	1.92E-53	1.32E-12	3.05E-127	0
F2	Best	1.60E+31	2.61E+01	8.89E-81	6.02E-06	3.66E-08	9.71E-56	1.04E-184	1.84E-08	1.53E-94	0
	Mean	3.13E+39	3.10E+01	3.14E-51	6.91E-05	1.93E-07	1.80E-50	4.10E-24	4.33E-08	2.61E-64	0
	Std	9.33E+39	2.82E+00	1.72E-50	4.79E-05	1.41E-07	4.87E-50	2.25E-23	1.48E-08	1.43E-63	0
F3	Best	4.19E+05	3.53E+05	1.15E-153	7.93E-01	2.23E+03	6.26E+05	0	5.49E+01	1.07E-181	0
	Mean	9.35E+05	4.37E+05	3.71E-97	2.06E+02	1.41E+04	1.08E+06	1.08E-22	6.13E+02	3.07E-02	0
	Std	2.60E+05	4.16E+04	2.03E-96	3.75E+02	9.41E+03	2.50E+05	5.67E-22	7.09E+02	1.68E-01	0
F4	Best	8.87E+01	8.40E+01	3.30E-79	3.61E+01	2.30E+01	4.68E+01	4.22E-108	1.15E-01	2.20E-93	0
	Mean	9.37E+01	8.95E+01	5.05E-51	7.02E+01	5.69E+01	8.26E+01	9.52E-28	5.61E-01	2.14E-58	0
	Std	1.99E+00	1.84E+00	2.76E-50	1.73E+01	1.52E+01	1.37E+01	4.79E-27	4.32E-01	1.17E-57	0
F5	Best	7.81E+08	1.28E+06	2.76E-04	9.80E+01	9.70E+01	9.75E+01	1.77E-06	9.61E+01	9.56E+01	0
	Mean	9.52E+08	2.34E+06	7.76E-03	9.87E+01	9.82E+01	9.82E+01	4.63E-03	9.79E+01	9.61E+01	5.87E-04
	Std	7.10E+07	7.41E+05	1.11E-02	2.77E-01	5.42E-01	2.24E-01	8.88E-03	5.90E-01	2.86E-01	2.03E-03
F6	Best	2.02E+05	1.24E+03	5.57E-07	1.69E+01	1.22E+01	2.38E+00	1.78E-10	7.88E+00	2.47E-01	0
	Mean	2.34E+05	1.79E+03	6.35E-04	1.87E+01	1.47E+01	4.08E+00	4.28E-06	9.92E+00	7.02E-01	2.12E-07
	Std	1.44E+04	2.66E+02	1.66E-03	6.71E-01	1.08E+00	1.38E+00	7.98E-06	1.08E+00	2.75E-01	7.60E-07
F7	Best	1.60E+03	2.28E+00	4.81E-07	1.02E-03	2.71E-02	4.28E-05	1.52E-05	2.55E-03	1.08E-04	2.56E-05
	Mean	1.89E+03	3.20E+00	9.48E-05	9.48E-03	5.37E-02	3.31E-03	1.99E-03	7.13E-03	2.25E-03	4.43E-04
	Std	1.45E+02	5.75E-01	8.03E-05	5.96E-03	2.10E-02	3.26E-03	2.43E-03	3.08E-03	1.63E-03	3.96E-04
F8	Best	-9.17E+03	-1.77E+04	-4.18E+04	-1.55E+04	-1.56E+04	-4.19E+04	-2.89E+04	-1.89E+04	-4.06E+04	-4.19E+04
	Mean	-7.17E+03	-1.65E+04	-1.05E+04	-1.03E+04	-1.28E+04	-3.36E+04	-2.64E+04	-1.62E+04	-2.80E+04	-4.19E+04
	Std	9.20E+02	5.87E+02	6.31E+03	1.98E+03	1.11E+03	5.88E+03	1.17E+03	2.29E+03	5.77E+03	5.21E-03
F9	Best	1.44E+03	7.26E+02	0	2.82E-07	7.01E+02	0	0	4.57E-09	0	0
	Mean	1.54E+03	7.91E+02	0	5.81E+00	9.83E+02	0	0	1.04E+01	0	0
	Std	4.61E+01	2.85E+01	0	6.97E+00	1.45E+02	0	0	8.67E+00	0	0
F10	Best	2.05E+01	6.66E+00	8.88E-16	2.00E+01	4.40E-07	8.88E-16	8.88E-16	5.50E-08	8.88E-16	8.88E-16
	Mean	2.07E+01	7.43E+00	8.88E-16	2.00E+01	2.98E-05	4.56E-15	8.88E-16	1.17E-07	8.88E-16	8.88E-16
	Std	8.89E-02	4.50E-01	0	3.65E-04	1.29E-04	3.16E-15	0	3.81E-08	0	0
F11	Best	1.73E+03	1.26E+01	0	8.93E-07	1.03E-11	0	0	1.02E-13	0	0
	Mean	2.07E+03	1.70E+01	0	1.92E-02	4.52E-03	0	0	2.81E-03	0	0
	Std	9.97E+01	2.60E+00	0	3.22E-02	1.19E-02	0	0	7.33E-03	0	0
F12	Best	1.54E+09	5.93E+05	3.58E-08	6.14E-01	5.62E+00	2.56E-02	4.92E-11	1.66E-01	3.80E-03	4.71E-33
	Mean	2.18E+09	1.48E+06	2.04E-06	7.71E-01	1.31E+01	5.30E-02	2.86E-08	2.80E-01	8.17E-03	1.73E-09
	Std	2.81E+08	5.59E+05	4.18E-06	9.65E-02	4.58E+00	2.13E-02	5.66E-08	4.87E-02	3.41E-03	4.98E-09
F13	Best	2.86E+09	2.57E+06	1.05E-09	8.79E+00	9.39E+00	1.54E+00	1.61E-09	5.51E+00	2.84E+00	1.35E-32
	Mean	4.16E+09	5.73E+06	2.95E-05	9.30E+00	1.42E+01	3.12E+00	2.42E-06	6.78E+00	4.99E+00	9.89E-08
	Std	4.22E+08	1.78E+06	4.27E-05	2.76E-01	6.80E+00	9.68E-01	5.28E-06	4.84E-01	1.04E+00	3.85E-07

逸能力;由函数 $F9 \sim F11$ 的迭代曲线可知虽然 DBO、AO、SSA、WOA 等几个对比算法和 MIDBO 的收敛精度一样,但由表 7 可知 MIDBO 在 $F9 \sim F11$ 上的平均迭代次数分别为 13、15、2,收敛速度远远快于其他对比算法;由函数 $F12$ 、 $F13$ 的迭代曲线可知, MIDBO 的收敛精度超过了全局搜索能力很强的 SSA,说明 MIDBO 有卓越的全局搜索能力;对于固定维度多峰值函数 $F14$ 、 $F15$ 、 $F19$ 、 $F20$, MIDBO

在保证收敛精度很高的情况下也具备最快的收敛速度,没有跟其他对比算法一样多次陷入局部最优解,说明 MIDBO 能够很好地平衡算法的探索能力和开发能力。

3.5 算法统计检验结果

仅根据平均值和标准差无法准确衡量一个算法的性能,因此采用 Wilcoxon 秩和检验验证 MIDBO 与对比算法是否存在显著差异^[23];对运算结果进行 Wil-

表5 各算法在函数 F14~F20 上的优化结果(固定维数)

Table 5 Optimization results of each algorithm on functions F14 to F20 (fixed dimensions)

函数	指标	PSO	DE	AO	SOA	TSA	WOA	SSA	GWO	DBO	MIDBO
F14	Best	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	Mean	4.68E+00	1.10E+00	2.54E+00	2.54E+00	7.85E+00	3.78E+00	5.29E+00	5.04E+00	2.21E+00	9.98E-01
	Std	2.97E+00	3.03E-01	2.90E+00	2.92E+00	5.38E+00	3.71E+00	5.25E+00	4.35E+00	2.58E+00	2.37E-10
F15	Best	2.84E-03	5.49E-04	3.76E-04	3.13E-04	3.08E-04	3.09E-04	3.07E-04	3.07E-04	3.07E-04	3.08E-04
	Mean	2.50E-02	7.52E-04	5.44E-04	1.16E-03	8.41E-03	7.16E-04	3.15E-04	5.75E-03	7.75E-04	3.19E-04
	Std	1.61E-02	1.30E-04	1.81E-04	2.47E-04	1.30E-02	5.29E-04	4.38E-05	8.97E-03	3.69E-04	1.05E-05
F16	Best	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Mean	-6.67E-01	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Std	2.08E-01	6.78E-16	3.97E-04	2.77E-06	9.65E-03	1.53E-09	4.88E-16	2.08E-08	6.18E-16	3.49E-05
F17	Best	-2.81E+00	-3.32E+00	-3.29E+00	-3.13E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00
	Mean	-1.49E+00	-3.32E+00	-3.16E+00	-2.98E+00	-3.19E+00	-3.15E+00	-3.28E+00	-3.22E+00	-3.24E+00	-3.26E+00
	Std	5.77E-01	7.10E-04	7.52E-02	2.54E-01	2.89E-01	1.56E-01	5.83E-02	8.48E-02	9.01E-02	7.65E-02
F18	Best	-4.40E+00	-1.02E+01	-1.02E+01	-1.01E+01	-1.01E+01	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01
	Mean	-1.65E+00	-8.91E+00	-1.01E+01	-4.32E+00	-6.50E+00	-8.87E+00	-8.79E+00	-9.31E+00	-6.35E+00	-1.02E+01
	Std	1.06E+00	2.40E+00	9.23E-03	4.57E+00	3.17E+00	2.37E+00	2.29E+00	1.92E+00	2.37E+00	6.59E-03
F19	Best	-3.46E+00	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01
	Mean	-1.78E+00	-1.02E+01	-1.04E+01	-5.70E+00	-7.23E+00	-8.13E+00	-8.63E+00	-9.79E+00	-7.78E+00	-1.04E+01
	Std	7.34E-01	7.67E-01	8.29E-03	4.59E+00	3.31E+00	2.84E+00	2.55E+00	1.89E+00	2.90E+00	4.83E-03
F20	Best	-3.76E+00	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01
	Mean	-1.75E+00	-1.05E+01	-1.05E+01	-6.07E+00	-5.41E+00	-6.09E+00	-9.09E+00	-9.99E+00	-7.93E+00	-1.05E+01
	Std	7.82E-01	1.42E-01	1.26E-02	3.87E+00	3.62E+00	3.39E+00	2.43E+00	2.06E+00	2.87E+00	5.59E-03

表6 不同算法的耗时对比

Table 6 Comparison of time cost of different algorithms

单位:s

函数	PSO	DE	AO	SOA	TSA	WOA	SSA	GWO	DBO	MIDBO
F1	0.064 3	0.272 7	0.109 3	0.118 3	0.118 0	0.060 7	0.144 0	0.109 0	0.112 0	0.331 3
F2	0.055 3	0.256 7	0.098 0	0.085 7	0.111 3	0.044 0	0.130 7	0.091 0	0.101 0	0.304 7
F3	0.092 0	0.575 0	0.189 7	0.247 3	0.251 3	0.245 7	0.372 7	0.283 0	0.304 7	0.866 0
F4	0.045 0	0.285 7	0.080 3	0.077 7	0.092 7	0.039 0	0.119 0	0.124 3	0.085 0	0.232 0
F5	0.061 0	0.231 0	0.113 7	0.095 0	0.110 7	0.055 3	0.156 3	0.112 7	0.102 3	0.293 7
F6	0.041 3	0.218 3	0.079 3	0.078 0	0.091 7	0.063 7	0.119 3	0.081 0	0.079 3	0.226 0
F7	0.055 7	0.239 7	0.107 7	0.103 7	0.119 7	0.104 7	0.168 0	0.121 0	0.114 3	0.420 7
F8	0.055 0	0.216 3	0.098 0	0.089 0	0.107 3	0.056 3	0.146 7	0.128 3	0.102 0	0.311 0
F9	0.045 3	0.196 7	0.081 3	0.084 3	0.104 3	0.044 7	0.127 0	0.147 3	0.085 7	0.243 3
F10	0.047 7	0.208 7	0.086 3	0.088 0	0.102 0	0.044 7	0.131 3	0.089 3	0.089 0	0.254 0
F11	0.060 7	0.224 3	0.107 0	0.097 7	0.109 3	0.052 7	0.157 3	0.101 7	0.102 7	0.294 7
F12	0.124 0	0.317 0	0.220 7	0.235 0	0.198 7	0.149 7	0.360 7	0.186 0	0.301 7	0.615 7
F13	0.190 3	0.324 3	0.221 0	0.182 0	0.198 3	0.175 0	0.308 0	0.187 3	0.195 0	0.620 3
F14	0.495 3	0.467 3	0.588 3	0.273 3	0.278 0	0.397 3	0.564 7	0.263 3	0.342 7	1.290 7
F15	0.062 0	0.175 0	0.082 7	0.036 3	0.036 3	0.046 0	0.117 0	0.034 0	0.079 0	0.211 7
F16	0.060 3	0.169 7	0.074 7	0.026 7	0.026 0	0.027 0	0.109 3	0.027 7	0.071 3	0.213 0
F17	0.064 3	0.212 3	0.096 3	0.047 7	0.046 3	0.034 0	0.143 0	0.043 7	0.086 0	0.237 3
F18	0.065 3	0.210 0	0.115 3	0.053 7	0.049 7	0.057 3	0.139 7	0.044 7	0.092 3	0.258 3
F19	0.066 0	0.201 3	0.144 7	0.051 0	0.062 3	0.055 0	0.141 3	0.055 0	0.098 7	0.261 7
F20	0.079 7	0.208 3	0.201 3	0.062 3	0.080 0	0.048 7	0.170 3	0.065 3	0.106 3	0.343 0

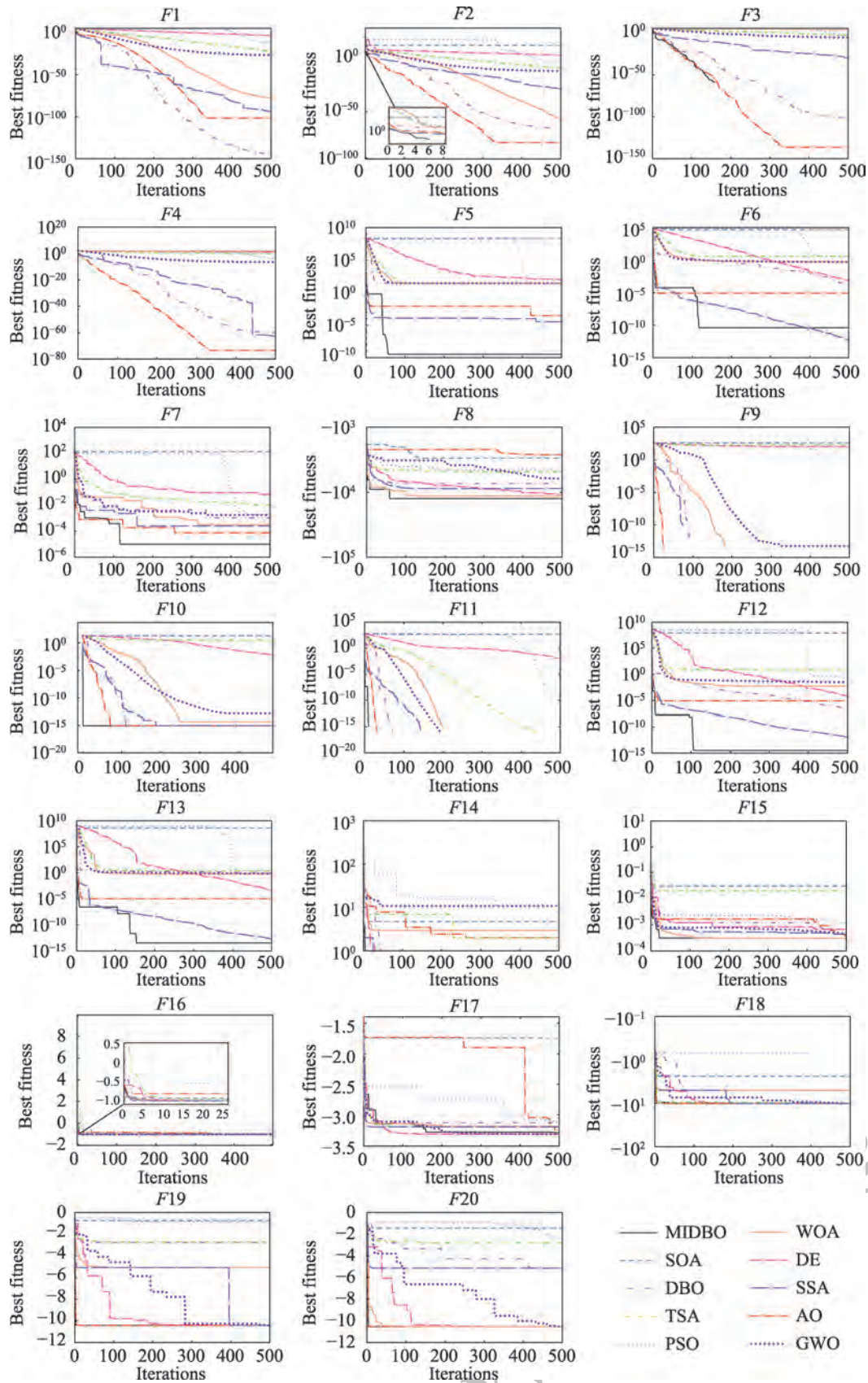


图3 各种算法收敛曲线图

Fig.3 Convergence curves of each algorithm

coxon 秩和检验并计算 p 值, 当 $p < 0.05$ 时, 可认为 MIDBO 与该算法具有显著性差异, 当 $p > 0.05$ 时, 表示两种算法之间差异不明显, NaN 表示两种算法寻

优结果高度接近, 无法进行显著性判断。统计测试函数 $F1 \sim F13$ 在 30 维度下和固定维度函数 $F14 \sim F20$ 的秩和检验结果, 结果如表 8 所示。

表 7 各种算法 30 次独立运行平均迭代次数

Table 7 Average number of iterations of 30 independent runs of each algorithm

函数	PSO	DE	AO	SOA	TSA	WOA	SSA	GWO	DBO	MIDBO
F1	60	498	333	500	500	500	500	500	499	13
F2	7	499	333	500	500	500	500	500	499	23
F3	43	381	333	500	500	500	499	500	499	53
F4	38	488	333	500	495	451	500	500	499	18
F5	15	490	376	500	478	465	496	500	499	143
F6	56	498	125	500	459	465	499	500	500	218
F7	1	450	231	467	431	371	304	375	295	198
F8	198	488	478	381	339	488	498	500	485	232
F9	3	458	58	498	350	190	88	371	124	13
F10	22	498	92	83	469	253	196	373	135	15
F11	379	497	39	500	448	211	84	247	98	2
F12	25	498	283	500	411	462	499	500	499	218
F13	23	498	230	500	402	453	499	500	500	161
F14	26	170	422	482	326	421	169	477	191	278
F15	4	452	467	498	366	464	497	499	486	377
F16	3	166	464	493	275	490	66	490	137	307
F17	1	425	495	499	329	480	143	499	258	465
F18	7	464	60	495	254	463	208	500	326	195
F19	7	465	51	482	309	465	247	499	362	219
F20	6	461	155	500	279	471	273	500	378	246

表 8 各算法 Wilcoxon 秩和检验 p 值

Table 8 Wilcoxon rank sum test p values of each algorithm

函数	PSO	DE	AO	SOA	TSA	WOA	SSA	GWO	DBO
F1	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	4.57E-12	1.21E-12	1.21E-12
F2	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F3	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	4.57E-12	1.21E-12	1.21E-12
F4	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F5	3.02E-11	3.02E-11	2.67E-09	2.98E-11	3.01E-11	2.98E-11	4.08E-05	3.01E-11	3.01E-11
F6	3.00E-11	2.92E-11	6.47E-11	3.01E-11	2.98E-11	2.98E-11	1.17E-03	2.52E-11	3.02E-11
F7	3.02E-11	3.02E-11	1.37E-03	1.21E-10	3.02E-11	8.35E-08	1.11E-06	2.61E-10	4.12E-06
F8	1.77E-11	2.58E-11	2.10E-11	2.10E-11	1.83E-11	1.99E-11	1.99E-11	1.99E-11	2.27E-11
F9	1.21E-12	1.21E-12	NaN	1.21E-12	1.21E-12	NaN	NaN	1.63E-11	NaN
F10	1.21E-12	1.21E-12	NaN	1.21E-12	1.21E-12	1.23E-09	NaN	1.11E-12	3.34E-01
F11	1.21E-12	1.21E-12	NaN	1.21E-12	4.79E-08	NaN	NaN	1.10E-02	NaN
F12	3.02E-11	3.00E-11	6.05E-11	2.98E-11	3.00E-11	2.95E-11	2.13E-05	3.02E-11	3.01E-11
F13	2.98E-11	3.02E-11	1.46E-10	3.01E-11	3.00E-11	3.02E-11	1.58E-01	3.02E-11	3.00E-11
F14	3.01E-11	1.19E-12	1.09E-10	3.32E-11	3.14E-11	1.00E-08	6.31E-02	4.96E-09	6.51E-02
F15	3.02E-11	3.02E-11	8.99E-11	3.02E-11	1.44E-03	1.86E-09	2.39E-08	5.26E-04	2.83E-08
F16	3.02E-11	1.21E-12	2.88E-06	1.20E-08	1.31E-08	3.02E-11	1.25E-11	3.02E-11	6.32E-12
F17	3.02E-11	1.56E-10	7.60E-07	8.35E-08	9.63E-02	1.41E-01	1.28E-04	1.12E-02	4.20E-01
F18	3.02E-11	9.94E-01	5.97E-05	3.69E-11	3.02E-11	1.10E-08	3.75E-01	6.20E-04	5.16E-02
F19	3.02E-11	1.49E-03	8.84E-07	3.34E-11	3.02E-11	1.07E-07	6.00E-05	8.12E-04	1.00E+00
F20	3.02E-11	3.73E-07	2.96E-05	4.08E-11	7.39E-11	9.76E-10	5.50E-08	2.15E-02	2.22E-01

由表8可知,除了在函数 $F9\sim F11$ 上几个算法的寻优结果差异不大外,大部分测试函数上的 p 值大部分都小于0.05,表明MIDBO在总体上与其他几个对比算法有显著性差异。

3.6 CEC2019函数测试

为进一步验证MIDBO具备处理复杂问题的能力,选取CEC2019函数测试集进行实验,函数具体信息如表9所示。其中包含10个单目标测试函数,且具有复杂的空间特征,这些函数的理论最优值均为1。本文选取粒子群优化算法(PSO)^[17]、鲸鱼优化算法(WOA)^[5]、麻雀搜索算法(SSA)^[7]、文献[24]所提的改进 Harris 鹰优化算法(Harris hawks optimization hybridized with multi-verse optimizer, HHMV)、文献[25]所提的改进鲸鱼优化算法(whale optimization algorithm based on multi-population, MEWOA)进行对比,对比结果如表10所示。

由表10可知,MIDBO在CEC01上达到了理论最优解1,标准差为0,两个指标都排名第一;在CEC02、CEC05、CEC08、CEC09、CEC10上,平均值都是最小的,在CEC03、CEC04、CEC06上平均值排名第三。由此可知,MIDBO在求解复杂函数时也具备一定的优势。

表9 CEC2019测试函数

Table 9 CEC2019 benchmark functions

编号	函数名称	维度	范围	最优值
CEC01	Storn's Chebyshev polynomial fitting problem	9	[-8 192,8 192]	1
CEC02	Inverse Hilbert matrix problem	16	[-16 384,16 384]	1
CEC03	Lennard-Jones minimum energy cluster	18	[-4,4]	1
CEC04	Rastrigin's function	10	[-100,100]	1
CEC05	Griewank's function	10	[-100,100]	1
CEC06	Weierstrass function	10	[-100,100]	1
CEC07	Modified Schaffer's function	10	[-100,100]	1
CEC08	Expanded Schaffer's F6 function	10	[-100,100]	1
CEC09	Happy cat function	10	[-100,100]	1
CEC10	Ackley function	10	[-100,100]	1

4 汽车碰撞优化问题求解

为进一步验证MIDBO在实际工程应用中的可靠性,选取汽车侧面碰撞设计问题进行验证。汽车碰撞设计问题需要最小化一个具有11个变量的目标函数,具体变量含义见文献[26]。

将MIDBO与WOA、黑猩猩优化算法(chimp op-

表10 各算法在CEC2019函数上的优化结果

Table 10 Optimization results of each algorithm on CEC2019 benchmark functions

函数	指标	PSO	WOA	SSA	HHMV	MEWOA	MIDBO
CEC01	Mean	8.04E+12	3.35E+06	1.32E+09	5.70E+04	1.00E+00	1.00E+00
	Std	8.64E+12	3.94E+06	3.48E+09	8.82E+03	2.91E-10	0.00E+00
CEC02	Mean	1.23E+04	7.07E+06	1.74E+01	1.74E+01	4.93E+00	4.93E+00
	Std	3.02E+03	2.73E+03	2.08E-02	6.91E-03	1.55E-01	2.15E-01
CEC03	Mean	1.27E+01	4.41E+00	1.27E+01	1.27E+01	4.21E+00	5.40E+00
	Std	0.00E+00	1.82E+00	1.78E-15	1.19E-05	1.28E+00	1.32E+00
CEC04	Mean	1.73E+01	5.04E+01	3.42E+01	6.17E+01	5.78E+01	5.00E+01
	Std	7.38E+01	2.09E+01	1.08E+01	4.48E+01	1.40E+01	1.51E+01
CEC05	Mean	1.48E+00	2.05E+00	1.18E+00	1.25E+00	2.30E+00	1.05E+00
	Std	1.59E-01	4.55E-01	1.20E-01	2.85E-01	6.03E-01	4.23E-02
CEC06	Mean	9.98E+00	8.13E+00	5.49E+00	3.36E+00	7.92E+00	6.40E+00
	Std	9.32E-01	1.83E+00	8.35E-01	1.90E+00	1.27E+00	1.08E+00
CEC07	Mean	2.01E+02	1.30E+03	4.61E+02	4.73E+02	1.31E+03	1.52E+03
	Std	1.80E+02	3.50E+02	1.18E+02	3.81E+01	2.94E+02	2.56E+02
CEC08	Mean	4.95E+00	4.53E+00	5.48E+00	4.79E+00	4.43E+00	4.43E+00
	Std	6.17E-01	3.29E-01	7.11E-01	9.09E-01	2.92E-01	1.82E-01
CEC09	Mean	2.41E+00	1.36E+00	2.38E+00	3.89E+01	1.44E+00	1.36E+00
	Std	3.38E-02	1.78E-01	2.19E-01	1.84E-01	1.56E-01	1.29E-01
CEC10	Mean	2.04E+01	2.12E+01	2.00E+01	2.05E+01	2.09E+01	2.00E+01
	Std	2.02E-01	1.05E-01	1.88E-03	1.24E-01	1.40E-01	2.23E+00

表 11 各算法的最佳求解方案

Table 11 Optimal solution scheme for each algorithm

变量	MIDBO	IDMO	AOA	SCA	GWO	SSA	ChOA	HHO	MPA	WOA
x_1	0.500 0	0.500 0	0.500 0	0.513 7	0.500 2	0.500 0	0.500 0	0.500 0	0.500 0	0.673 2
x_2	1.143 7	1.231 9	1.234 5	1.252 1	1.245 9	1.253 7	1.114 7	1.334 2	1.176 7	1.319 5
x_3	0.500 0	0.500 0	0.846 8	0.500 0	0.500 1	0.500 0	0.673 6	0.500 0	0.500 0	0.500 0
x_4	1.299 6	1.204 9	1.028 3	1.217 9	1.181 0	1.171 5	0.836 6	1.140 7	1.235 3	1.190 2
x_5	0.500 0	0.500 0	0.549 3	0.559 1	0.518 2	0.673 6	0.500 0	0.548 3	0.500 0	0.500 0
x_6	1.500 0	1.351 7	0.686 2	1.500 0	1.123 4	0.823 1	1.500 0	0.500 0	1.500 0	0.500 0
x_7	0.549 8	0.512 2	0.500 0	0.536 3	0.501 6	0.500 0	0.500 0	0.500 0	0.500 0	0.500 0
x_8	0.304 6	0.345 0	0.342 1	0.304 4	0.345 0	0.325 5	0	0.200 0	0.877 8	0
x_9	0.308 4	0.345 0	0.345 0	0.199 0	0.314 9	0.235 5	0	0.305 2	0.666 8	0
x_{10}	-19.652 7	0.989 9	0.029 5	0.893 2	3.312 2	2.727 3	-29.723 3	1.772 3	-8.753 3	6.694 2
x_{11}	2.638 0	-0.254 1	-0.000 2	-0.318 8	3.137 4	3.931 1	-3.081 3	8.205 0	-0.022 2	18.653 0
$\min f(x)$	23.150 9	23.256 9	25.258 2	23.682 0	23.259 9	23.543 9	23.620 9	23.723 3	22.978 1	24.598 0

timization algorithm, ChOA)、哈里斯鹰优化算法(Harris hawk optimization, HHO)、SSA、GWO 正余弦算法(sine cosine algorithm, SCA)^[27]、算术优化算法(arithmetic optimization algorithm, AOA)^[28]、海洋捕食者算法(marine predators algorithm, MPA)^[29]、文献[11]提出的透镜成像反向学习的精英池侏儒猫鼬优化算法(improved dwarf mongoose optimization, IDMO)的优化结果进行对比,其中 IDMBO、AOA、SCA、GWO、SSA 的数据来源于文献[11], ChOA、HHO、MPA、WOA 数据来源于文献[26]。

MIDBO 与对比算法求解汽车碰撞优化问题的实验结果如表 11 所示。由表 11 可知, MIDBO 得到的最小质量为 23.150 9, 小于其余的对比算法, 验证了 MIDBO 算法在实际工程问题上具有良好的实用性。

(1) 目标函数

$$\min f(X) = 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7$$

(2) 变量范围

$$0.5 \leq x_1 \sim x_7 \leq 1.5; x_8, x_9 \in (0.192, 0.345); -30 \leq x_{10} \sim x_{11} \leq 30$$

(3) 约束条件

$$g_1 = 1.16 - 0.371 7x_2x_4 - 0.009 31x_2x_{10} - 0.484x_3x_9 + 0.013 43x_6x_{10} \leq 1$$

$$g_2 = 0.261 - 0.015 9x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.014 4x_3x_5 + 0.000 875 7x_5x_{10} + 0.080 405x_6x_9 + 0.001 39x_8x_{11} + 0.000 015 75x_{10}x_{11} \leq 0.32$$

$$g_3 = 0.214 + 0.008 17x_5 - 0.131x_1x_8 - 0.070 4x_1x_9 + 0.030 99x_2x_6 - 0.018x_2x_7 + 0.020 8x_3x_8 + 0.121x_3x_9 - 0.003 64x_5x_6 + 0.000 771 5x_5x_{10} - 0.000 535 4x_6x_{10} + 0.001 21x_8x_{11} \leq 0.32$$

$$g_4 = 0.074 - 0.061x_2 - 0.163x_3x_8 + 0.001 232x_3x_{10} - 0.166x_7x_9 + 0.227x_7^2 \leq 0.32$$

$$g_5 = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.020 7x_5x_{10} + 6.63x_6x_9 - 7.7x_7x_8 + 0.32x_9x_{10} \leq 32$$

$$g_6 = 33.86 + 2.95x_3 + 0.179 2x_{10} - 5.05x_1x_2 - 11x_2x_8 - 0.021 5x_5x_{10} - 9.98x_7x_8 + 22x_8x_9 \leq 32$$

$$g_7 = 46.36 - 9.9x_2 - 12.9x_1x_8 + 0.110 7x_3x_{10} \leq 32$$

$$g_8 = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.012 2x_4x_{10} + 0.009 325x_6x_{10} + 0.000 191x_{11}^2 \leq 4$$

$$g_9 = 10.58 - 0.647x_1x_2 - 1.95x_2x_8 + 0.020 54x_3x_{10} - 0.019 8x_4x_{10} + 0.028x_6x_{10} \leq 9.9$$

$$g_{10} = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.043 2x_9x_{10} - 0.055 6x_9x_{11} + 0.000 786x_{11}^2 \leq 15.7$$

5 结束语

本文为提高蜚螂优化算法的优化性能, 提出一种多策略改进的蜚螂优化算法(MIDBO), 使用标准测试函数验证了改进策略的有效性, 将 MIDBO 与新型群智能优化算法在 20 个测试函数上的优化结果进行对比, 并对结果采用 Wilcoxon 秩和检验, 结果表明 MIDBO 比原始 DBO 和对比算法具有更高的收敛精度、更快的收敛速度和更强的鲁棒性; 再将 MIDBO 应用于汽车碰撞优化问题实例, 验证了 MIDBO 在实际工程方面的适用性和可靠性; 在后续的工作中, 将继续提高算法的性能, 提高算法的效率, 减少算法耗时, 并考虑把 MIDBO 运用在其他实际问题中。

参考文献:

[1] CRUMPACKER J B, ROBBINS M J, JENKINS P R. An ap-

- proximate dynamic programming approach for solving an air combat maneuvering problem[J]. *Expert Systems with Applications*, 2022, 203: 117448.
- [2] JABER A, LAFON P, YOUNES R. A branch-and-bound algorithm based on NSGAI for multi-objective mixed integer nonlinear optimization problems[J]. *Engineering Optimization*, 2022, 54(6): 1004-1022.
- [3] ALAAS Z, ELTAYEB G E A, ALDHAIFALLAH M, et al. A new MPPT design using PV-BES system using modified sparrow search algorithm based ANFIS under partially shaded conditions[J]. *Neural Computing and Applications*, 2023, 35(19): 14109-14128.
- [4] SHIKAI S, YUANJIE Z. Constrained trajectory planning for unmanned aerial vehicles using asymptotic optimization approach[J]. *Transactions of the Institute of Measurement and Control*, 2023, 45(13): 2421-2436.
- [5] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [6] XUE J K, SHEN B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization[J]. *The Journal of Supercomputing*, 2022, 79(7): 7305-7336.
- [7] XUE J K, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. *Systems Science & Control Engineering*, 2020, 8(1): 22-34.
- [8] LAITH A, ABD M E, PUTRA S, et al. Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer[J]. *Expert Systems with Applications*, 2022, 191: 116158.
- [9] KHAYYAT M M. Improved bacterial foraging optimization with deep learning based anomaly detection in smart cities [J]. *Alexandria Engineering Journal*, 2023, 75: 407-417.
- [10] XIONG Y, ZOU Z M, CHENG J C. Cuckoo search algorithm based on cloud model and its application[J]. *Scientific Reports*, 2023, 13(1): 10098.
- [11] 贾鹤鸣, 陈丽珍, 力尚龙, 等. 透镜成像反向学习的精英池侏儒猫鼬优化算法[J]. *计算机工程与应用*, 2023, 59(24): 131-139.
- JIA H M, CHEN L Z, LI S L, et al. Optimization algorithm of elite pool dwarf mongoose based on lens imaging reverse learning[J]. *Computer Engineering and Applications*, 2023, 59(24): 131-139.
- [12] LIU Z C, BAI Y S, JIA X S. Multi-strategy improved sparrow search algorithm[J]. *Journal of Physics: Conference Series*, 2023(1).
- [13] 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.
- FU H, LIU H. Improved sparrow search algorithm with multi-strategy integration and its application[J]. *Control and Decision*, 2022, 37(1): 87-96.
- [14] 潘劲成, 李少波, 周鹏, 等. 改进正弦算法引导的蜣螂优化算法[J]. *计算机工程与应用*, 2023, 59(22): 92-110.
- PAN J C, LI S B, ZHOU P, et al. Dung beetle optimization algorithm guided by improved sine algorithm[J]. *Computer Engineering and Applications*, 2023, 59(22): 92-110.
- [15] DUAN J H, GONG Y P, LUO J, et al. Air-quality prediction based on the ARIMA-CNN-LSTM combination model optimized by dung beetle optimizer[J]. *Scientific Reports*, 2023, 13(1): 12127.
- [16] SHEN Q, ZHANG D, XIE M, et al. Multi-strategy enhanced dung beetle optimizer and its application in three-dimensional UAV path planning[J]. *Symmetry*, 2023, 15(7): 1432.
- [17] KENNEDY J, EBERHART R. Particle swarm optimization [C]//*Proceedings of the 1995 International Conference on Neural Networks*, Perth, Nov 27-Dec 1, 1995. Piscataway: IEEE, 1995: 1942-1948.
- [18] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces [J]. *Journal of Global Optimization*, 1997, 11(4): 341-359.
- [19] ABUALIGAH L, YOUSRI D, ABD E M, et al. Aquila optimizer: a novel metaheuristic optimization algorithm[J]. *Computers & Industrial Engineering*, 2021, 157: 107250.
- [20] DHIMAN G, KUMAR V. Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems[J]. *Knowledge-Based Systems*, 2019, 165: 169-196.
- [21] KAUR S, AWASTHI L K, SANGAL A L, et al. Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization[J]. *Engineering Applications of Artificial Intelligence*, 2020, 90: 103541.
- [22] MIRJALILI S, MIRJALILI M S, LEWIS A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46-61.
- [23] 柴岩, 任生. 多策略协同优化的改进HHO算法[J]. *计算机应用研究*, 2022, 39(12): 3658-3666.
- CHAI Y, REN S. Improved HHO algorithm based on multi-strategy cooperative optimization[J]. *Application Research of Computers*, 2022, 39(12): 3658-3666.
- [24] LAITH A, ALI D, DAVOR S, et al. Boosted Harris hawks gravitational force algorithm for global optimization and industrial engineering problems[J]. *Journal of Intelligent Manufacturing*, 2022, 34(6): 2693-2728.
- [25] YA S, CHEN Z, FARHAD G S, et al. An improved whale optimization algorithm based on multi-population evolution for global optimization and engineering design problems [J]. *Expert Systems with Applications*, 2023, 215: 119269.
- [26] 王逸文, 王维莉, 杨宇鸽, 等. 多策略融合改进的海洋捕食者算法及其工程应用[J/OL]. *计算机集成制造系统* [2023-08-04]. <http://kns.cnki.net/kcms/detail/11.5946.TP.20230515.1111.008.html>.

WANG Y W, WANG W L, YANG Y G, et al. Improved marine predators algorithm with multi-strategy fusion and its engineering applications[J/OL]. Computer Integrated Manufacturing Systems [2023-08-04]. <http://kns.cnki.net/kcms/detail/11.5946.TP.20230515.1111.008.html>.

[27] MIRJALILI S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016, 96: 120-133.

[28] LAITH A, ALI D, SEYEDALI M, et al. The arithmetic optimization algorithm[J]. Computer Methods in Applied Mechanics and Engineering, 2021, 376: 113609.

[29] FARAMARZI A, HEIDARINEJAD M, MIRJALILI S, et al. Marine predators algorithm: a nature-inspired metaheuristic [J]. Expert Systems with Applications, 2020, 152: 113377.



郭琴(1998—),女,江西赣州人,硕士研究生,主要研究方向为智能优化算法、装配线平衡等。
GUO Qin, born in 1998, M.S. candidate. Her research interests include swarm intelligence optimization algorithm, assembly line balancing, etc.



郑巧仙(1978—),女,浙江衢州人,博士,副教授,硕士生导师,主要研究方向为优化问题建模与算法设计。

ZHENG Qiaoxian, born in 1978, Ph.D., associate professor, M.S. supervisor. Her research interests include optimization problem modeling and algorithm design.

征文通知:第二十一届CCF中国信息系统及应用大会(WISA 2024)

由中国计算机学会(CCF)主办、CCF信息系统专业委员会和宁夏大学共同承办的旗舰会议——第二十一届CCF中国信息系统及应用大会(WISA 2024)将于2024年8月2—4日在宁夏银川召开。

大会将围绕“数智时代信息系统”主题,从信息系统全生命周期的角度,针对采集、存储、组织、处理、分析、挖掘和服务等环节,关注重点领域数字化转型与信息系统自主可控,聚焦信息系统体系架构、数据存储管理、大模型与知识图谱、人机融合智能信息系统等关键问题,搭建学术、企业、政府交流平台。会议将邀请信息系统关键行业领域的资深专家做主题报告,分享最新的前沿技术与产业热点。会议期间,还将组织专题论坛、企业前沿论坛、专委沙龙等活动,力求全方位地扩大交流范畴,拓展交流形式,提升交流效果。

大会同时接收中文、英文投稿。为确保所录用论文的质量及水平,程序委员会将对所投稿件组织严格的同行专家评议审稿,所有录用论文均要求参会交流。所录用中文论文将推荐到《计算机研究与发展》《计算机科学与探索》《计算机工程与应用》《小型微型计算机系统》《计算机科学》《计算机工程》《计算机应用》《计算机与数字工程》等中文期刊发表(须再次审稿)。所录用英文论文将由斯普林格出版社(Springer)以LNCS系列出版会议论文集(WISA 2019、WISA 2020、WISA 2021、WISA 2022和WISA 2023论文集全部论文均被EI收录)。

大会热忱欢迎从事信息系统及相关领域的专家、学者、学生和专业技术人员踊跃投稿和与会交流。

一、WISA 2024的征文主题包括(但不限于):

- | | | | |
|------------|-------------|-------------|--------------|
| • 信息系统体系架构 | • 信息系统检索技术 | • 知识图谱与信息系统 | • 信息系统自然语言处理 |
| • 信息系统数据融合 | • 信息系统安全与防护 | • 区块链与信息系统 | • 智慧教育与教学系统 |
| • 信息系统数据挖掘 | • 信息系统可视化技术 | • 机器学习与信息系统 | • 智慧健康与医疗系统 |
| • 信息系统存储管理 | • 信息系统数字化转型 | • 智能决策与信息系统 | |
| • 信息系统查询处理 | • 大模型与信息系统 | • 推荐系统与信息系统 | |

二、投稿信息

(1)会议投稿网址:<https://cmt3.research.microsoft.com/WISA2024>。

(2)格式与要求:中文论文格式采用《计算机工程与应用》投稿模板,论文长度要求不超过7页,<http://cea.ceaj.org/CN/column/column18.shtml>;英文论文格式采用Springer LNCS模板,论文长度要求不超过12页,<https://www.springer.com/cn/computer-science/lncs/conference-proceedings-guidelines>。

(3)论文应是未发表的研究成果(且未同时提交给其他刊物或会议评审),应包括中英文题目、中英文摘要、关键词、正文和参考文献等。

(4)论文投稿与审稿采取双盲评审。

三、重要日期

- | | |
|-------------------------------|---------------------|
| • 摘要提交:2024年4月15日(23:59,北京时间) | • 正式版本提交:2024年6月24日 |
| • 全文提交:2024年4月22日(23:59,北京时间) | • 会议日期:2024年8月2—4日 |
| • 审稿结果:2024年6月17日 | |