

## 侏儒猫鼬优化算法

Jeffrey O. Agushaka<sup>a</sup>——押沙龙 E. Ezugwu<sup>a</sup>, Laith Abualigah<sup>b, c, \*</sup>

<sup>a</sup> 夸祖鲁-纳塔尔大学数学、统计和计算机科学学院, 爱德华国王路, 彼得马里茨堡, 夸祖鲁-纳塔尔 3201, 南非

<sup>b</sup> 阿拉伯安曼大学计算机科学与信息学院, 约旦安曼 11953

2021 年 12 月 17 日收到;已于 2021 年 12 月 31 日收到订正本;2022 年 1 月 28 日在线

### 摘要

本文提出了一种新的元启发式算法——矮子猫鼬优化算法(DMO),用于求解经典和 CEC 2020 基准函数和 12 个连续/离散工程优化问题。DMO 模拟了矮猫鼬的觅食行为。猫鼬捕捉猎物(进食)的限制性模式极大地影响了它们的社会行为和生态适应,以弥补有效的家庭营养。猫鼬的补偿行为适应包括猎物大小、空间利用、群体大小和食物供应。该算法利用了矮猫鼬的三个社会群体,即 alpha 群体、保姆群体和侦察兵群体。整个家庭以觅食为一个单位,雌性首领开始觅食,决定觅食路径、覆盖距离和睡眠的土丘。一定数量的猫鼬(通常是雄性和雌性的混合体)充当保姆。它们和幼龙待在一起,直到晌午或傍晚狼群返回。第一名保姆会被交换,与群体一起觅食(剥削阶段)。侏儒猫鼬不为它们的幼崽筑巢;它们把它们从一个沉睡的土堆移到另一个土堆,不再返回之前的觅食地点。侏儒猫鼬在足够大的领地内(探索阶段)采用了一种半游牧的生活方式。游牧行为防止了对某一特定地区的过度开发。它还确保了对整个领土的探索,因为以前没有访问过的沉睡的土丘被归还。将所提出的 DMO 算法与其他 7 种算法的性能进行了比较,从不同的性能指标和统计数据方面展示了该算法的有效性。在大多数情况下,DMO 获得的近似最优解优于当前最先进算法获得的最佳解。DMO 的 Matlab 代码可以在 <https://www.mathworks.com/matlabcentral/fileexchange/105125-dwarf-mongoose-optimization-algorithm> 上找到。

©2022 爱思唯尔 B.V.版权所有。

关键词:矮子猫鼬优化算法;元启发式;自然启发算法;全局优化;工程设计问题

### 1.简介

自然是许多元启发式算法的灵感来源。研究人员已经成功地利用自然现象开发了元启发式算法 [1]。例如,送

电子邮件地址 :208088307@stu.ukzn.ac.za (J.O. Agushaka) , Ezugwu@ukzn.ac.za (A.E. Ezugwu) , aligah.2020@gmail.com (L. Abualigah)

粒子群优化(PSO)背后的灵感一般来说,自然启发的元启发式算法已经成功地解决了多个领域,包括旅行商问题[4]、最优控制[5]、医学图像处理[6]等[7,8]。大自然启发的算法的成功源于模仿大自然最好的一面。

优化出现在我们生活的方方面面,从医药、工程、制造等等。求解最优化问题的两种著名方法是数学方法和元启发式方法。数学方法是梯度相关的,对初始起点[9]很敏感。这一缺陷以及全局优化问题的性质和复杂性已经引发了基于自然的算法被提出的速度的激增。他们对以隐喻为基础的范式的依赖招致了批评。然而,已经有很多成功的故事,特别是在 CEC 2020[12]和实际工程场景[13]等基准中找到复杂优化问题的解决方案。

不同的元启发式算法在解决问题时表现不同,这取决于它们所模拟的自然现象。没有一种算法能最优地解决所有问题,因此需要开发一种新的高性能算法来解决特定类型的问题。基于非免费午餐理论,新的元启发式算法不断被开发出来,以寻找特定复杂和大规模优化问题的更好的最优解。大多数研究者专注于开发更好地搜索问题空间的算法,并平衡对提出的算法[14]的探索 and 开发。在过去的十年里,受自然启发的优化算法的数量呈指数级增长。灵感来源于自然,新颖,优化能力强。

虽然许多算法受到自然界觅食和社会行为结构的启发[15-18],但侏儒猫鼬独特的补偿行为适应尚未被建模来解决优化问题。这些独特的适应性使矮猫鼬脱颖而出,成为最娇小的食肉动物,并使 DMO 在这项研究中使用的一些最先进的算法中具有竞争优势。因此,在本研究中,我们提出了一种新的元启发式算法,该算法利用矮猫鼬的补偿行为适应来模拟它们的觅食行为。我们为优化过程建模了三种社会结构转变:阿尔法组、保姆组和童军组。优化从阿尔法群出发(探索空间)觅食开始,把保姆和幼崽留在巢中。

一旦找到了觅食地点, alpha 群就会进食到中午,这是根据保姆交换标准模拟出来的,然后它们会回来交换保姆。一旦保姆被交换,它们就不会回到之前的觅食地点,以避免过度放牧。侦察兵会找到一个新的觅食地点,并通知雌性首领,带着家族来到新的地点。保姆交换开始了一个新的探索阶段,接着是密集的剥削,直到晚上,当氏族返回一个新的睡觉的巢穴。该算法用于求解基准测试函数和工程领域的 12 个不同优化问题。将 DMO 方法得到的结果与其他 7(7)个优化器的 5 个性能指标进行比较,在大多数情况下,DMO 方法的求解结果优于其他方法。

本文提出的 DMO 在理论上比文献中的一些算法更能找到不同优化问题的全局最优解,因为它具有以下独特的属性:

- DMO 随机创建并改进一组给定优化问题的候选解决方案,依赖于 DMO 的探索 and 开发能力,这模仿了侏儒猫鼬的半游牧行为和补偿适应。
- 问题搜索空间的不同区域是探索矮子猫鼬从一个食物来源或睡觉的土堆到另一个
- 搜索空间的有希望的区域被开发,因为 DMO 是在矮子猫鼬无法捕捉大型猎物供家庭喂养,但单独寻找足够的食物来满足个体后建模的。
- DMO 只有一个可调参数。

本文的其余部分组织如下:第 2 节给出了不同提出的自然启发算法的简要背景。在第 3 节中,提出了侏儒猫鼬优化算法(DMO)。第 4 节介绍了实验设置、结果和详细的讨论。最后,在第 5 节给出了结论和未来的工作。

## 2.背景

在过去的十年里,新提出的基于自然的元启发式算法的速度有了巨大的增长。不同的自然现象激发了许多最先进的算法,包括生物,

表 1  
一些自然启发的元启发式算法(2019-2021 年)。

算法	灵感	参考文献
群教优化算法	小组教学机制	烤肉 烤肉 烤肉 烤肉 烤肉 烤肉 烤肉
黑寡妇优化算法	黑寡妇蜘蛛独特的交配行为。	
混沌游戏优化	混沌理论的一些原理	
青少年身份搜索算法(AISA)	青少年身份发展/寻找的过程	
原子轨道搜索	量子力学的基本原理	
的行为启发了一个新的元启发式优化器	海蜇在海洋中的行为	
海洋中的水母		
量子海豚群算法	海豚群算法	
算法优化算法	算术运算符	
高级算法优化算法	高级运算符	
埃博拉优化搜索算法(EOSA)	埃博拉病毒	烤肉 烤肉 烤肉 烤肉 烤肉 烤肉 烤肉
黄金比例优化法(GROM)	生长在大自然的利用中	
	斐波那契数列的黄金比例	
秃鹰搜索优化算法	秃鹰	
黑洞力学优化	黑洞力学	
Capuchin 搜索算法	卷尾猴	
Tiki-taka 算法	足球踢球风格	
合作搜索算法	现代企业中的团队合作行为	
Aquila 优化器	天鹰鸟	
旗鱼优化器	旗鱼群狩猎	
社交网络搜索	社交网络用户的努力获得更多	烤肉 烤肉
	受欢迎	

物理基础，化学基础，等等。本文提出的算法已被用于解决不同的优化问题，并取得了不同程度的成功[19-27]。同样重要的是，需要注意的是，这些算法中有许多并没有在所有复杂的优化问题上进行测试，也没有应用于处理所有现实世界的优化问题。

这些算法的成功引起了研究人员的极大兴趣，并努力提高其性能。一种先进的离散萤火虫算法用于调度不相关的并行机器与序列相关的设置时间[28]。使用自然启发的元启发式优化卷积神经网络模型[29]实现了乳腺癌图像中异常的特征。在[30]中对求解二次分配问题的元启发式方法进行了性能研究。将共生生物搜索算法与生态系统服务相结合，用于血库系统[31]的动态血液分配。

表 1 显示了过去两年(2019-2021 年)新提出的元启发式算法的一些例子。不幸的是，我们看到一些灵感似乎并不是来自自然或自然发生的现象或系统。例如，Tiki-taka 算法的灵感来自足球，这不是一种自然现象。杂交元启发式算法是研究人员提出新的或改进算法的另一种方式。其中包括萤火虫算法与混沌理论[32]、基于和谐搜索的蚁群策略(HPSACO)[33]、结合岛屿布谷鸟搜索的粒子群优化器(iCSPM)[34]、自装配与粒子群优化(SAPSO)[11]、模糊控制器与黏土堆算法(SMAF)[27]。

### 3.侏儒猫优化算法(DMO)

给出了 DMO 的总体框架，并给出了优化过程。

#### 3.1。灵感

侏儒猫鼬，Helogale，在白蚁丘，岩石和中空的树(用于隐藏)的地区被发现，特别是在非洲的半沙漠和大草原灌木丛地区。矮猫鼬是非洲最小的食肉动物，平均体长 47 厘米，成年体重约 400 克。他们通常生活在

在一个母系社会的家庭群体中,由一对阿尔法伴侣领导终生结合[53,54]。在猫鼬家族中,在每个年龄组中,雌性猫鼬的地位高于雄性,年幼的猫鼬的地位高于年长的兄弟姐妹。在这些群体中,劳动分工和利他主义是哺乳动物中最高的记录。根据年龄和性别,不同的猫鼬扮演着守卫、保姆、攻击捕食者和攻击同种入侵者的角色。关于角色、各种等级顺序和个体间关系的详细分析已经发表[56,57]。

矮猫鼬有自己的领地,通常用它们的颊腺和肛门腺在它们的领地上标记垂直或水平的物体。那些斑纹熟悉的气味让矮人更加自信,不那么焦虑了。所有的家庭成员都对领地的标记有贡献,而这些贡献的频率取决于它们在群体中的优先级顺序。在一项对在塔鲁沙漠发现的矮猫鼬的研究中,据报道,矮猫鼬的领地由许多白蚁丘、岩石和空心树组成。一个家庭花了 21.8 天的时间才走完标记的领地。研究结果表明:家系大小影响植被盖度、取食方式和睡丘利用;然而,标记的边界从未改变。矮猫鼬保持着他们能保卫的最小的经济区域,并为群体提供足够的资源。

侏儒猫鼬已经发展了特定的行为和适应与领地和它与捕食回避的关系。它们没有致命的咬痕;相反,它们用猎物的眼睛寻找方向,进行颅骨粉碎式的咬咬。这种捕杀方式限制了它们猎物的大小,而且没有观察到大型猎物的合作捕杀。猎物捕获的限制模式显著影响猫鼬的社会行为和生态适应,以补偿有效的家庭营养[58]。猫鼬的两种中央补偿行为适应如下:

### 1. 猎物大小,空间利用率和群体大小

矮猫鼬限制性的捕食模式意味着只有很小的猎物被杀死,这些猎物除了喂养幼崽外,体型不够大,无法分享猎物。矮猫鼬的典型猎物主要是节肢动物,有时也会捕获小型哺乳动物、壁虎蜥蜴和小型鸟类。矮猫鼬的猎物范围是一个分散的食物来源,是不可预测的,需要广泛的搜索单个猫鼬来获得一顿饱饭。侏儒猫鼬采用了一种半游牧的生活方式,这保证了它们的家庭能够走相对较长的距离来觅食,而且很少回到以前去过的睡觉的地方。这种空间利用模式确保了没有区域被过度捕猎,减少了猎物的消耗,甚至保证了整个领地的被捕食。

群体作为一个单位觅食,通过发声或短的鼻“窥视”在 2kHz 维持凝聚力。群体每天的行程取决于群体的大小、幼仔的出现以及躲避捕食者所造成的觅食中断。雌性首领开始觅食,决定觅食路径,所覆盖的距离,以及沉睡的土丘。

### 2. 食物供应

在矮猫鼬中,没有发现哺乳的母猫鼬或幼猫鼬的捕食行为。这影响了猫鼬的社会组织,特别是养育行为,导致了以下所述的补偿行为适应。

#### • 异地育儿或临时保姆

一定数量的猫鼬(通常是雄性和雌性的混合体)充当保姆。它们和小企鹅待在一起,直到中午企鹅群回来,保姆们就会交换第一个保姆,和企鹅群一起觅食。侏儒猫鼬不为它们的幼崽筑巢;它们把它们从一个沉睡的土堆移到另一个。当幼崽开始陪伴队伍时,它们完全发育不全,皮毛稀疏,不能跑长距离。这影响了平均每天的觅食,从而限制了群体的活动。

综上所述,矮猫鼬无法捕捉到能够为整个群体提供食物的大型猎物。由于没有致命的咬伤和集体狩猎,侏儒猫鼬采用了一种社会结构,使得每一个成员都能自力更生,而且总是从一个地方迁移到另一个地方。侏儒猫鼬在一块足够大的领地上采用了一种半游牧的生活方式。游牧的行为防止了对某一特定区域的过度开发,同时也确保了对整个区域的探索,因为之前没有人访问过沉睡的土丘被归还。

3.2。人口初始化

DMO 优化从初始化猫鼬(X)的候选种群开始，如式(1)所示。种群在给定问题的上界(UB)和下界(LB)之间随机生成。

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d-1} & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d-1} & x_{2,d} \\ \vdots & \vdots & x_{i,j} & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d-1} & x_{n,d} \end{bmatrix} \tag{1}$$

其中 X 是当前候选总体的集合，它们是用式(2)随机生成的， $x_{i,j}$  表示第 i 个总体的第 JTH 维的位置，n 表示总体大小，d 为问题的维数。

$$xi, j = \text{unifrnd}(\text{VarMin}, \text{VarMax}, \text{VarSize}) \tag{2}$$

其中 unifrnd 为均匀分布随机数，VarMin 和 VarMax 分别为问题的下界和上界。VarSize 是决策变量或问题维度的大小。每次迭代中的最佳解就是到目前为止获得的最佳解。

3.3。DMO 模型

所提出的 DMO 算法模拟了矮猫鼬的补偿行为适应。补偿行为适应包括限制猎物的大小，社会组织(保姆)，半游牧生活，等等。为了实现我们的模型，我们将侏儒猫鼬的社会结构分层为 alpha 群体、侦察兵和保姆。每一个群体都有一种补偿性的行为适应，这导致了在一个足够大的区域内维持一种半游牧的生活方式。侏儒猫鼬以集体觅食和侦察而闻名;在我们的实现中，寻找新土堆和觅食是同时进行的，并且是由同一组猫鼬完成的。当阿尔法小组开始觅食时，他们也会寻找一个新的土堆，一旦保姆交换标准得到满足，他们就会访问这个土堆。我们通过计算每次迭代的平均睡眠丘值来模拟这一点;根据睡眠丘的平均值，侏儒猫鼬种群的下一步决定如式(7)所示。这种游牧行为防止了对特定区域的过度开发，也确保了对整个领土的探索，因为之前没有访问过的睡眠丘被归还。本文提出的 DMO 算法的优化过程分为三个阶段，如图 1 所示。

3.3.1。群

一旦种群初始化，就会计算每个解的适应度。每个群体适应度的概率值由公式(3)计算，并根据该概率选择 alpha female (α)。

$$\alpha = \frac{fit_i}{\sum_{i=1}^n fit_i} \tag{3}$$

alpha 群中的猫鼬数量对应于 n - bs。b 是保姆的数量。雌性首领的鸣叫使整个家族保持在一条道路上，这种鸣叫用窥视来表示。

最初的睡丘设∅，每只猫鼬都睡在里面。为了产生一个候选的食物位置，DMO 使用式(4)中的表达式。

$$X_{i+1} = X_i + phi * peep \tag{4}$$

其中为均匀分布随机数[-1,1]，每次迭代后，沉睡丘如式(5)所示。

$$sm_i = \frac{fit_{i+1} - fit_i}{\max\{|fit_{i+1}, fit_i|\}} \tag{5}$$

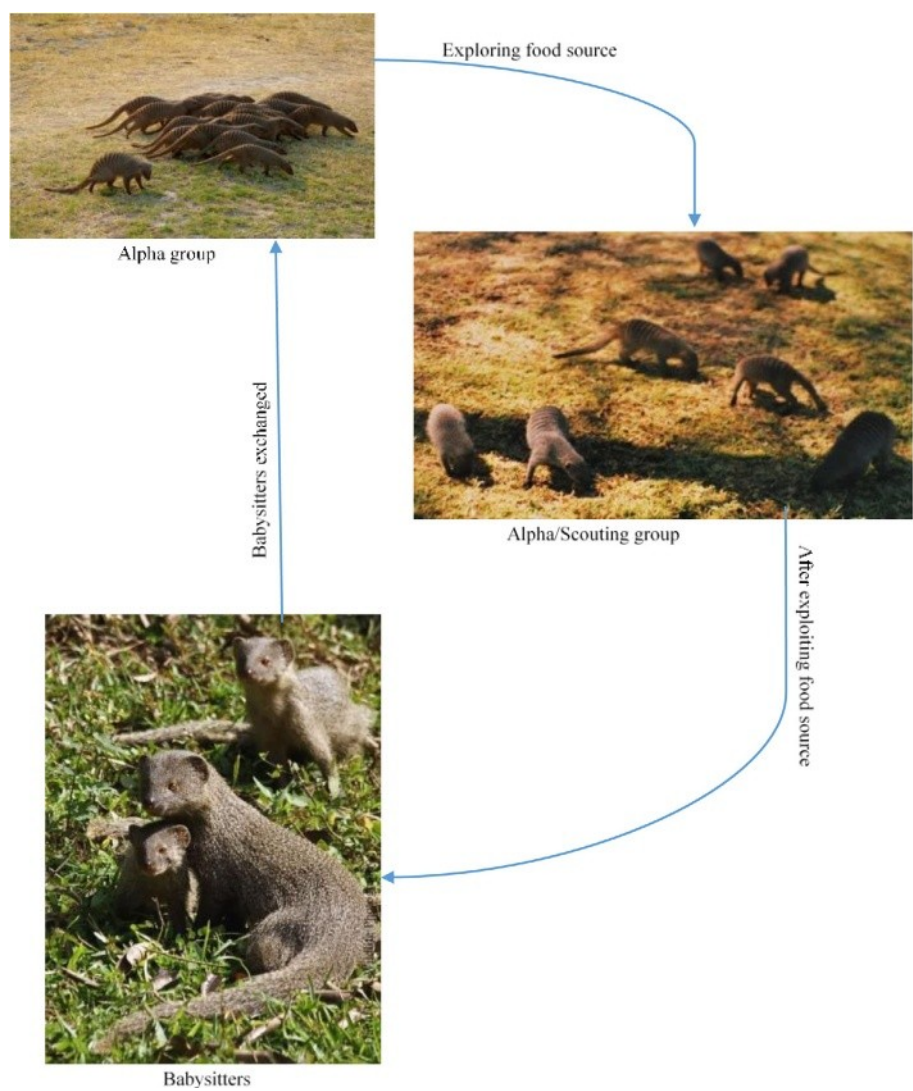


图 1。提出的 DMO 的优化程序。

沉睡土丘的平均值由式(6)给出。

$$\varphi = \frac{\sum_{i=1}^n}{n} \tag{6}$$

该算法进入侦察阶段，一旦保姆交换标准得到满足，就会评估下一个食物来源或睡觉的土堆。

3.3.2。童军团体

侦察兵寻找下一个沉睡的土堆，因为猫鼬已经知道它们不会再回到前一个沉睡的土堆，这就保证了探险。在我们的模型中，侦察与觅食是同时进行的，如[40]中所观察到的。这个动作被建模为寻找一个新的休眠丘的整体成功或失败的评估。换句话说，这个动作取决于猫鼬的整体表现。这样做的理由是，如果这一家人找到足够远的地方，他们就会发现一个新的睡觉的土堆。Eq.(7)模拟了侦察猫鼬。

$$X_{i+1} = \begin{cases} X_i - CF * \phi * rand * [X_i - \vec{M}] & \text{if } \varphi_{i+1} > \varphi_i \\ X_i + CF * \phi * rand * [X_i - \vec{M}] & \text{else} \end{cases} \tag{7}$$





其中，

rand 为 [0,1] 之间的随机数， $C F = \left( \frac{iter}{Maxiter} \right)^{\left( 2 - \frac{iter}{Maxiter} \right)}$  1-Maxiter: 控制 iter 的参数  
猫鼬群的集体意志运动随着迭代线性减少。 $\rightarrow$   
 $m = \sum_{i=1}^n x_i$  是决定猫鼬移动到新的休眠丘的矢量。

3.3.3. 保姆们

照顾幼崽的人通常是群体中的从属成员，他们和幼崽在一起，并定期轮换，以便让雌性首领(母亲)领导群体中的其他成员进行日常觅食。它通常在中午和晚上回来给幼崽喂奶。保姆的数量取决于人口规模;它们通过减少基于百分比集的总体大小来影响算法。我们通过减少保姆代表的百分比来模拟这一组。保姆交换参数用于重置之前由家庭成员持有的侦察和食物来源信息。保姆的适应度权重被设为零，这保证了 alpha 组在下一次迭代中的平均权重被降低，这意味着群体移动受到阻碍，强调了剥削。算法的伪代码在算法 1 中给出。

3.4. 计算复杂度

DMO 的计算复杂度通常依赖于两个规则:解的初始化和主要算法函数，包括以下内容:计算适应度函数、选择 alpha female、评估下一个休眠丘、计算运动向量和更新解。设 n 为解的个数，O(n)为解初始化过程的计算复杂度。

解决方案更新过程的计算复杂度为  $O(iter \times d \times \alpha \times sm \times M) + O(C FE)$ ，包括寻找最佳位置、选择最佳女性、评估下一个休眠丘、计算运动向量和更新所有解决方案的解决方案位置。迭代的总次数称为 iter，给定问题的维数称为 d, C FE 是函数求值的代价。

因此，所提出的 DMO 的总计算复杂度为  $O(iter \times d \times \alpha \times sm \times M \times n + C FE \times n)$ 。

直观详细的 DMO 过程如图 2 所示。优化过程从阿尔法群出发觅食开始，把保姆留在巢中。一旦找到觅食地点，它们就会进食到中午，然后回来交换保姆。它们不会回到之前的觅食地点，因为侦察兵会找到一个新的觅食地点，并通知雌性首领是谁领导这个家庭。保姆交换开始了一个新的探索阶段，接着是密集的剥削，直到晚上，当氏族返回一个新的睡觉的巢穴。重要的是，DMO 算法只有一个需要微调的具体参数(保姆的数量)，这是它的优势之一。为了使用 DMO 算法，它只需要确定种群大小和最大的评估或迭代次数

4.结果与讨论

我们使用 19 个经典基准函数[43]、10 个 CEC 2020 测试函数[44]和 12 个不同工程领域的基准问题来评估 DMO 算法的性能。然后，我们将 DMO 的结果与文献中 7 种不同的基于种群的元启发式算法进行比较。所采用的算法有:算术优化算法(AOA)、基于混合收缩系数的算法(PSO)与重力搜索算法(GSA)简称 CPSOGSA、粒子群优化算法(PSO)、蚁群优化算法(ACO)、salp 群算法(SSA)、正弦余弦算法(SCA)、灰狼优化算法(GWO)。所使用的算法列表和各自的控制参数如表 2 所示。我们利用 MATLAB R2020b 实现了算法、经典和 CEC 2020 测试函数以及工程设计问题。实验使用 Windows 10 操作系统，Intel Core i7-7700@3.60 GHz CPU, 16G RAM。每个算法的独立运行次数设置为 30 次。五个(5)绩效指标相对于获得的解决方案的质量，即:Best, Worst, Average, standard deviation (SD)， and Median values，被用来表示 DMO 取得的结果。采用均值、标准差 (或 SD)、Friedman 均值秩和 Wilcoxon 符号秩检验对所得结果进行统计分析检验。

**Algorithm 1****begin***Initialize the algorithm parameters:**[peep]**Initialize the mongoose populations (search agents): n**Initialize the number of babysitters: bs**Set n=n-bs**Set babysitter exchange parameter L***For** iter=1: max\_iter*Calculate the fitness of the mongoose**Set time counter C**Find the alpha based on Equation 3*

$$\alpha = \frac{fit_i}{\sum_{i=1}^n fit_i}$$

*produce a candidate food position using Equation 4*

$$X_{i+1} = X_i + phi * peep$$

*Evaluate new fitness of  $X_{i+1}$* *Evaluate sleeping mound using equation 5*

$$sm_i = \frac{fit_{i+1} - fit_i}{\max\{|fit_{i+1}, fit_i|\}}$$

*Compute the average value of the sleeping mound found using Equation 6.*

$$\varphi = \frac{\sum_{i=1}^n sm_i}{n}$$

*Compute the movement vector using*

$$\vec{M} = \sum_{i=1}^n \frac{X_i \times sm_i}{X_i}$$

*Exchange babysitters if  $C \geq L$ , and set**Initialize bs position (Equation 1) and calculate fitness*

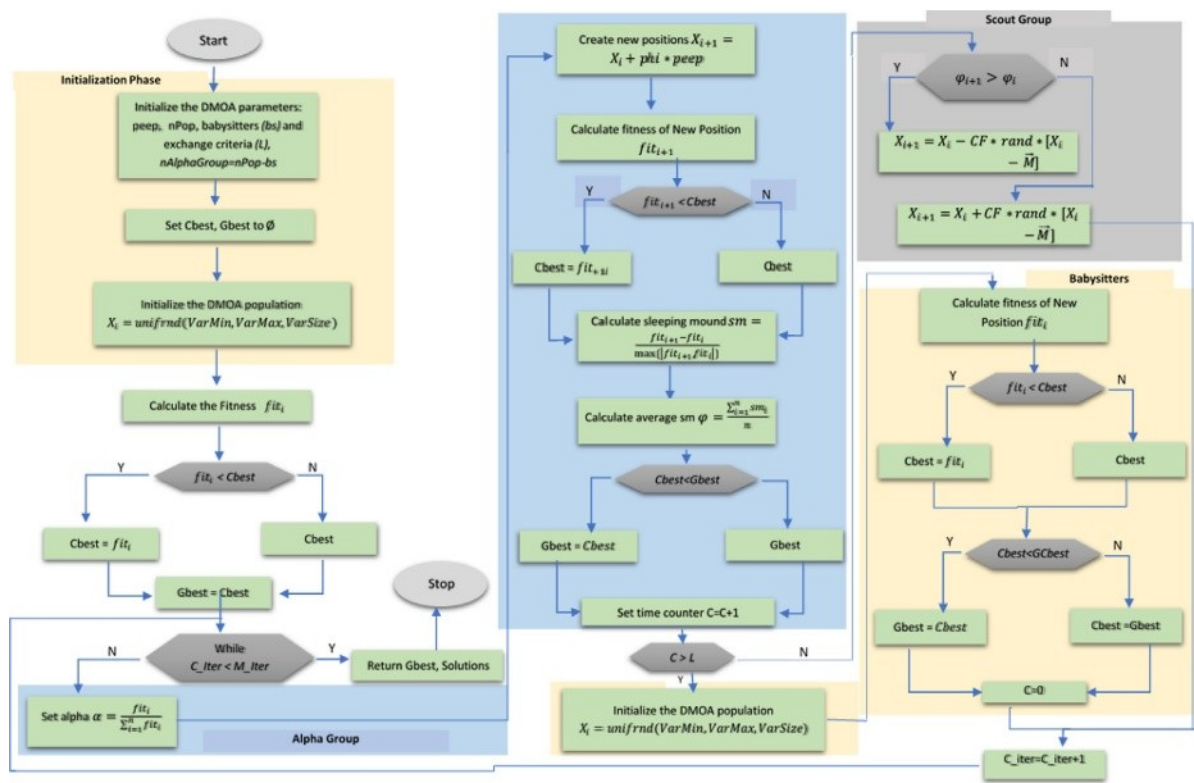
$$fit_i \leq \alpha$$

*Simulate the scout mongoose next position using Equation 7.*

$$X_{i+1} = \begin{cases} X_i - CF * rand * [X_i - \vec{M}] & \text{if } \varphi_{i+1} > \varphi_i & \text{Exploration} \\ X_i + CF * rand * [X_i - \vec{M}] & \text{else} & \text{Exploitation} \end{cases}$$

*Update best solution so far***End For****Return** best solution**End**

初始种群的性质和多样性、种群大小以及迭代或函数评估的次数都会影响基于种群的元启发式算法的性能 [60,61]。我们从初始种群的性质、规模和迭代次数等方面对 DMO 进行了敏感性分析。结果表明，DMO 的性能受种群大小和迭代次数的影响。当种群大小设置为 50，并且最大迭代次数为 1000 时，将返回最佳性能。我们的研究结果还表明，DMO 对 [61] 中提出的不同初始化方案并不特别敏感；但与随机数分布、beta 分布、对数分布、指数分布、瑞利分布和威布尔分布相比，均匀分布的结果最好。



图二。算法流程图。

4.1。元启发式算法

本小节介绍了用于比较提议的 DMO 的最先进的算法。根据解决相关优化问题的性能和在线源代码的可用性 (出于可复制性和可伸缩性的目的)，我们选择了 7 种算法。

算法优化算法

算术优化算法(AOA)的主要灵感是在解决算术问题时使用算术运算符(乘、除、减、加)。AOA 是根据数学中算术运算符的规则来建模的。该算法随机初始化初始解，并将每次迭代的最佳解视为接近最优解[42]。面向对象方法已经成功地应用于许多领域的优化问题的求解。本研究中 AOA 的控制参数如表 2 所示。

粒子群优化算法

粒子群算法(Particle swarm optimization, PSO)是一种流行的群智能算法，在不同的优化领域都取得了巨大的成功。为了解决现实世界中一些复杂问题的过早收敛问题，人们提出了许多变式。本研究将改进的粒子群优化算法与基于岛屿的布谷鸟搜索算法[30]相结合

基于 PSO 和 GSA 的收缩系数

基于约束系数的粒子群优化与引力搜索算法(CPSOGSA)是一种基于约束系数的粒子群优化与引力搜索算法的混合算法。目标是结合粒子群算法和 GSA 的开发和探索能力，以获得更好的结果。该算法已成功地应用于工程[58]的优化问题中。

表 2  
控制参数。

算法	参考文献	参数名称	参数值
协定	烤肉	$\alpha$	5
		$\beta$	0
		$\gamma$	0
		$\mu$	5
微粒	烤肉	$C1$ 、 $C2$ (个人和社会常数) $W_{max}$ (最大惯性权重) $W_{min}$ (最小惯性权重)	2
			0
			9
			0
cpsogsa	[62]	$\langle p1, f \rangle 2$ (控制参数)	2
			0
			5
国	[63]	从 2 线性减小到 0 r1f r2(随机向量)	[0,2]
			[0,1]
sca	[64]	A(常数)	2
sca	[65]	$C2, c3$ (随机数)	[0,1]
acso	[66]	信息素更新常数初始信息素信息素敏感性可见性敏感性	1
			1
			0
			0
			3
			0
			1

灰狼优化

灰狼优化器(GWO)是最近提出的一种元启发式算法，基于灰狼群体在领导和狩猎方面的社会智能。它已被广泛应用于许多领域的各种优化问题。该算法的成功主要归功于其相对于其他群智能方法的显著特点，以及其参数少、初始搜索不要求导信息等特点。在搜索过程[59]中，GWO 声称简单、灵活以及探索和利用之间的正确平衡。

正弦余弦算法

正弦余弦算法(SCA)是最近提出的优化问题的解决方案。SCA 通过创建多个初始随机候选解决方案并使用基于正弦和余弦函数的数学模型向外或向最佳解决方案传播它们来执行优化。该算法引入了多个随机变量和自适应变量，进一步增强了算法的开发过程[60]。

Salp 群算法

SSA 的灵感来自樽海鞘在海洋中航行和觅食时的群集行为。该算法模拟了寻找最佳食物来源的樽海鞘链。链条分为领导者和追随者。领导者发起一个觅食链，追随者跟随它来指导他们的行动。该算法已经在几个数学优化函数和实际应用中进行了成功的测试[61]。本研究使用的 ssparameter 如表 2 所示。

**蚁群优化**

蚂蚁的觅食行为启发了蚁群优化算法的初始设计。蚁群算法是模仿蚂蚁利用信息素标记出其他成员跟随的有利路径(食物来源)。蚁群算法的优化过程遵循类似的机理。蚁群算法在 90 年代初首次提出，目前仍吸引着越来越多的研究人员的注意，并有许多成功的应用[62]。用于比较 DMO 算法性能优势的各自竞争算法的控制参数如表 2 所示。

*4.2。基准测试功能*

本研究中考虑的各算法的结果如表 3 所示。使用 19(19)个单模态、定维多模态和多模态经典基准函数测试 DMO 的开发和探索能力，如表 3 函数列所示。单模态函数、多模态函数和定维多模态函数在确定设计变量或设计变量的数量方面有不同的困难

表 3  
经典基准函数的结果。

命令 功能	全局 优化	价值	d m o	协定	cpsogsa	微粒	ac o	s s a	s c a	国
步骤 F 1	0	最佳	0	6.25	0	0	992	0	3.345	7.28e-0
		最差		83			7.5		1	6 1.00e
		平均	0	6.25	0	0	992	0	4.734	+00
		SD 中		83			7.5		4	3.69e-0
		位数	0	6.25	0	0		0	4.006	1
				83 0					7	3.08e-0
			0	6.25	0	0		0	0.348	1
球 体 F2	0	最佳	0	0	0	0	94	0	1.761e-0	0
		最差					55	1.365e-0	8	
		平均	0	0	0	0	94	8	0.21316	0
		SD 中					55	3.735e-0	0.008479	
		位数	0	0	0	0	94	9	4	0
							55	5.402e-0	0.038747	
			0	0	0	0	0	9 0	8.701e-0	0
和方尺 F3	0	最佳	0	0	0	0	9 039	0	1.232e-0	0
		最差					1.02e +	1.4972	8	
		平均	0	0	0	0	5 96 262	0.1835	0.002360	0
		SD 中					2322.6	3	8	
		位数	0	0	0	0	96 372	0.3348	0.000181	0
								9	0.000501	
			0	0	0	0		0.0493	9	0
四次 F4	0	最佳	0	0	0	0	3.66e	0	0	0
		最差					+07		4.677e-0	
		平均	0	0	0	0	4.42e	0	6	0
		SD 中					+07		3.926e-0	
		位数	0	0	0	0	4.11e	0	7	0
							+07		9.234e-0	
			0	0	0	0	1.63e	0	7	0
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0	0	0
				0.47	0	0	4	0.0005	0.13	2.52e-
				742			0	867	414	05
Mat  yas  F5  Colv  ile	0	最佳	0	0	0	0	0	0	0	0
		最差					3	0	0	0
		平均	0	0	0	0	4	0	0	0
		SD 中					0	0	0	0
		位数	0	0	0	0	3	0		

f 6		最差	0	0.477	7.8628	0	611.1	1.0161	1.627	6.81e
		平均		42	0.26284		611.1	0.05917	2	+00
		SD 中	0	0.477	1.4354	0	1.16e-1	5	0.979	6.84e-
		位数		42 0	3.93e-0		3 611.1	0.21953	86	01
			0	0.477	6	0		0.00134	0.457	1.34e
				42				99	55	+00
			0			0			1.096	2.27e-
									6	03
扎哈 罗夫 F7	0	最佳	0	0	0	0	1.46e	0	0	0
		最差					+08			
		平均	0	0	0	0	1.60e	0	0	0
		SD 中					+08			
		位数	0	0	0	0	1.53e	0	0	0
							+08			
			0	0	0	0	3.37e	0	0	0
							+06			
			0	0	0	0	1.55e	0	0	0
							+08			
Schwefell .2 F8	0	最佳	0	0	124.	0	1.62e	0.144	54.	0
		最差	0.002038		4		+06	18	267	
		平均	9	0	565	0	1.66e	138.9	727	0
		SD 中	6.80e-05		6.9		+06	4	8.9	
		位数	0.000372	0	127	0	1.62e	42.49	261	0
			25 0.0		2.2		+06 23	3	9.1	
				0	104	0	255	32.46	234	0
					1.3		1.62e	1	5.2	
				0	107	0	+06	36.03	170	0
					5.3			9	0.9	

(下一页继续)

解决方案。利用单峰、可分和不可分基准函数(F1-F9)测试了 DMO 在开发中的有效性和准确性。显然，DMO、GOA、AOA、CPSOGSA、PSO、SSA、SCA 和 GWO 表现出最好的性能，因为他们发现除了 F9 之外的单峰、可分离和不可分离基准函数的全局最优;然而，ACO 的性能很差。这也证实了 DMO 有效执行剥削的能力。

采用多模态、可分和不可分基准函数(F10-F19)测试 DMO 有效、准确地进行勘探的能力。所有被考虑的算法都未能找到 F14 和 F17 的全局最优解;然而，DMO 在大多数功能上表现最好，在其他功能上具有竞争力。Friedman 的检验显示，DMO 的平均排名最低，排名第一

表 3(续)。



			0.1125	49				17 0.088 997	0.09697 3	8 1.59e-0 8
谢弗 6 F15	0	最佳	0.484	0	10.90	0	10.39	0.843	3.30	2.44e
		最差	78		4		11.15	53	84	+00
		平均	6.671	0	13.28	0	6	9.807	10.8	5.9285
		SD 中	9		9		10.77	1	26	4.1315
		位数	3.390	0	12.32	0	6	7.995	8.50	1.0824
			4		7		0.185	6	64	4.1357
			1.672	0	0.578	0	93	1.079	1.63	
			8		45		10.77	4	42	
			3.202	0	12.41	0	2	8.053	9.08	
			7		9		7	7	85	
DropWa ve F16	0	最佳	0	0	-	9.38e-0	0.806	0	0	0
		最差			1	1	43	0.999		
		平均	0	0	-	0.9494	0.806	97	0	0
		SD 中			1	0.94375	43	0.918		
		位数	0	0	-	0.00798	0.806	15	0	0
					1	18	43 0	0.115		
			0	0		0.94375	0.806	71	0	0
					0		43	0.918		
			0	0	-			15	0	0
					1					

(下一页继续)

所有 19 个基准函数的整体性能。从而证实了 DMO 解决优化问题的能力。同时，DMO 的标准差值明显小于其他算法，说明 DMO 是稳定的。

收敛速度比较如图 3 所示，可以看出 DMO 在迭代过程的早期就能在其他竞争算法中找到最优解。我们把这归因于 alpha 的吸引力效应，在第一次迭代中将种群中的其他成员拉向最优解，也就是在

表 3(续)。

命令 功能	全局 优化	价值	d m o	协 定	cpsogsa	微 粒	a c o	s s a	s c a	国
罗森布 洛克 F17	0	最佳	0.701	28.1	22.4	0	4.12e	0.701	28.1	2.52e
		最差	32	47	06		+08	32	75	+01
		平均	24.64	28.1	900	0	4.14e	819.9	120	28.509
		SD 中	5	47	023		+08	6	4.4	26.56
		位数	6.103	28.1	305	0	4.13e	132.8	79.0	0.81019
			9	47 0	9.4		+08	4	44	26.212
			6.437	28.1	16	0	5.64e	2013	215.	
			1	47	425		+05	29.29	17	
			3.536		27.2	0	4.14e	9	29.1	
			5		52		+08		62	
格里万 克 F18	0	最佳	0	0.0920	0.000296	0	3.3582	0	2.015e-0	0
		最差		59	58 1.2424		3.3635	0.02951	5	0.02025
		平均	0	0.0920	0.28118	0	3.3626	8	0.92268	9
		SD 中		59 0	0.45161		0.0011	0.01263	0.17617	0.00194
		位数	0	0.0920	0.032588	0	41	7	0.26948	65
				59			3.363	0.00828	0.007378	0.00528
			0			0		84	6	28 0
								0.01109		
			0			0		3		
艾克 利 F19	0	最佳	0	0	0	0	19.426	2.128e-0	0.00	0
		最差			19.1		19.426	8 3.1591	09	
		平均	0	0	84	0	19.426	1.6136	20.3	0
		SD 中			1.87		7.23e-1	0.91477	01	
		位数	0	0	7	0	5 19.426	1.6462	15.6	0
					5.49				94	
			0	0	33 0	0			7.98	0
									72	
卡方									20.1	0
									33	
卡方		66. 406	2	3	4	3	7	5.05263 158	6.13157 895	3
			9	4	2	3	4			2
			7	7	6	9	7			4
渐进式。 西格。		0. 00 0	1	4	5	3	8	6	7	2

球面(F2)、平方和(F3)和四次方(F4)的问题搜索区域的中间。但是，在 Schwefell.2 (F8)的迭代过程中，DMO 并没有向最优解收敛，而是在迭代过程的后期向最优解收敛。

在优化过程中，监测搜索代理的行为是很重要的，可以看到他们从初始阶段到最终阶段如何改变或提高适应度函数的值。如图 4 所示，我们使用六(6)函数的二维实现来观察 DMO 中搜索代理的行为。从图中可以看出，DMO 算法是围绕搜索空间中有希望的区域进行搜索。在全局最优解附近的采样点分布很大，这表明 DMO 算法利用和探索了问题搜索空间中最有前途的区域。

4.3。CEC 2020 测试功能

这一套函数被设计成具有较高的难度，因为它们具有许多全局最优值和许多诱饵。利用 CEC 2020 评价 DMO 算法的有效性、稳定性和鲁棒性。所报道的实验均采用了 CEC 2020 测试功能，结果如表 4 所示。DMO 算法的结果优于其他算法，因为它能够找到 F20、F21、F23、F25 和 F27 的全局最优值，其余的值都非常接近最优值。GWO、AOA 和 CPSOGSA 也给出了竞争结果。Friedman 的测试结果也证实了 DMO 的优越性，DMO 在所有算法中排名第一，平均排名最低。收敛速度比较如图 3 所示。DMO 具有最好的收敛性，GWO、CPSOGSA 和 AOA 紧随其后。在这里，ACO 和 SCA 的结果比经典函数略差。图 4 显示了这六个基准函数的搜索历史。

#### 4.4。工程问题

新提出的 DMO 算法也被用于解决 12 个不同的工程设计问题，结果将在本节中介绍。本研究所采用的工程优化问题是在设计原则、资源限制和安全要求等特殊条件下寻找最优解。通常，元启发式算法不能直接找到约束优化问题的解。然而，配备了约束处理技术(cht)，优化器可以应付目标函数和相应的约束。该算法在每次迭代中使用目标函数和约束条件来评估候选总体的适应度，并根据计算出的适应度函数来评估下一代候选总体。

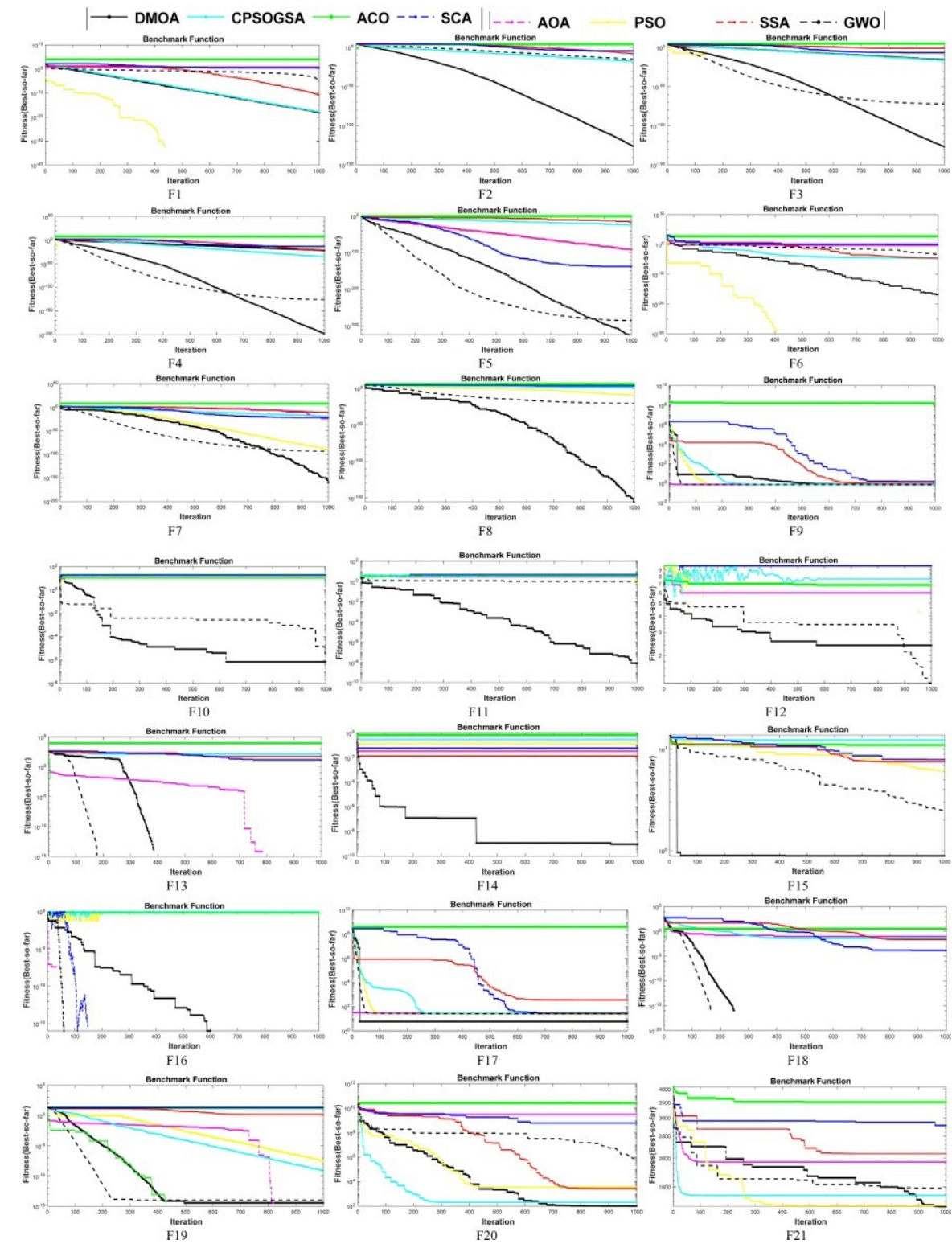


图3。经典和CEC 2020基准函数的收敛速度比较。



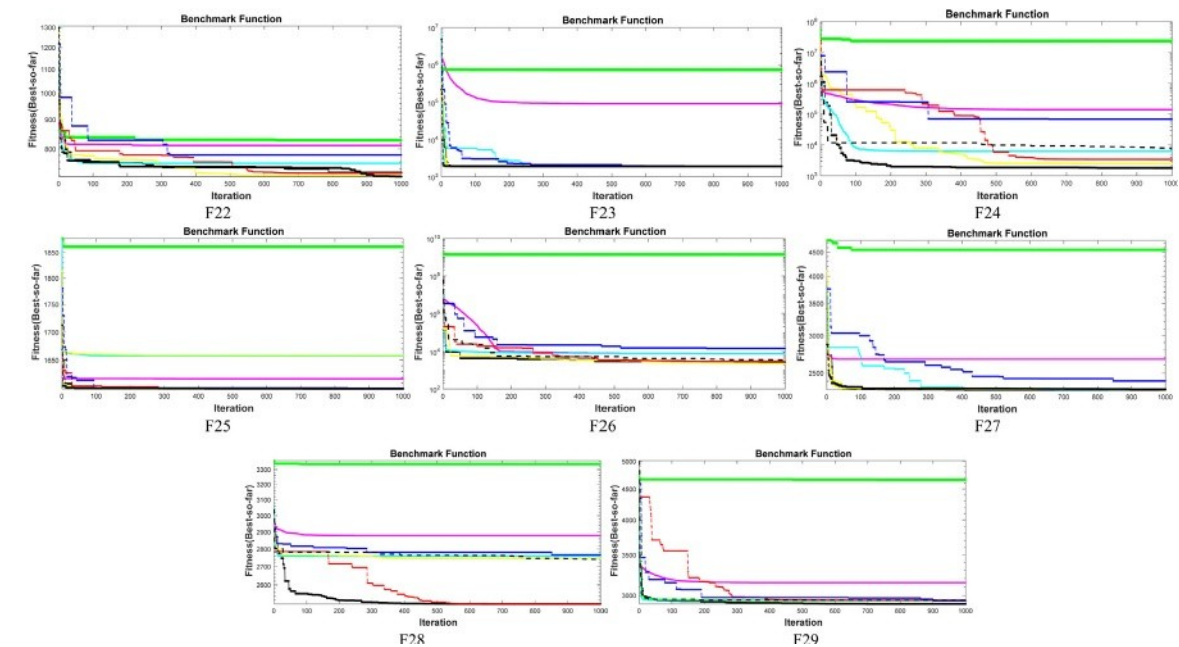


图 3。(续)。

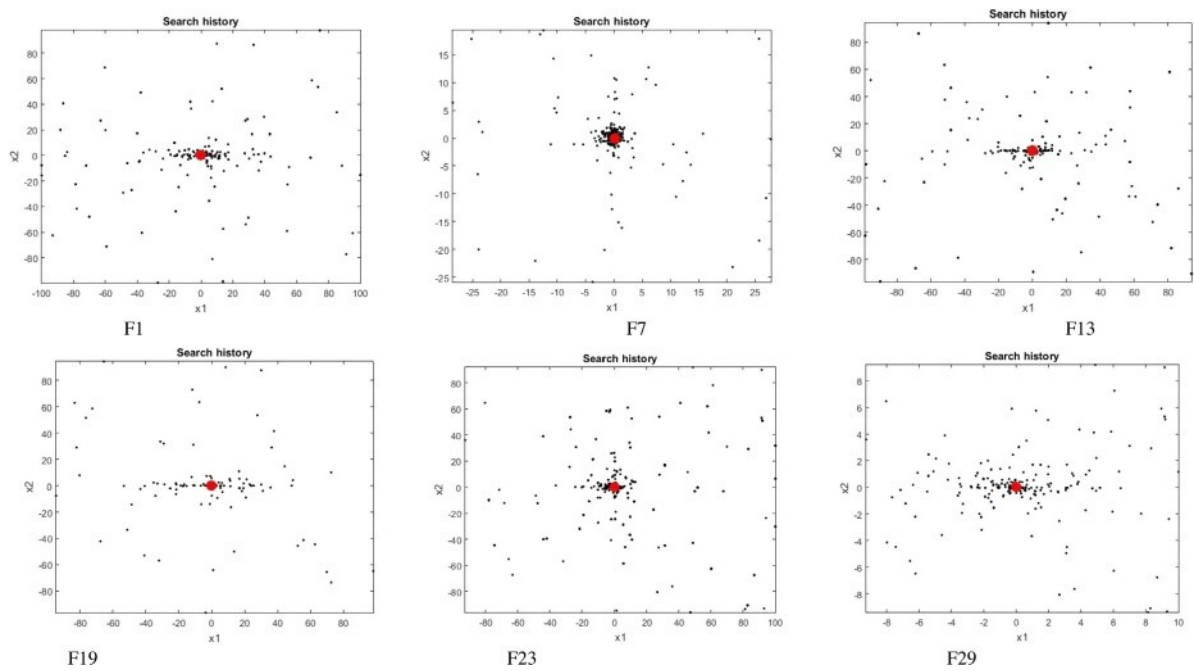


图 4。六个基准函数(F1-F6)的搜索历史记录。

4.4.1。焊接梁设计问题

焊接梁设计问题是一个梁受垂直力影响的最小化问题[67]。将所提出的 DMO 算法用于求解设计问题的最小制造成本，并与其他 7 种算法的结果进行比较。图 5 显示了 WBD 的示意图。受剪切应力( $\tau$ )、梁混合应力( $\theta$ )、杆件屈曲荷载( $P_c$ )、梁端挠度( $\delta$ )和侧约束。



表 4  
CEC 2020 测试功能结果。

命令 功能	全局选 择	价值	d m o	协定	cpsogsa	微粒	a c o	s s a	s c a	国
f 2 0	O n e  h u n d r e d .	最佳	100.	3.02e	106.	103.	2.57e	759.	3.24e	5729.7
		最差	43	+09	97	04	+10	13	+08	3.20e
		平均	0.50	3.02e	12	564	2.58e	895	1.46e	+08
		SD	24	+09	698	3	+10	9.3	+09	1.62e
		中位	100.	3.02e	599	209	2.57e	214	7.26e	+07
		数	2	+09	9.7	3.1	+10	7.5	+08	5.88e
				3.02e	459	203	3.13e	227	2.59e	+07 27
				+09	9.4	9.7	+07	4.4	+08	937
					550	128	2.57e	138	6.66e	
					8.4	4.4	+10	9.9	+08	
f 2 1	1 1 0 0	最佳	110	192	137	111	333	187	236	121.
		最差	1.4	9.3	8.5	2	5.4	0.4	9.5	1
		平均	134	192	252	180	351	246	178.	197
		SD	7	9.3	9.9	3.4	0	4	51	6.7
		中位	112	192	201	140	342	185	238	154
		数	6.3	9.3	4.7	4.5	6.3	9.9	9.7	7.3
			96.4	0	283	176.	43.6	322.		204.
			7		.22	23	99	15		43
			113		200	144	342	180		155
			6.5		2.6	1.8	8	1.5		9.4
f 2 2	7 0 0	最佳	715.	811.	719	708.	818.	725.	762.	710.
		最差	3	84	.81	63	4	3	47	94
		平均	742.	811.	798	725.	834.	754.	797.	755.
		SD	87	84	.66	83	1	19	74	16
		中位	733.	811.	757	717.	827.	728.	778.	728.
		数	22	84 0	.22	63	1	91	4	87
			5.10	811.	21.	4.30	3.77	12.3	8.56	11.9
			87	84	584	53	33	59	69	28
			734.		754	717.	827.	724.	779.	724.
			77		.56	68	88	86	07	77
f 2 3	1 9 0 0	最佳	1900.	91	1900.	1900	7.30e	1900.	191	190
		最差	3	91	6	1902.	+05	3	5.8	0.7
		平均	1902.	91	1903.	6	7.39e	1903.		190
		SD	5	91	3	1901.	+05	9		3.5
		中位	1901.	91	1901.	1	7.34e	1901.		190
		数	2	91	8	0.520	+05	7		2.3
			0.543	91	0.686	35	2416.8	0.840		0.75
			11		64	1901	7.34e	53		11
			1901.		1901.		+05	1901.		190
			1		7			4		2.4
f 2 4	1 7 0 0	最佳	171	1.39e	1836.7	203	2.34e	275	6	2849.3
		最差	5.3	+05	1.73e	2.5	+07	5.3	617	3.85e
		平均	181	1.39e	+05 19	14	2.34e	17	69	+05 19
		SD	9.2	+05	899 36	164	+07	993	235	155 69
		中位	174	1.39e	141	507	67.78	582	27	228
		数	9	+05 0	7518	1.5	2.34e	4.5	377	5619.5
			18.6	1.39e		331	+07	388	17	
			7	+05		1.8		8.4	830	
			171			337		434	18	
			5.6			0.7		2.9	894	
f 2 5	1 6 0 0	最佳	1600	161	160	160	1861.5	1600.	1600.	160
		最差	1600	7.6	0	0	1861.5	1	6	0
		平均	0.0007	161	183	165	1861.5	1601.	1602.	161
		SD	26	7.6	8.8	9	1.16e-1	2	9	7.1
		中位	1600	161	163	161	2 1861.5	1600.	1601.	160
		数		7.6	2.7	2.1	.e	6	1	2.4
				0	53.1	15.1		0.406	0.459	5.14
				161	09	5		04	16	03
				7.6	160	161		1600.	1601	160
					2.7	6.8		5		0.3
f 2 6	2 1 0 0	最佳	210	789	258	218	1.39e	226	400	252
		最差	3.8	0.4	9.3	2.3	+09	3.8	7.4	0.4
		平均	217	789	23	421	1.39e	26	35	15



		SD	8.5	0.4	934	1.3	+09	256	695	961
		中位	210	789	802	264	1.39e	618	11	780
		数	9.3	0.4	9.7	8.2	+09	1.6	483	9.7
			9.74	0	633	480.	7.443	567	648	450
			214	789	9.6	71	1.39e	0.6	4.4	2.9
			3.9	0.4	504	249	+09	357	10	682
					7.2	6.2		8	399	4.7
f	2	最佳	220	267	224	230	454	223	227	221
2	2	最差	0.1	5.6	2	0	7.2	4.3	6	2
7	0	平均	230	267	230	230	459	230	330	242
	0	SD	3.3	5.6	6.8	5.3	7.7	7.6	9.1	6.5
		中位	226	267	229	230	457	230	238	230
		数	9.4	5.6	9.4	2.2	0.2	0.4	7.6	8.5
			14.2	0	12.0	1.17	12.4	14.1	177.	29.1
			28	267	74	230	3	46	18	34
			230	5.6	230	2.2	457	230	236	230
			2.3		2.3		1.3	2.6	3.5	6.8
f	2	最佳	250	287	250	250	333	250	254	254
2	4	最差	0	7.9	0	0	5.8	0	2.5	6.9
8	0	平均	259	287	282	277	333	277	280	276
	0	SD	0.2	7.9	5.9	2.3	9.6	8.4	0.8	3
		中位	252	287	273	273	333	274	275	274
		数	5	7.9	8.6	2.1	7.9	0.3	9.1	0.7
			27.2	0	96.8	44.6	1.0	46.2	73.1	38.1
			03	287	14	73	415	54	38	26
			251	7.9	276	273	333	274	278	274
			5.2		9.8	7.4	8.1	7.5	1.7	3.8

(下一页继续)

WBD 的设计变量为:h = x<sub>1</sub> , l = x<sub>2</sub> , t = x<sub>3</sub> , b = x<sub>4</sub> 其中 , l 为长度 , h 为高度 , t 为厚度 , b 为棒材的焊缝厚度。WBD 问题的目标函数的数学表达式如式(8)所示。

min f (X) = x<sub>1</sub><sup>2</sup>x<sub>2</sub>1.10471 + 0.04811x<sub>3</sub>x<sub>4</sub>(14.0 + x<sub>2</sub>)

(8)

表 4(续)。

命令 功能	全局选 择	价值	d m o	协 定	cpsogsa	微 粒	ac o	s s a	s c a	国
f 2 9	2 5 0 0	请 接受 我 最 美 好 的 祝 愿	259 8.1	315 1.2	289 8.3	289 7.8	464 9.9	289 8.1	292 9.4	290 3.5
		最 坏 的	284 4.4	315 1.2	302 4.2	294 6.3	466 4.8	2 9 4 9	298 5.9	294 9.7
		海损	280 3.2	315 1.2	293 4.9	292 4.5	465 7.2	292 3.7	2 9 5 8	2 9 3 8
		s d	9.2 352	0	29. 383	23. 234	4.7 329	24. 085	14. .6 3	14. 205
		中 位 数	280 9.7	315 1.2	294 4.7	294 3.5	465 7.6	294 3.7	296 1.9	2 9 4 6
卡方		57.63 333	1 4 0	6 5 0	4 6 0	2 4 0	8 0 0	3 0 0	5 9 0	4 2 0
渐进式。 西格。		4.47e- 10	1	7	5	2	8	3	6	4

受制于

$$s_1\left( X \right) = \tau \left( X \right) - \tau_{max} \leq 0,$$

$$s_2\left( X \right) = \sigma \left( X \right) - \sigma_{max} \leq 0,$$

$$s_3\left( X \right) = \delta \left( X \right) - \delta_{max} \leq 0,$$

$$s_4\left( X \right) = x_1 - x_4 \leq 0,$$

$$s_5\left( X \right) = P - P_c\left( X \right) \leq 0,$$

$$s_6\left( X \right) = 0.125 - x_1 \leq 0,$$

$$s_7\left( X \right) = 1.10471x_1^2 + 0.04811x_3x_4\left( 14.0 + x_2 \right) - 5.0 \leq 0$$

设计变量的间隔时间:

$$0.1 \leq x_1 \leq 2, \quad 0.1 \leq x_2 \leq 10, \quad 0.1 \leq x_3 \leq 10, \quad 0.1 \leq x_4 \leq 2$$

在哪里

$$\tau\left( \vec{l} \right) = \sqrt{\tau'^2 + 2\tau'\tau''\left( \frac{x_2}{R} \right) + \left( \tau'' \right)^2},$$

$$\tau' = P/\sqrt{2x_1x_2}, \tau'' = MR/J, M = P\left( L + \frac{x_2}{2} \right),$$

$$J = 2 \left\{ \sqrt{2} x_1 x_2 \left[ \left( \frac{(x_2^2)}{4} \right) + \left( x_1 + \frac{x_3}{2} \right)^2 \right] \right\},$$

$$P_c(X) = \frac{4.013E \sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2} \left( 1 - \frac{x_3}{2L} \sqrt{E/4G} \right)$$

WBD 的参数设置如下:

$$\sigma_{max} = 3000 \text{ psi}, P = 6000 \text{ lb}, L = 14 \text{ in}, \delta_{max} = 0.25 \text{ in}, E = 3 \times 10^6 \text{ psi}, \\ \tau_{max} = 13600 \text{ psi}, \text{ and } G = 12 \times 10^6 \text{ psi}$$

用于解决焊接梁设计问题的 8 种优化器的结果和统计分析如表 5 所示。DMO 能够在所有实验使用的最大迭代中找到 WBD 的全局最小成本。掸邦军和国航密切关注这一点;而 DMO 的标准差最小,表明了 DMO 在解决该工程问题中的有效性和鲁棒性。图 6 所示的收敛速度证实了 DMO 在解决这一问题上的优越性。Wilcoxon 符号秩检验假设得到的结果的平均分布是相同的。对 DMO 和所有被考虑的算法进行两两比较(p 值和 Z 分数),除 SSA 和 GWO 外,均返回负秩。负秩表示 DMO 的均值最小,说明 DMO 返回的均值是最好的,并且在期望的结果附近是稳定的。

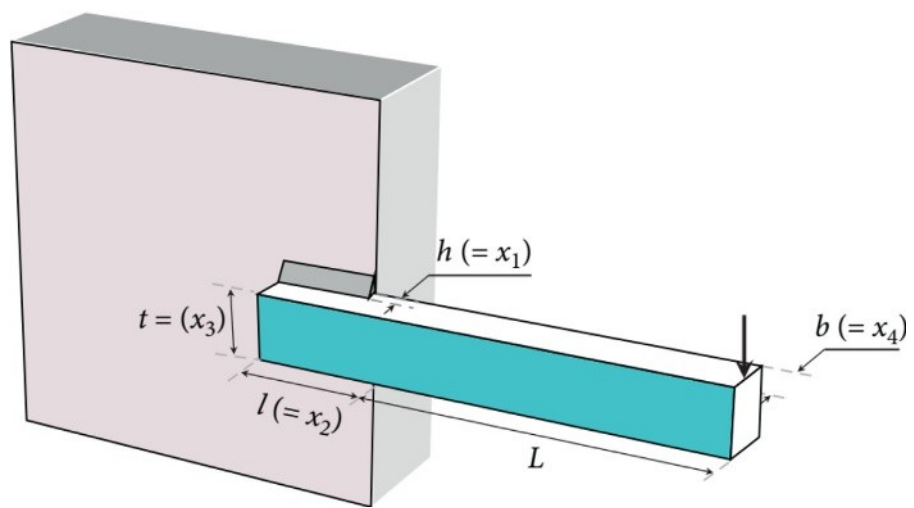


图 5。WBD 的说明。

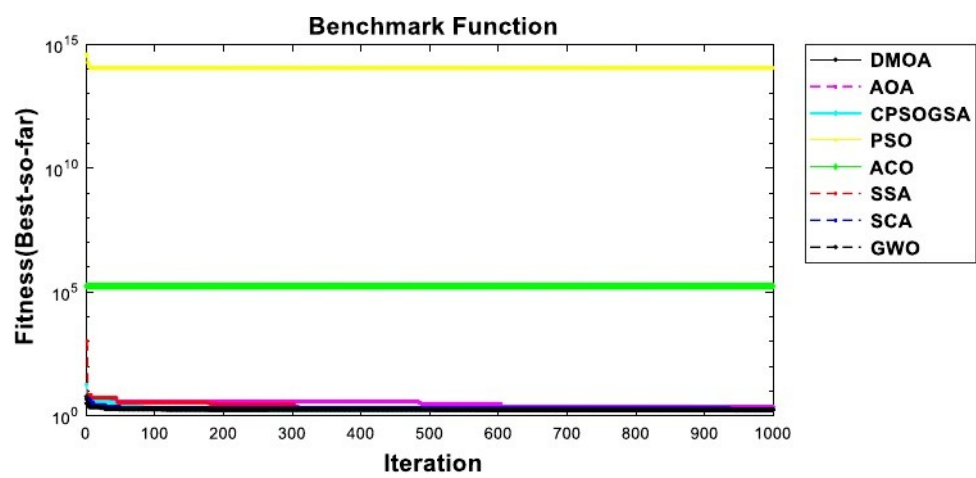


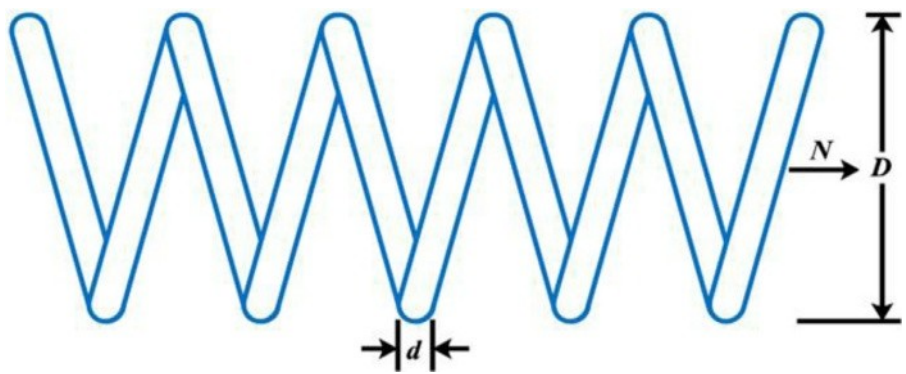
图 6。WBD 的收敛速度。

表 5

焊接梁设计比较结果。

x1 x2 x3 DMO 0.2055705 3.2567724				x4	请接受我 最美好的祝愿，	最坏的	海损	sd	中位数 Z	p 值
9.036177	AOA 0.1792865	3.7660206	9.429738	0.2057696		1.7004	1.6964	0.0014007	1.6959 na	na
CPSOGSA 0.1862275	3.6614473	8.958526	PSO	0.2336683		2.1397	2.1397	0	2.1397-1.569a	0.054
0.786028	22 ACO 14	3 SSA 0.1817486	3.738515	0.2093323		2.1397	1.8998	0.18888	1.8261-1.714a	0.15
9.036625	SCA 0.2074989	3.7838778	8.982884	2	1.6953	2.3703	1.09e +14	0.047676	1.09E+14-1.594a	0.073
GWO 0.203239	3.3027979	0.037416	a 基于负秩。	2	2.1397	1.09e +14	1.69e +05	2.96e-11	1.69E+05-1.947a	0.127
b 基于正秩。				0.2057296	1.6986	1.69e +05	1.769	0.082132	1.7357-2.019b	0.047
				0.2169487	1.09e +14	2.0663	1.8195	0.026297	1.8211-1.859a	0.04
				0.2057535	1.69e +05	1.8776	1.6975	0.0015368	1.6969-1.119b	0.048
					1.6953	1.7012				
					1.7738					
					1.6955					





图七。CSD 问题的示意图。

4.4.2。压缩弹簧设计问题(CSD)

压缩弹簧设计问题(CSD)是使压缩弹簧的重量最小化，如图 7 所示[68]。该问题有三个设计变量，即：有源线圈个数  $P = x_1 \in [2,15]$ ，线圈或绕组的直径  $D = x_2 \in [0.25,1.3]$ ，导线的直径  $D = x_3 \in [0.05,2]$ 。式(9)给出了 CSD 的目标函数模型。

假设  $l = [l_1 l_2 l_3] = [d DP]$

$$Min f(\vec{l}) = (l_3 + 2) l_2 l_1^2 \tag{9}$$

受制于

$$g_1(\vec{l}) = 1 - \frac{l_2^3 l_3}{7178514} \leq 0,$$
$$g_2(\vec{l}) = \frac{4l_2^2 - l_1 l_2}{12566 (l_3 l_1^3 - l_1^4)} + 1/5108 l_1^2 \leq 0,$$
$$g_3(\vec{l}) = 1 - \frac{140.45 l_1}{l_2^2 l_3} \leq 0,$$
$$g_4(\vec{l}) = \frac{l_1 + l_2}{1.5} - 1 \leq 0.$$

设计变量的间隔为：

$$0.05 \leq l_1 \leq 2.00, 0.25 \leq l_2 \leq 1.30, 2.00 \leq l_3 \leq 15.0$$

表 6 中的结果比较了 DMO 与其他 7 种优化算法的最佳结果和统计分析。DMO、CPSOGSA、SSA 和 GWO 均返回目标函数的最佳最小代价。但是，DMO 返回的均值和标准差最小，说明本文算法的结果是稳定的。同样，从图 8 的收敛速度可以看出，DMO、CPSOGSA、SSA 和 GWO 能够在迭代过程的早期找到最优代价，并稳定在最优代价附近。Wilcoxon 符号秩检验结果表明，除 SSA、CPSOGSA 和 GWO 外，DMO 和所有考虑的算法的两两比较结果均为负秩。从而证实了 DMO 返回了最好的平均值，并且在期望的结果周围是稳定的。

4.4.3。压力容器设计问题(PVD)

压力容器设计问题(PVD)包括一个两端装有半球形封头的圆柱形容器。PVD 的示意图如图 9 所示[69]。容器的厚度为  $T_s(x_1)$ 、封头厚度  $T_h(x_2)$ 、船舶内半径  $R(x_3)$ 和船舶盘头长度  $L(x_4)$ 为四个决策变量。Eq.(10)表示 PVD 的目标函数模型。

给定  $l = [l_1 l_2 l_3 l_4] = [T_s T_h R L]$  ,

$$Min f(\vec{l}) = 0.6224 l_1 l_3 l_4 + 1.781 l_2 l_3^2 + 3.1661 l_1^2 l_4 + 19.84 l_1^2 l_3 \tag{10}$$



表 6  
与惩罚部署的比较结果。x1

	x2	x3	请接受我 最美好的 祝愿，	最 坏 的	海损	sd	中位数	z	p 值
DMO 0.13915 AOA									
0.148288 CPSOGSA 0.13915	1.3	11.89243		3.6619	3.6619	1.56e-15 0	3.6619	na	na
PSO 2 ACO 3 SSA 0.137177	1.3	15		6.265	6.265	0.022278	6.265	- 1.549	0.128
SCA 0.138635 GWO	1.3	11.89243		3.6717	3.6717	2.89e-13	3.6619	a-1.59	0.119
0.139154 a 基于负秩。 b 基于 正秩。	2	2	3.6619	3.7303	409.77	5.78e-14	409.77	1b-2.1	0.029
	2	1	6.265	409.77	209.93	0.016854	209.93	98a-1.	0.09
	1.242554	12.85631	3.6619	209.93	3.6826	0.022163	3.6817	641a-1	0.108
	1.3	11.8299	409.77	3.7183	3.6878	4.59e-06	3.6793	.750b-	0.063
	1.3	11.89314	209.93	3.7327	3.6619		3.6619	1.829a	0.328
			3.6619	3.6619				-1.527	
			3.6635					b	
			3.6619						

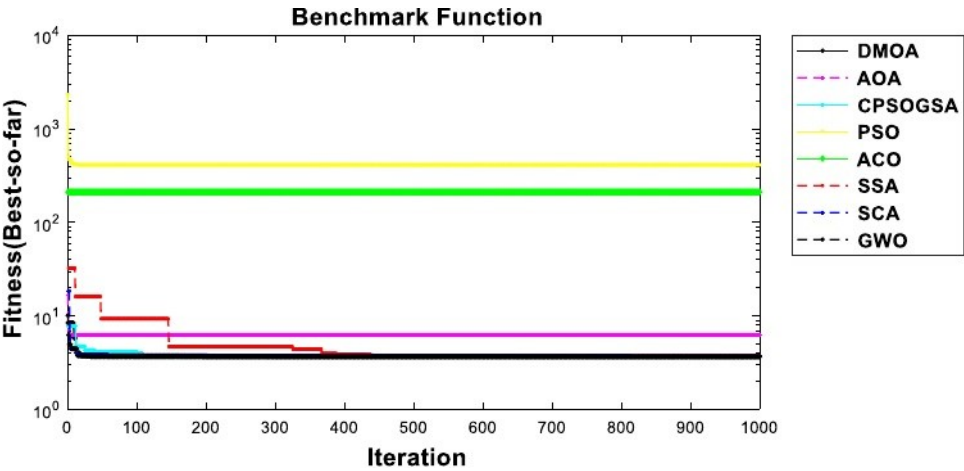


图 8。CSD 的收敛速度。

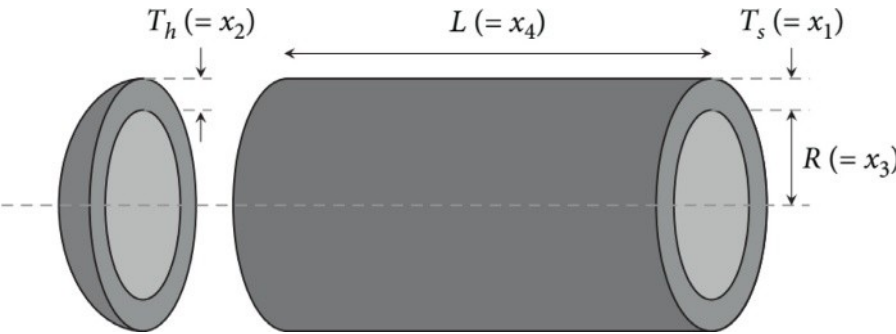


图 9。PVD 示意图。

$$g_1(\vec{l}) = -l_1 + 0.0193l_3 \leq 0,$$



$$g_2\left(\vec{l}\right)=-l_3+0.00954l_3\leq 0,$$

$$g_3\left(\vec{l}\right)=-\pi l_3^2l_4-\frac{4}{3}\pi l_3^3+1296000\leq 0,$$

$$g_4\left(\vec{l}\right)=l_4-240\leq 0.$$

表 7  
PVD 结果比较。

	x 1	x 2	x 3	x 4	请 接受 我 最 美 好 的 祝 愿	最 坏 的	海 损	s d	中 位 数	z	p 值
d m o	1.093 571	1.04e-1 7	65.22 523	1 0	230 2.5	230 2.5	230 2.5	9.25e-1 3	230 2.5	n a	n a
协 定	0.107 099	0	41.30 816	2 0 0	403 0.5	403 0.5	403 0.5	0	403 0.5	-1.8 59a	0. 05 7
cpsogsa	1.093 588	0	65.22 523	1 0	230 2.5	363 7.9	315 3.3	580 .38	3.46e +03	-2.0 95b	0. 02 9
微 粒	1 0	1 0	53.66 112	71.7 154	2.04e +05	2.04e +05	2.04e + 05	1.9 492	2.04e +05	-2.2 95a	0. 03 7
ac o	1	4	3 .	2	1.68e +16	1.68e +16	1.68e +16	8.1 368	1.68e +16	-2.1 73a	0. 01 8
s s a	0.553 754	0	43.71 611	157.5 719	230 2.5	363 8.5	320 9.8	567 .76	362 4.6	-2.0 16b	0. 04 1
s c a	0	0	40.32 405	2 0 0	231 3.8	606 0.6	506 3.1	167 7.1	605 6.8	-1.4 30a	0. 19 1
国	1.090 845	0.002 287	65.22 627	1 0	230 2.6	605 5.6	280 4.3	129 7.1	230 2.7	-1.0 19b	0. .4 1

a 基于 b 在消极的等级上。  
基于 在积极的行列。

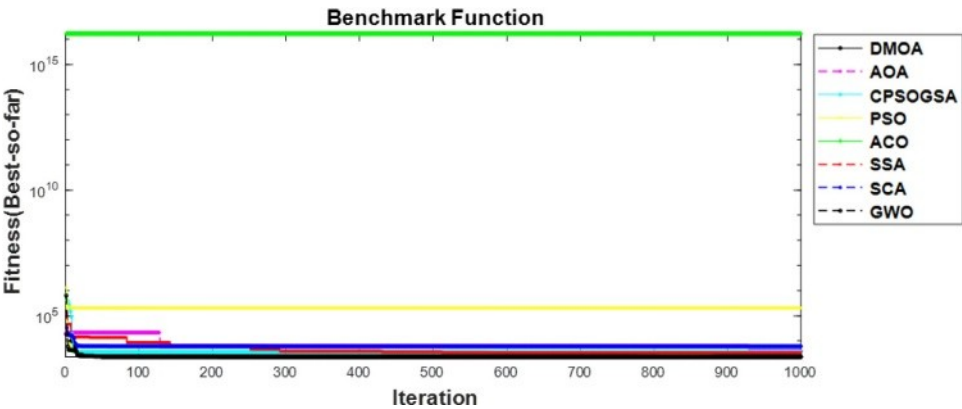


图 10。PVD 的收敛速度。

区间内定义的设计变量如下:

$0 \leq l_1 \leq 99, 0 \leq l_2 \leq 99, 10 \leq l_3 \leq 200, 10 \leq l_4 200$

同样，表 7 比较了 DMO 与其他七种优化算法的最佳结果和统计分析。DMO、CPSOGSA 和 SSA 的目标函数的最小代价均为最佳，GWO 的最小代价次之。像之前的案例一样，DMO 通过返回最小平均和标准偏差来显示其优越性。图 10 的收敛速度证实了 DMO 的这种优势，紧随其后的是 CPSOGSA、SSA 和 GWO。Wilcoxon 符

号秩检验结果表明，除 SSA、CPSOGSA 和 GWO 外，DMO 和所有考虑的算法的两两比较结果均为负秩。这证实了 DMO 返回了最好的平均值，并且在期望的结果周围是稳定的。

#### 4.4.4。减速器设计问题(SRD)

减速机是齿轮箱系统中最关键的部件之一。该问题的目标是在 11 个约束条件下使减速器的重量最小(图 11) [70]。这个设计问题有 7 个设计变量，即: $x_1$  面宽(b)， $x_2$  牙齿的模量(m)是  $x_3$  表示小齿轮的齿数(z)， $x_4$  表示轴承之间的第一个轴的长度( $l_1$ )，( $l_2$ ，则轴承之间的第二轴的长度为  $x_5$  的直径( $d_1$ )和第二轴( $d_2$ )用  $x_6, x_7$ ，分别。式(11)给出了该问题的目标函数的数学模型。

已知  $x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2]$

$$\begin{aligned} \min f(x) = & 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) \\ & + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \end{aligned} \tag{11}$$

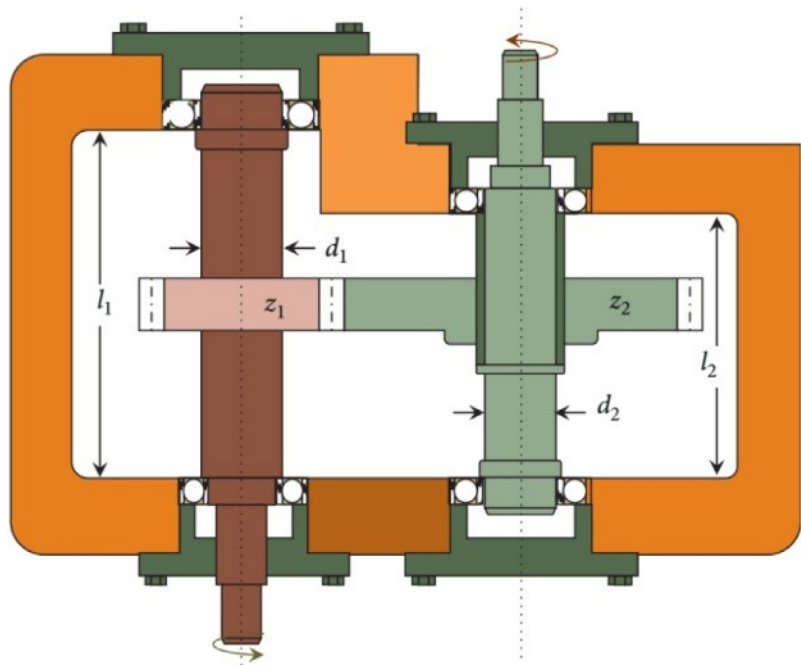


图 11。SRD 示意图。

受制于

$$g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0,$$
$$g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0,$$
$$g_3(x) = \frac{1.93x_4^2}{x_2x_6^4x_3} - 1 \leq 0,$$
$$g_4(x) = \frac{1.93x_5^2}{x_2x_7^4x_3} - 1 \leq 0,$$
$$g_5(x) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16}}{110x_6^3} - 1 \leq 0,$$
$$g_6(x) = \frac{\sqrt{\frac{x_2x_3}{85x_7^3}}}{85x_7^3} - 1 \leq 0,$$
$$g_7(x) = \frac{x_2x_3}{40} - 1 \leq 0,$$
$$g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0,$$
$$g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0,$$
$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0,$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0$$

$$-1 \leq 0,$$

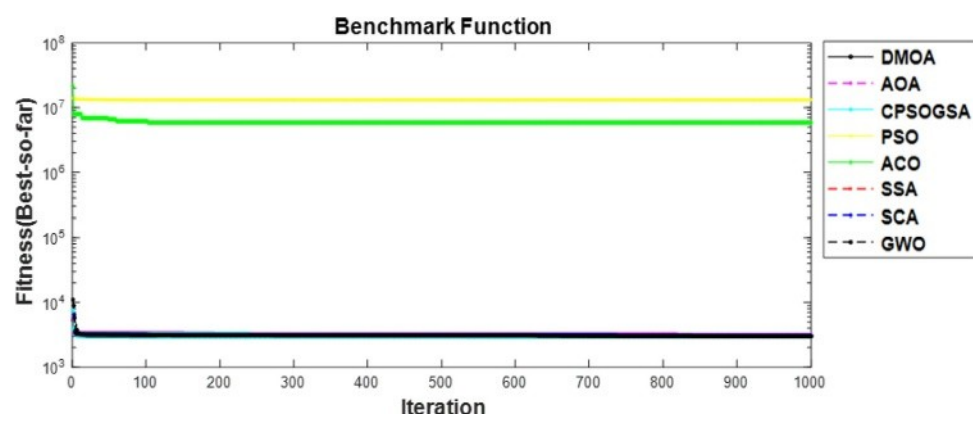


图 12。SRD 的收敛速度。

时间间隔给出如下:

$$2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, 7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, 5.0 \leq x_7 \leq 5.5$$

如表 8 所示的结果和统计分析表明，DMO 能够找到最优的代价，并返回最小的平均和标准差，这说明本文算法的结果是稳定的。我们注意到 AOA 陷入了局部最小值。同样，图 12 中的收敛速度显示了 DMO 如何能够在迭代过程的早期找到最优代价，并稳定在最优代价附近。除 GWO 外，Wilcoxon 符号秩检验结果返回一个负秩。从而证实了 DMO 返回了最好的平均值，并且在期望的结果周围是稳定的。

4.4.5。三杆桁架设计问题(3-BTD)

三杆桁架问题的目标是使受载三杆桁架的体积在各杆的应力( $\sigma$ )约束下达到最小。有两个设计变量，即横截面积  $A_1(=x_1)$ 和  $A_2(=x_2)$ 如图 13 所示。3-BTD 的数学模型为式(12)[71]。

$$\min f(X) = \left(2\sqrt{2}x_1 + x_2\right) \times l \tag{12}$$

受制于

$$\begin{aligned} g_1(X) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0, \\ g_2(X) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0, \\ g_3(X) &= \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0 \end{aligned}$$

$L = 100$  厘米。 $P = 2$  千牛顿/厘米<sup>3</sup>， $\sigma = 2$  kN/cm<sup>3</sup>，取值范围: $0 \leq x_1, x_2 \leq 1$

表 9 给出了所使用算法的最佳结果和静态分析。DMO 算法的最佳目标值与 SSA 算法的最佳目标值相等，且优于其他算法。除蚁群算法和粒子群算法外，所有被测试算法的返回值都接近全局最小代价，如图 14 所示。平均结果、标准差和 Wilcoxon 符号秩检验(Z 和 p-value)结果的值证实了 DMO 的优越性。

4.4.6。轮系设计问题(GTD)

齿轮传动设计问题(GTD)，最早由[72]提出，属于机械工程领域的无约束离散设计问题。目标是使输出/输入



	x 1	x 2	x 3	x 4	x 5	x 6	x 7	请接受我最好的祝愿	最坏的	海损	s d
d m o 协 定	3.497 599	0 .7	1 7	7 .3	7.713 535	3.350 056	5.285 631	299 3.6	299 3.6	299 3.6	4.63e- 13
cpsogsa	3 .6	0 .7	1 7	7.620 008	8 .3	3.520 421	5.275 857	311 4.2	311 4.2	311 4.2	0
微粒	3.497 599	0 .7	1 7	7 .3	7.713 535	3.350 056	5.285 631	299 3.6	299 3.6	299 3.6	5.00e- 13
ac o	5 .5	5	5	5 .5	5 .5	5	5.101 362	1.32e +07	1.32e +07	1.32e +07	4.02e- 09
s s a	5	2	6	1	3	4	7	5.87e +06	5.87e +06	5.87e +06	2.84e- 09
s c a	3.497 619	0 .7	1 7	7.89 341	7.925 864	3.528 122	5.285 751	299 3.6	307 6.2	3 0 1 7	19. 782
国	3 .6	0 .7	1 7	8 .3	8 .3	3.423 716	5.338 549	302 9.5	316 1.4	309 7.1	28. 334
	3.497 891	0 .7	17.00 134	7.837 797	7.892 772	3.352 131	5.286 415	299 4.3	301 5.5	300 0.7	4.0 934

a 基于在消极的等级上。  
b 基于在

表 8 S R D 的比较结果。



积  
极  
的

行  
列  
。

表 9  
3-BTD 的比较结果。

	x 1	x 2	请 接受我 最美好的 祝愿 ，	最 坏 的	海 损	s d	中 位 数	z	p 值
dm	0.2195	0.1883	1.69	1.70	1.69	0.0022	1.69	n	n
o	15	73	53	35	68	24	58	a	a
协	0.2168	0.1910	2.27	2.27	2.27	0	2.27	-2.5	0.0
定	93	97	66	66	66		66	97a	26
cpsogsa	0.2611	0.1821	1.72	2.34	1.88	0.179	1.82	-1.2	0.2
	33	65	49	61	74	01	21	85a	05
微	0.2195	0.1883	1.09e	1.09e	1.09e	0.0476	1.09e	-1.2	0.2
粒	15	73	+14	+14	+14	76	+14	57a	12
ac	1	2	1.69e	1.69e	1.69e	2.96e-1	1.69e	-1.2	0.2
o			+05	+05	+05	1	+05	58a	19
ss	0.2195	0.1883	1.69	2.07	1.78	0.0839	1.75	-2.0	0.0
a	15	73	53	31	15	79	76	47b	40
sc	0.219	0.1877	1.76	1.89	1.83	0.0374	1.83	-2.7	0.0
a	32	73	65	48	09	59	56	31a	17
国	0.2195	0.1883	1.69	1.70	1.69	0.0013	1.69	-1.6	0.
	06	77	58	23	74	01	71	30a	1 4

基于消极的等级。基于积极的排名。

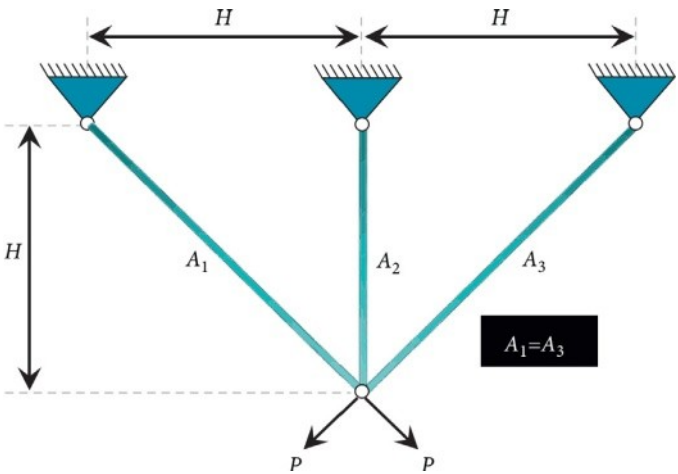


图 13。三杆桁架设计问题示意图。

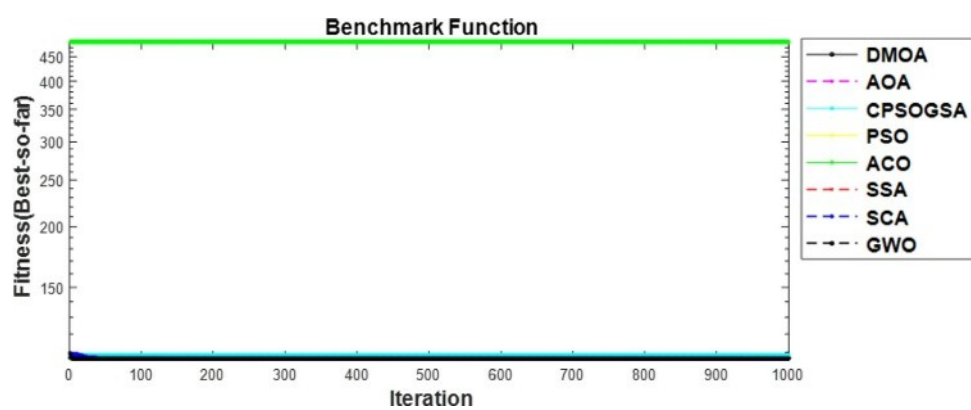


图 14。3-BTD 的收敛速度。