

# 求解复杂约束优化问题的集成多策略差分进化算法

张佳玉,潘志庚

(南京信息工程大学人工智能学院(未来技术学院),南京 210044)  
E-mail:003443@nuist.edu.cn

**摘要:**工程和科学领域中的优化问题常常具有大量的约束限制,称为约束优化问题.这类问题要求算法有能力在可行域中寻找问题的最优解.本文针对约束优化问题提出一种集成多策略的差分进化算法(Differential Evolution with Ensemble Multi-Strategies, EMSDE).首先,提出一种用于约束优化的参数自适应策略,利用归一化罚函数作为权重引导参数自适应地生成.其次,结合约束和动态罚函数法设计一种新的约束处理技术.最后,采用 CEC2017 约束优化基准函数来测试 EMSDE 和 7 种经典的约束优化算法.实验结果表明,相比 7 种经典的算法,EMSDE 算法具有很强的竞争力.

**关键词:**约束优化;差分进化;参数自适应;约束处理技术

中图分类号: TP18

文献标识码: A

文章编号: 1000-1220(2024)07-1629-10

## Differential Evolution with Ensemble Multi-strategies for Solving Complex Constrained Optimization Problems

ZHANG Jiayu, PAN Zhigeng

(School of Artificial Intelligence/School of Future Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China)

**Abstract:** Optimization problems in engineering and science often have a large number of constraints, which are called constrained optimization problems. This type of problem requires the algorithm to be able to find the optimal solution in the feasible domain. In this paper, we propose the differential evolution with ensemble multi-strategies (EMSDE) for constrained optimization problems. First, a parameter adaptation strategy for constrained optimization is proposed, using a normalized penalty function as a weight to guide the adaptive generation of parameters. Second, a new constraint handling technique is designed by combining the -constraint and the dynamic penalty function method. Finally, the CEC2017 constrained optimization benchmark functions are used to test EMSDE and seven typical constrained optimization algorithms. The experimental results demonstrate that the EMSDE algorithm is competitive compared with the seven typical algorithms.

**Keywords:** constraint optimization; differential evolution; parameter self-adaptive; constraint handling techniques

### 0 引言

约束优化问题(Constraint Optimization Problems, COPs)广泛存在于现实世界的诸多领域,例如作业调度<sup>[1]</sup>、金融投资<sup>[2]</sup>、网络通信<sup>[3]</sup>、传感器部署<sup>[4]</sup>、压力容器设计<sup>[5]</sup>和电路布线<sup>[6]</sup>等.这类问题要求在满足约束条件的前提下实现目标函数的最大化或最小化<sup>[7]</sup>.基于种群的元启发式优化算法常用于求解约束优化问题,特别是求解具有非凸、不可微、不连续等复杂特征的约束优化问题时,能够实现出色的结果.差分进化(Differential Evolution, DE)是元启发式优化算法的代表之一,在过去的 20 多年时间里已经被用于解决大量实际问题<sup>[8,9]</sup>和经典测试集问题<sup>[10,11]</sup>.

在 DE 的一般框架中,比例系数  $F$  和交叉概率  $CR$  是两个影响 DE 性能的关键参数,决定了子代向量的生成和种群的搜索趋向.迄今,学者针对和的最优参数设置开展了大量的研究.早期的研究致力于以实验的方式寻找  $F$  和  $CR$  的一个固

定最优配置.例如,Storn 等人<sup>[12]</sup>建议把  $F$  固定为 0.5,以实现更好的效果.但这种固定参数的方法逐渐被认为缺乏可靠性.于是,有学者提出为  $F$  和  $CR$  设置一个取值区间平衡种群的开发性与探索性<sup>[13]</sup>.例如,Gämperle 等人<sup>[14]</sup>将  $CR$  的范围设置在  $[0.3, 0.9]$ . Wang 等人<sup>[15]</sup>提出将若干控制参数随机组合来生成一组新的  $F$  和  $CR$ .随后, Brest 等人<sup>[16]</sup>提出 jDE,每一代以不同的概率更新  $F$  和  $CR$  的值.目前,参数自适应策略是一种主流方法,它能根据不同的优化问题自适应地确定不同的参数以搜索全局最优解.例如, Qin 等人<sup>[17]</sup>提出了一种自适应 DE 算法,称为 SaDE,其中  $F$  和  $CR$  是通过学习历史成功经验来自适应生成的. Zhang 和 Sanderson<sup>[18]</sup>提出的 JADE 同样采用了自适应机制,根据柯西分布和高斯分布,分别为每个个体生成  $F$  和  $CR$ ,并使用上一代成功的个体信息更新  $F$  和  $CR$  的分布均值  $\mu_F$  和  $\mu_{CR}$ ,以生成新的  $F$  和  $CR$ .

尽管 DE 已经在全局优化问题上取得了一定的成功,但是其无法直接应用于求解 COPs. 一个重要的原因在于 COPs

要求算法在寻找最优解的同时还需要考虑约束限制,即在可行域内寻找最优解<sup>[19]</sup>.因此,设计约束优化算法,不仅要考虑优化算子的设计,还需要设计高效的约束处理技术.目前,流行的约束处理技术有:可行解优先<sup>[20]</sup>、 $\varepsilon$ 约束<sup>[21]</sup>、随机排序<sup>[22]</sup>和罚函数法<sup>[23]</sup>.可行解优先是 Deb 等人<sup>[20]</sup>针对 COPs 提出的一种比较规则,鼓励优化算法保留约束冲突更小的个体,让种群朝着可行区域进化. $\varepsilon$ 约束由 Takahama 和 Sakai<sup>[24,25]</sup>提出,是针对可行解优先的一种改进,使用阈值  $\varepsilon$  来松弛比较规则,适当接受约束冲突较差但是适应度较好的个体.另外,他们还提出了基于等级的  $\varepsilon$  约束来给等级不同的个体分配不同的参数,很大程度上平衡了种群的多样性和收敛性.但是  $\varepsilon$  约束还是存在过度依赖  $\varepsilon$  初始值和阈值变化函数的问题.随机排序由 Runarsson 和 Yao<sup>[26]</sup>提出,通过一定的概率决定以目标函数或是约束冲突来决定两个个体的优劣,进而排序整个种群.和上述基于比较规则的约束处理技术相比,罚函数法避免了人工比较规则的设计,因其构造简洁得到了广泛应用.罚函数法的主要思想是增加惩罚项来构造新的适应度函数,根据惩罚项系数是否变化分为静态罚函数法和动态罚函数法.静态罚函数法的惩罚项系数不依赖于迭代代数,通常是一个定值.Homaifar 等人<sup>[27]</sup>提出了一种分级静态罚函数法,对不同个体采用不同等级的惩罚系数,违反约束的等级越高,惩罚系数越大.虽然这种方法相较于单一的惩罚系数更合理,但其通用性不强.动态罚函数法不需要事先设定惩罚项系数. Joines 等人<sup>[28]</sup>采用了动态罚函数法,惩罚项系数随着迭代次数的增大而增大,使得算法在迭代后期的搜索区间集中在可行区域内.虽然这种方法相比于静态惩罚函数的性能有所提升,但是寻找理想的惩罚系数依然是一个挑战.

综上所述,处理约束优化问题的关键在于设计优秀的优化算子和约束处理技术.为此,本文提出一种集成多策略的差分进化算法(Differential Evolution with Ensemble Multi-Strategies, EMSDE).首先,采用 DE 的一个优秀变体 JADE 作为优化算子,并且针对约束优化问题,提出将罚函数归一化之后作为权重来自适应地调节关键参数  $F$  和  $CR$ .其次,结合  $\varepsilon$  约束和动态罚函数法设计一种新的约束处理技术,并且采用阈值  $\varepsilon$  作为惩罚项系数动态调整罚函数.具体细节在第 2 节进行详细阐述.

## 1 预备知识

### 1.1 问题定义

现实世界中的许多含有实际参数的优化问题都受到一定的约束.不失一般性,以最小化为例,COPs 可以用公式(1)描述:

$$\begin{aligned} \min f(\mathbf{x}), \mathbf{x} = (x_1, x_2, \dots, x_D) \in S \\ \text{s. t. } g_i(\mathbf{x}) \leq 0, i = 1, \dots, p \\ h_j(\mathbf{x}) = 0, j, \dots, q \end{aligned} \quad (1)$$

其中,  $f(\mathbf{x})$  是目标函数,  $\mathbf{x}$  是决策向量,  $x_k$  是向量  $\mathbf{x}$  的第  $k$  个分量,  $D$  代表  $\mathbf{x}$  的维度,  $g_i(\mathbf{x}) \leq 0$  代表第  $i$  个不等式约束,  $h_j(\mathbf{x}) = 0$  是第  $j$  个等式约束.  $S = \prod_{i=1}^D [L_i, U_i]$  是向量  $\mathbf{x}$  的决策空间,  $L_i$  和  $U_i$  分别代表  $x_i$  的下界和上界.  $p$  和  $q$  分别代表不等式约束和等式约束的数量.当全部的不等式约束  $g_i(\mathbf{x})$

$\leq 0$  和等式约束  $h_j(\mathbf{x}) = 0$  时,  $\mathbf{x}$  被定义为可行解.由于计算误差,在实际求解中,若等式约束满足公式(2):

$$|h_j(\mathbf{x})| - \delta \leq 0, j = 1, \dots, q \quad (2)$$

则认为  $h_j(\mathbf{x}) = 0$ , 即满足等式约束.式(2)中,等式约束的容忍参数一般设置为  $\delta = 10^{-4}$ .  $\mathbf{x}$  的约束冲突定义为:

$$\varphi(\mathbf{x}) = \sum_{i=1}^p \max\{g_i(\mathbf{x}), 0\} + \sum_{j=1}^q |h_j(\mathbf{x})| \quad (3)$$

如果约束冲突  $\varphi(\mathbf{x}) = 0$ , 则代表  $\mathbf{x}$  是可行解, 否则  $\mathbf{x}$  是不可行解.

### 1.2 $\varepsilon$ 约束

$\varepsilon$  约束作为一种约束处理技术,采用阈值  $\varepsilon$  重定义了两个个体的比较规则.在比较两个个体的优劣时,当两个个体的约束冲突都小于阈值或约束冲突相等时,适应值越小的个体越好;否则,约束冲突越小的个体越好. $\varepsilon$  约束中的阈值  $\varepsilon$  通常用公式(4)和公式(5)计算:

$$\varepsilon^{(g)} = \begin{cases} \varepsilon^{(0)} \left(1 - \frac{g}{G}\right)^{cp}, & \frac{g}{G} \leq T_c \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$cp = \frac{\ln \varepsilon^{(0)} + con}{\ln(1 - T_c)} \quad (5)$$

其中,  $G$  代表总迭代次数,  $g$  代表当前迭代次数,  $\varepsilon^{(0)}$  代表  $\varepsilon$  的初始值.  $\varepsilon^{(0)}$  一般设置为初始种群中个体的最大冲突值.  $T_c$  控制迭代次数的变化区间,  $cp$  控制  $\varepsilon$  的下降速度,  $con$  是一个常数.

### 1.3 罚函数法

罚函数法是另一种被广泛使用的约束处理技术,其思想是在原适应度函数上增加一个惩罚项来构造一个新的适应度函数,将约束优化问题转化成全局优化问题.罚函数通常由公式(6)定义:

$$F_{pen}(\mathbf{x}) = f(\mathbf{x}) + \alpha \cdot \varphi(\mathbf{x}) \quad (6)$$

其中,  $F_{pen}(\mathbf{x})$ ,  $f(\mathbf{x})$  和  $\varphi(\mathbf{x})$  分别代表新的适应度函数、原适应度函数和约束冲突函数.  $\alpha$  是惩罚系数,用于调整适应度函数与冲突函数所占的比例.根据  $\alpha$  是否动态变化,罚函数法又可分为静态罚函数法和动态罚函数法.

### 1.4 差分进化

差分进化是一种基于种群的元启发式优化算法,主要步骤包括种群初始化、变异、交叉和选择.

一般而言,种群初始化操作使用均匀分布的随机向量生成初始种群.种群中第  $i$  个个体的第  $j$  维由公式(7)给出:

$$x_{i,j} = L_j + rand(0, 1) \cdot (U_j - L_j) \quad (7)$$

其中,  $rand(0, 1)$  是介于 0 和 1 之间均匀分布的随机数.EMSDE 采用的是佳点集初始化策略,以保证初始种群更加均匀的分布在定义域空间.

变异操作是差分进化算法的重要步骤之一,这里介绍了 3 种重要的变异形式<sup>[29]</sup>:

DE/rand/1

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (8)$$

DE/best/1

$$\mathbf{v}_i = \mathbf{x}_{best} + F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) \quad (9)$$

DE/current-to-pbest/1

$$\mathbf{v}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{best}^p - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) \quad (10)$$

其中,  $x_{r1}, x_{r2}, x_{r3}$  是从当前种群中随机选取的个体,  $x_i$  代表当前个体, 并且  $r1 \neq r2 \neq r3 \neq i$ ,  $x_{best}$  代表当前种群中最优的个体,  $x_{best}^p$  代表的是当前种群中前  $p\%$  最好个体中随机选择的一个,  $v_i$  是变异向量.  $F$  表示比例系数, 其决定着差分向量对变异向量  $v_i$  所占的比重.

在变异操作之后是交叉操作. 二项交叉是常见的交叉操作, 具体步骤如式(11)所示:

$$u_{i,j} = \begin{cases} v_{i,j}, & rand(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases} \quad (11)$$

其中,  $u_i$  是交叉向量,  $j_{rand}$  是  $\{1, \dots, D\}$  中的一个随机数. 二项交叉保证了交叉向量至少有一个维度是从变异向量继承而来.  $CR$  是交叉概率,  $CR$  越大, 交叉向量从变异向量继承的维度越多. 交叉概率是关乎算法收敛效果的关键因素.

最后一个步骤是选择操作. 选择操作将生成的后代与其父代进行比较, 然后选择较好的后代. 具体操作方法如公式(12)所示:

$$x_i = \begin{cases} u_i, & f(u_i) < f(x_i) \\ x_i, & \text{otherwise} \end{cases} \quad (12)$$

### 1.5 具有可选外部存档的自适应差分进化算法

具有可选外部存档的自适应差分进化算法 (Adaptive Differential Evolution with Optional External Archive, JADE) 是一种用于解决全局优化问题的 DE 变体<sup>[18]</sup>. JADE 使用 DE/current-to-pbest/1 变异策略, 但与公式(10)介绍的 DE/current-to-pbest/1 策略有一些区别, 即个体  $x_{r2}$  不是从当前种群中选取, 而是从归档集中选取的. 其中, 归档集保留了进化过程中失败的个体, 这是因为尽管被淘汰的个体适应值较差, 但进化方向可能正确, 因此保留适应值差的个体可以避免种群陷入局部最优区域. 此外, JADE 中的比例系数  $F$  和交叉概率  $CR$  是向进化过程中成功的个体学习而得出的, 这增强了进化成功的个体在后期进化中起到的作用. 通过学习进化过程中成功的参数, 可以优化进化后期的参数并提高种群收敛速度.  $S_F$  的平均值  $mean_L(S_F)$  和  $S_{CR}$  的平均值  $mean_A(S_{CR})$  分别由公式(13)和公式(14)计算:

$$mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \quad (13)$$

$$mean_A(S_{CR}) = \frac{\sum_{CR \in S_{CR}} CR}{|S_{CR}|} \quad (14)$$

其中,  $S_F$  代表上一代进化成功个体的比例系数  $F$  的集合,  $S_{CR}$  代表上一代进化成功个体的交叉概率  $CR$  的集合,  $|S_{CR}|$  代表集合  $S_{CR}$  中个体的数量. 公式(14)是集合  $S_{CR}$  中个体的算数平均值. 公式(13)采用的是莱默平均值. 莱默平均值不同于算数平均值, 它可以给成功的比例系数赋予更大的权重, 从而提高进化效率.

## 2 集成多策略的差分进化算法 (EMSDE)

本节首先提出了一种参数自适应策略, 即将罚函数归一化后的结果作为权重来调节 DE 的  $F$  和  $CR$ . 其次, 提出了一种结合  $\varepsilon$  约束和动态罚函数法的约束处理技术. 最后, 给出了集成多策略差分进化的算法流程.

### 2.1 罚函数归一化的参数自适应策略

DE 在求解优化问题的性能高度依赖于比例系数  $F$  和交叉概率  $CR$ . 为了增强 DE 在约束优化问题中的表现, 本文提出了一种新的参数自适应方法使之适用于约束优化问题, 称为罚函数归一化的参数自适应策略. EMSDE 的参数  $F$  和  $CR$  分别按照公式(15)和公式(16)进行柯西扰动和高斯扰动:

$$F_i = randc_i(mtF_i, 0.1) \quad (15)$$

$$CR_i = randc_i(mtCR_i, 0.1) \quad (16)$$

其中,  $F_i$  和  $CR_i$  分别是第  $i$  个个体的比例系数和交叉概率.  $randc(mtF_i, 0.1)$  代表以  $mtF_i$  为位置参数, 以 0.1 为尺度参数的柯西分布,  $randn(mtCR_i, 0.1)$  代表以  $mtCR_i$  为均值, 0.1 为标准差的高斯分布. 特别地, 当时  $F_i > 1$ , 令  $F_i = 1$ , 当  $F_i < 0$  时, 则重新由柯西分布(15)生成  $F_i$ ; 当  $CR_i > 1$  时, 令  $CR_i = 1$ , 当  $CR_i < 0$  时, 令  $CR_i = 0$ . 通过上述方法, 限制  $F$  和  $CR$  的值在 0~1 之间. 柯西分布和高斯分布可以避免  $F$  和  $CR$  的值固定不变, 有助于实现比例系数和交叉概率的多样化, 从而规避种群过早收敛. 其中,  $mtF_i$  和  $mtCR_i$  分别使用公式(17)和公式(18)进行计算:

$$mtF_i = (1-\lambda_i) \cdot \mu_F + \lambda_i \cdot randn(\mu_{CR}, 0.05) \quad (17)$$

$$mtCR_i = (1-\lambda_i) \cdot \mu_{CR} + \lambda_i \cdot randn(\mu_F, 2) \quad (18)$$

其中, 系数  $\lambda_i$  按照公式(19)生成:

$$\lambda_i = \frac{F_{pen}(x_i) - \min F_{pen}}{\max F_{pen} - \min F_{pen}} \quad (19)$$

$F_{pen}(x_i)$  代表当前个体  $x_i$  的罚函数值,  $\min F_{pen}$  和  $\max F_{pen}$  分别代表种群中最小的罚函数值和最大的罚函数值. 归一化方法可以使  $\lambda$  的值较为稳定地保持在一个特定的区间内, 便于不同量级的罚函数值进行比较和加权, 同时可以加快算法的收敛速度. 若个体  $x_i$  的罚函数值越小, 则表明  $x_i$  的可行度越高, 此时, 公式(17)中的高斯扰动项的系数  $\lambda_i$  就越小, 这种方法更倾向于保留可行度高的个体的参数  $\mu_F$ . 当  $\lambda_i$  的值为 0 时, 说明当前个体是种群中最好的个体, 则进化过程中会完全保留  $\mu_F$  供后面的个体学习. 反之, 若当前的个体较差, 那么  $\lambda_i$  的值越接近 1, 此时,  $\mu_F$  应当适当改变. 提出的策略在自适应参数时, 利用罚函数归一化全面考虑了约束优化中的适应度函数和约束函数, 更好地引导种群搜索可行区域中的最优解.

在公式(17)和公式(18)中, 系数  $\lambda_i, \mu_F$  和  $\mu_{CR}$  的初始值设为 0.5, 并分别用公式(20)和公式(21)生成:

$$\mu_F = (1-c) \cdot \mu_F + c \cdot mean_L(S_F) \quad (20)$$

$$\mu_{CR} = (1-c) \cdot \mu_{CR} + c \cdot mean_A(S_{CR}) \quad (21)$$

更好的控制参数能生成更有优势的个体, 因此, EMSDE 记录上一代成功的比例系数和交叉概率, 并且使用它们引导新的  $F$  和  $CR$  的生成.  $c$  的值通常设置为 0.1. 每个个体  $x_i$  都与其参数  $\{F_i, CR_i\}$  相关联, 参数  $\{F_i, CR_i\}$  具有记忆性, 通过公式(20)和公式(21)记忆历史进化中的成功参数.

### 2.2 两阶段变异策略

进化过程根据迭代次数是否超过总迭代次数的 50% 分为进化前期和进化后期, 变异方法如公式(22)所示:

$$v_i = \begin{cases} x_i + F_i \cdot (x_{best}^p - x_i) + F_i \cdot (x_{r1} - \bar{x}_{r2}), & \frac{g}{G} < 0.5 \\ x_i + F_i \cdot (x_{r1} - \bar{x}_{r2}), & \text{otherwise} \end{cases} \quad (22)$$

其中,  $\bar{x}_n$  表示从历史进化失败个体构成的存档集  $A$  中随机选择一个个体。

### 2.3 集成的约束处理技术

根据公式(4)和公式(5), 初始的  $\varepsilon^{(0)}$  是种群中最大约束值。当迭代次数的占比变成  $T_c$  时,  $\varepsilon$  的值设置为 0, 以获得具有最小约束冲突的解。

EMSDE 结合了  $\varepsilon$  约束和罚函数约束法, 首先使用了公式(4)和公式(5)的  $\varepsilon$  约束, 阈值根据进化代数不断减小到 0。在种群迭代前期, 相较于只使用适应度函数, 动态罚函数法既考虑了冲突函数又考虑了适应度函数, 避免了单一使用适应度函数的片面性, 可以使种群最快地找到满足约束条件的可行解。在种群迭代的后期, 阈值  $\varepsilon = 0$ , 此时, 罚函数的值等于适应度值, 也等同只使用了  $\varepsilon$  约束。这种方法可以使种群在迭代前期较多关注可行性的优化, 使得种群尽可能在可行区域内, 在迭代后期, 较多地关注适应值的优化, 提高了种群收敛到可行区域最优解的概率。集成的约束处理技术在原惩罚函数公式(6)上做出了改进, 具体方法如公式(23)所示:

$$F_{pen}(x) = f(x) + \varepsilon \cdot \varphi'(x) \quad (23)$$

其中,  $\varphi'(x)$  用公式(24)进行定义。

$$\varphi'(x) = \sum_{i=1}^p g_i^2(x) + \sum_{j=1}^q h_j^2(x) \quad (24)$$

### 2.4 EMSDE 算法的流程

- Step 1.** 初始化参数。适应值评估次数  $FV = 0$ ;
- Step 2.** 用佳点集初始化的方法生成初始种群  $P = \{x_1, \dots, x_n\}$ , 初始化归档集  $A = P$ ;
- Step 3.** 评估当前种群的适应值和冲突值;
- Step 4.** 初始化  $\varepsilon^{(0)} = \max\{\varphi(x_1), \dots, \varphi(x_n)\}$ ;
- Step 5.** While  $FV < MaxFEs$
- Step 6.** 根据公式(4)和公式(5)更新  $\varepsilon$ ;
- Step 7.** For  $x_i \in P$
- Step 8.** 根据公式(15)和公式(16)更新  $x_i$  的  $F_i$  和  $CR_i$ ;
- Step 9.** 根据公式(22)生成变异向量  $v_i$ ;
- Step 10.** 根据公式(11)生成交叉向量  $u_i$ ;
- Step 11.** If  $\varphi(u_i), \varphi(x_i) \leq \varepsilon$  or  $\varphi(u_i) = \varphi(x_i)$
- Step 12.** If  $F_{pen}(u_i) < F_{pen}(x_i)$
- Step 13.** 更新个体  $x_i \leftarrow u_i$ ;
- Step 14.** 更新归档集  $A \leftarrow A \cup \{x_i\}$ ;
- Step 15.** 更新成功参数集合  $S_p \leftarrow S_p \cup \{F_i\}, S_{CR} \leftarrow S_{CR} \cup \{CR_i\}$ ;
- Step 16.** Else
- Step 17.** 更新归档集  $A \leftarrow A \cup \{u_i\}$ ;
- Step 18.** End If
- Step 19.** Else If  $\varphi(u_i) < \varphi(x_i)$
- Step 20.** 更新个体  $x_i \leftarrow u_i$ ;
- 更新归档集  $A \leftarrow A \cup \{x_i\}$ ;
- Step 21.** Else If  $\varphi(u_i) > \varphi(x_i)$
- Step 22.** 更新归档集  $A \leftarrow A \cup \{u_i\}$
- Step 23.** End If
- Step 24.** End For
- Step 25.** If 归档集规模  $|A| > n$
- Step 26.** 从归档集  $A$  中随机删除  $|A| - n$  个个体;
- Step 27.** End If
- Step 28.** 根据公式(13)和公式(14)计算  $mean_L(S_p)$  和  $mean_A(S_{CR})$ ;

**Step 29.** 根据公式(20)和公式(21)来更新  $\mu_F$  和  $\mu_{CR}$ ;

**Step 30.** 根据公式(19)对权重  $\lambda_i$  进行归一化操作;

**Step 31.** 根据公式(17)和公式(18)计算  $mf_i$  和  $mtCR_i$ ;

**Step 32.** End While

## 3 实验

### 3.1 实验条件和参数设置

本文使用了 CEC2017 测试集<sup>[30]</sup>上 28 个约束优化函数作为基准问题, 目标是搜索满足约束条件并且适应值函数最小的可行解。该测试集包含了可分问题、不可分问题和旋转问题。CEC2017 测试集建议每个基准问题的最大适应值评估次数设置为  $20000D$  ( $D$  为问题的维度)。算法在达到最大适应值评估次数时终止。另外, 每个问题进行 25 次独立实验以获得统计结果, 包括适应值函数的平均值 Mean, 成功率 SR 和约束冲突的平均值 viol。其中 SR 表示 25 次独立实验中可行运行占总运行次数的比率, 可行运行表示当次运行至少找到了一个可行解。本实验检验了 EMSDE 算法在 10 维和 30 维基准问题上的性能。

实验中 EMSDE 的参数设置如下:  $T_c = 0.5, con = 6$ 。

本文代码运行的硬件参数平台为 12 代的 Intel(R) Core(TM) i5-12490F, 3.00GHz, 16GB。软件平台是 Windows 11 操作系统, 编译软件为 Matlab R2022b。

### 3.2 算法对比与结果分析

本文选取了一些有代表性的算法: LSHADE44 + IDE<sup>[31]</sup>、CAL-SHADE<sup>[32]</sup>、UDE<sup>[23]</sup>、PMode<sup>[33]</sup>、AMTPG-Jaya<sup>[34]</sup>、UDE-P (UDE with penalty)<sup>[35]</sup>、UDE-S (UDE with SOF)<sup>[35]</sup> 作为对比算法。表 1 和表 2 展示了对比实验结果, 每个问题的最优结果都用粗体突出表示。倒数第 2 行的符号“+ / = / -”分别代表 EMSDE 比其它的对比如算法“更好/相等/更差”。最后一行列出了从 Friedman 检验获得的 8 种算法的排名, rank 值越小的算法效果越好, 其中 Friedman 检验是一种利用秩判断总体分布是否存在显著差异的非参数检验方法。任意两种算法的比较规则是: 首先比较成功率 SR, 然后再比较平均约束冲突 viol, 最后对比平均适应值 Mean。

表 1 展示了 EMSDE 算法在 CEC2017 测试集 28 个 10 维函数与其他算法的对比结果。实验结果表明 EMSDE 算法在 C01, C02, C03, C04, C05, C08, C09, C10, C11, C12, C13, C14, C15, C16, C18, C20, C21, C22, C23, C24, C25 这 21 个问题中达到了 100% 的成功率。从 Friedman 检验的结果可以看出, EMSDE 算法以 rank 值为 3.13 位列第一。对比另外 7 种算法, EMSDE 算法在 C01, C02, C04, C08, C09, C10, C12, C15, C18, C20, C21, C23, C24, C25, C27 这 15 个函数上都达到了最好的效果。具体而言, EMSDE 有 15 个函数优于 LSHADE44 + IDE 算法, 有 4 个函数和 LSHADE44 + IDE 算法持平, 超过了测试集一半数量的函数都比 LSHADE44 + IDE 的效果好。EMSDE 有 17 个函数优于 CAL-SHADE 算法, 有 5 个函数与 CAL-SHADE 算法持平, 超过了测试集 60% 数量的函数都比其结果要好。相较于 PMode 算法、U-P 算法和 U-SOF 算法, EMSDE 算法分别有超过测试集 60% 数量的函数优于这几种算法。相较于 UDE 算法和 AMTPG-Jaya 算法, EMSDE 算法也有较大的优势。

表 1 在 CEC2017 测试集 10 维基准问题上 28 个函数的实验对比结果

Table 1		Experimental comparison results of 28 functions on the 10-dimensional benchmark problem of the CEC2017 test set								
函数名	problem	EMSDE	LSHADE44 + IDE	CAL-SHADE	UDE	PMODE	AMTPG-Jaya	U-P	U-SOF	
F1	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	8.64E-24	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	
	SR	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	100%	<b>100%</b>	<b>100%</b>	<b>100%</b>	
	viol	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	0.00E+00	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	
F2	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	6.49E-24	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	
	SR	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	100%	<b>100%</b>	<b>100%</b>	<b>100%</b>	
	viol	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	0.00E+00	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	
F3	Mean	8.05E+01	3.26E+05	1.10E+05	<b>7.31E+01</b>	5.95E+04	1.10E+05	7.37E+01	5.96E+04	
	SR	100%	100%	44%	<b>100%</b>	4%	100%	100%	72%	
	viol	0.00E+00	0.00E+00	6.34E-04	<b>0.00E+00</b>	1.66E-02	0.00E+00	0.00E+00	4.80E-03	
F4	Mean	<b>0.00E+00</b>	1.44E+01	3.87E+01	2.44E+01	8.14E+01	3.41E+01	2.42E+01	2.39E+01	
	SR	<b>100%</b>	100%	100%	100%	100%	100%	100%	100%	
	viol	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F5	Mean	3.19E-01	<b>0.00E+00</b>	9.57E-01	1.59E-01	6.08E-20	1.60E-01	<b>0.00E+00</b>	<b>0.00E+00</b>	
	SR	100%	<b>100%</b>	100%	100%	100%	100%	<b>100%</b>	<b>100%</b>	
	viol	0.00E+00	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	<b>0.00E+00</b>	<b>0.00E+00</b>	
F6	Mean	1.97E+02	8.08E+02	5.50E+02	8.67E+01	2.16E+02	2.03E+02	<b>7.59E+01</b>	1.92E+02	
	SR	0%	0%	96%	96%	16%	72%	<b>100%</b>	92%	
	viol	9.31E-01	3.77E-02	5.37E-03	6.80E-03	5.22E-01	4.17E-02	<b>0.00E+00</b>	4.05E-02	
F7	Mean	-1.65E+02	-3.40E+01	-4.87E+01	-7.14E+01	-5.62E+01	-5.21E+01	<b>-6.84E+01</b>	-5.24E+00	
	SR	8%	80%	68%	96%	4%	100%	<b>100%</b>	100%	
	viol	4.61E-03	3.19E-05	3.09E-03	1.22E-03	7.86E+01	0.00E+00	<b>0.00E+00</b>	0.00E+00	
F8	Mean	<b>-1.35E-03</b>	0.00E+00	<b>-1.35E-03</b>	<b>-1.35E-03</b>	<b>-1.35E-03</b>	<b>-1.35E-03</b>	-1.30E-03	-1.30E-03	
	SR	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	100%	100%	
	viol	<b>0.00E+00</b>	0.00E+00	<b>1.46E-05</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	0.00E+00	0.00E+00	
F9	Mean	<b>-4.98E-03</b>	0.00E+00	1.25E-01	<b>-4.98E-03</b>	<b>-4.98E-03</b>	<b>-4.98E-03</b>	-4.90E-03	-4.90E-03	
	SR	<b>100%</b>	100%	100%	<b>100%</b>	<b>100%</b>	<b>100%</b>	100%	100%	
	viol	<b>0.00E+00</b>	0.00E+00	4.00E-06	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	0.00E+00	0.00E+00	
F10	Mean	<b>-5.10E-04</b>	0.00E+00	<b>-5.10E-04</b>	<b>-5.10E-04</b>	<b>-5.10E-04</b>	<b>-5.10E-04</b>	-5.00E-04	-5.00E-04	
	SR	<b>100%</b>	100%	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	100%	100%	
	viol	<b>0.00E+00</b>	0.00E+00	<b>3.96E-06</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	0.00E+00	0.00E+00	
F11	Mean	-4.39E-02	0.00E+00	-1.56E-01	-6.00E+00	-3.56E+02	-1.54E-01	-6.00E+00	<b>-1.66E-01</b>	
	SR	100%	100%	100%	0%	0%	100%	0%	<b>100%</b>	
	viol	0.00E+00	0.00E+00	2.00E-05	3.00E-03	2.38E+02	0.00E+00	2.90E-03	<b>0.00E+00</b>	
F12	Mean	<b>3.99E+00</b>	<b>3.99E+00</b>	<b>3.99E+00</b>	<b>3.99E+00</b>	4.84E+00	<b>3.99E+00</b>	4.74E+00	<b>3.99E+00</b>	
	SR	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	100%	<b>100%</b>	100%	<b>100%</b>	
	viol	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	0.00E+00	<b>0.00E+00</b>	0.00E+00	<b>0.00E+00</b>	
F13	Mean	3.19E-01	<b>0.00E+00</b>	1.12E+00	3.19E-01	1.59E-01	3.19E-01	5.63E+00	2.82E+00	
	SR	100%	<b>100%</b>	100%	100%	100%	100%	100%	100%	
	viol	0.00E+00	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F14	Mean	2.50E+00	3.00E+00	2.64E+00	<b>2.47E+00</b>	3.45E+00	2.50E+00	2.48E+00	3.35E+00	
	SR	100%	100%	100%	<b>100%</b>	48%	100%	100%	88%	
	viol	0.00E+00	0.00E+00	0.00E+00	<b>0.00E+00</b>	5.32E-05	0.00E+00	0.00E+00	9.40E-06	
F15	Mean	<b>5.50E+00</b>	1.13E+01	1.62E+01	6.63E+00	1.22E+01	1.09E+01	7.5084	16.054	
	SR	<b>100%</b>	100.00%	68.00%	100.00%	40.00%	100.00%	96.00%	64.00%	
	viol	<b>0.00E+00</b>	0.00E+00	1.51E+01	9.56E-05	3.24E-04	0.00E+00	3.69E-05	0.0002	
F16	Mean	7.67E+00	4.04E+01	6.54E+01	6.63E+00	6.16E+01	4.51E+01	<b>6.28E+00</b>	6.43E+01	
	SR	100%	100%	60%	100%	12%	100%	<b>100%</b>	80%	
	viol	0.00E+00	0.00E+00	1.30E-02	0.00E+00	1.99E-04	0.00E+00	<b>0.00E+00</b>	2.96E-05	
F17	Mean	9.82E-01	1.00E+00	1.00E+00	9.49E-01	9.81E-01	<b>1.18E+00</b>	9.82E-01	9.51E-01	
	SR	0.00%	0%	0%	0%	0%	<b>0%</b>	0%	0%	
	viol	1.05E+01	5.22E+00	5.54E+00	5.38E+00	5.51E+00	<b>4.86E+00</b>	5.46E+00	5.46E+00	
F18	Mean	<b>4.10E+01</b>	3.17E+03	1.35E+03	9.96E+01	4.10E+01	6.45E+02	6.16E+02	7.49E+02	
	SR	<b>100%</b>	0%	0%	0%	4%	0%	0%	0%	
	viol	<b>0.00E+00</b>	8.23E+06	1.40E+08	8.29E+03	2.93E+00	2.87E+02	1.07E+06	3.10E+06	

F19	Mean	0.00E+00	0.00E+00	<b>3.44E-06</b>	0.00E+00	0.00E+00	1.52E-06	0.00E+00	0.00E+00
	SR	0.00%	0%	<b>0%</b>	0%	0%	0%	0%	0%
	viol	1.33E+04	6.63E+03	<b>4.42E+03</b>	6.63E+03	6.63E+03	6.63E+03	6.63E+03	6.63E+03
F20	Mean	<b>2.55E-01</b>	4.16E-01	8.69E-01	1.50E+00	1.82E+00	7.92E-01	1.26E+00	1.54E+00
	SR	<b>100%</b>	100%	100%	100%	100%	100%	100%	100%
	viol	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F21	Mean	<b>3.99E+00</b>	<b>3.99E+00</b>	6.86E+00	4.41E+00	5.46E+00	<b>3.99E+00</b>	5.59E+00	<b>3.99E+00</b>
	SR	<b>100%</b>	<b>100%</b>	100%	100%	100%	<b>100%</b>	100%	<b>100%</b>
	viol	<b>0.00E+00</b>	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	<b>0.00E+00</b>	0.00E+00	<b>0.00E+00</b>
F22	Mean	2.81E+00	1.60E-01	6.67E+00	5.31E+00	<b>2.88E-19</b>	2.98E+00	5.31E+00	3.93E+00
	SR	100.00%	100%	100%	100%	<b>100%</b>	100%	100%	100%
	viol	0.00E+00	0.00E+00	0.00E+00	0.00E+00	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00
F23	Mean	<b>2.41E+00</b>	3.02E+00	2.53E+00	2.47E+00	3.30E+00	2.94E+00	<b>2.41E+00</b>	3.06E+00
	SR	<b>100%</b>	100%	100%	100%	52%	100%	<b>100%</b>	100%
	viol	<b>0.00E+00</b>	0.00E+00	1.20E-05	0.00E+00	6.83E-05	0.00E+00	<b>0.00E+00</b>	0.00E+00
F24	Mean	<b>2.73E+00</b>	8.77E+00	1.57E+01	5.50E+00	1.38E+01	1.10E+01	5.62E+00	1.39E+01
	SR	<b>100%</b>	100%	40%	100%	20%	100%	100%	96%
	viol	<b>0.00E+00</b>	0.00E+00	1.01E-01	0.00E+00	4.23E-04	0.00E+00	0.00E+00	1.04E-05
F25	Mean	<b>5.78E+00</b>	3.77E+01	6.11E+01	6.28E+00	6.26E+01	4.96E+01	6.53E+00	6.47E+01
	SR	<b>100%</b>	100%	60%	100%	4%	100%	100%	84%
	viol	<b>0.00E+00</b>	0.00E+00	1.22E-02	0.00E+00	2.32E-04	0.00E+00	0.00E+00	2.51E-05
F26	Mean	1.02E+00	9.37E-01	1.00E+00	9.70E-01	9.09E-01	<b>1.24E+00</b>	8.97E-01	8.96E-01
	SR	0.00%	0%	0%	0%	0%	<b>0%</b>	0%	0%
	viol	1.08E+01	5.05E+00	5.50E+00	5.42E+00	5.47E+00	<b>4.84E+00</b>	5.38E+00	5.46E+00
F27	Mean	<b>1.12E+02</b>	7.94E+03	2.44E+03	1.36E+02	2.33E+01	2.16E+03	8.58E+02	7.44E+02
	SR	<b>96%</b>	0%	0%	0%	0%	0%	0%	0%
	viol	<b>3.91E-02</b>	4.05E+07	1.26E+07	1.05E+04	2.48E-01	1.25E+03	2.18E+07	4.30E+06
F28	Mean	5.94E+00	1.08E+01	<b>3.11E+01</b>	4.90E+00	1.62E+00	1.67E+01	8.69E+00	1.66E+00
	SR	0.00%	0%	<b>0%</b>	0%	0%	0%	0%	0%
	viol	1.33E+04	6.64E+03	<b>4.44E+03</b>	6.64E+03	6.64E+03	6.64E+03	6.64E+03	6.63E+03
+/-/-	-	15/4/9	17/5/6	12/6/10	17/3/8	12/8/8	16/3/9	16/4/8	
rank	<b>3.13</b>	5.07	6.18	3.21	4.68	4.80	4.00	4.93	

表2 在CEC2017测试集30维基准问题上28个函数的实验对比结果

Table 2 Experimental comparison results of 28 functions on the 30-dimensional benchmark problem of the CEC2017 test set

函数名	Problem	EMSDE	LSHADE44 + IDE	CAL-SHADE	UDE	PMODE	AMTPG-Jaya	U-P	U-SOF
F1	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	5.45E-29	1.02E+02	6.31E-27	7.34E-29	7.34E-29
	SR	<b>100%</b>	<b>100%</b>	<b>100%</b>	100%	100%	100%	100%	100%
	viol	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	5.55E-29	8.79E+01	6.28E-27	7.39E-29	7.39E-29
	SR	<b>100%</b>	<b>100%</b>	<b>100%</b>	100%	100%	100%	100%	100%
	viol	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	Mean	5.04E+03	6.70E+06	1.30E+06	7.46E+01	5.25E+05	3.37E+05	<b>7.33E+01</b>	<b>7.33E+01</b>
	SR	100%	100%	32%	100%	8%	100%	<b>100%</b>	<b>100%</b>
	viol	0.00E+00	0.00E+00	2.43E-02	0.00E+00	2.98E-02	0.00E+00	<b>0.00E+00</b>	<b>0.00E+00</b>
F4	Mean	1.42E+01	<b>1.39E+01</b>	1.16E+02	7.82E+01	3.26E+02	1.20E+02	8.24E+01	8.24E+01
	SR	100%	<b>100%</b>	100%	100%	100%	100%	100%	100%
	viol	0.00E+00	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F5	Mean	1.59E-01	<b>0.00E+00</b>	7.97E-01	1.59E-01	1.77E+02	4.17E-27	2.32E-17	2.32E-17
	SR	100%	<b>100%</b>	100%	100%	100%	100%	100%	100%
	viol	0.00E+00	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	Mean	2.05E+03	5.53E+03	3.75E+03	<b>3.03E+02</b>	2.76E+03	2.18E+03	3.04E+02	3.04E+02
	SR	0%	0%	100%	<b>100%</b>	0%	0%	100%	100%
	viol	8.43E-01	2.57E-02	1.16E-05	<b>0.00E+00</b>	3.07E+00	2.04E-03	0.00E+00	0.00E+00
F7	Mean	-3.08E+02	-8.11E+01	-24.1162	-5.64E+02	-6.32E+01	-1.07E+02	<b>-5.98E+02</b>	<b>-5.98E+02</b>
	SR	24%	96%	52%	100%	0%	100%	<b>100%</b>	<b>100%</b>
	viol	1.86E-03	4.06E-06	3.56E-03	0.00E+00	1.38E+03	0.00E+00	<b>0.00E+00</b>	<b>0.00E+00</b>

F8	Mean	-2.77E-04	-2.63E-04	<b>-2.84E-04</b>	-2.81E-04	9.09E+00	<b>-2.84E-04</b>	-2.80E-04	-2.80E-04
	SR	100%	100%	<b>100%</b>	100%	0%	<b>100%</b>	100%	100%
	viol	0.00E+00	0.00E+00	<b>0.00E+00</b>	0.00E+00	2.64E+02	<b>0.00E+00</b>	0.00E+00	0.00E+00
F9	Mean	-2.67E-03	-2.67E-03	2.34E-02	<b>-2.67E-03</b>	1.22E+00	<b>-2.67E-03</b>	<b>-2.67E-03</b>	<b>-2.67E-03</b>
	SR	100%	100.00%	96.00%	<b>100.00%</b>	60.00%	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>
	viol	0.00E+00	0.00E+00	1.07E+06	<b>0.00E+00</b>	4.80E-02	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
F10	Mean	<b>-1.03E-04</b>	-9.78E-05	<b>-1.03E-04</b>	-1.01E-04	5.79E+00	<b>-1.03E-04</b>	-1.00E-04	-1.00E-04
	SR	<b>100%</b>	100%	<b>100%</b>	100%	0%	<b>100%</b>	100%	100%
	viol	<b>0.00E+00</b>	0.00E+00	<b>6.60E-06</b>	0.00E+00	1.04E+05	<b>0.00E+00</b>	0.00E+00	0.00E+00
F11	Mean	-1.05E+01	<b>-8.65E-01</b>	7.81E-01	-2.70E+01	-1.58E+03	-7.92E-01	-2.83E+01	-1.22E+00
	SR	16%	<b>100%</b>	100%	0%	0%	100%	0%	64%
	viol	6.91E-02	<b>0.00E+00</b>	2.40E-05	1.25E-02	8.30E+02	0.00E+00	2.07E-02	4.43E-11
F12	Mean	1.08E+01	6.07E+00	1.42E+01	1.65E+01	9.72E+02	<b>5.38E+00</b>	1.87E+01	1.57E+01
	SR	100%	100%	100%	100%	0%	<b>100%</b>	100%	100%
	viol	0.00E+00	0.00E+00	1.60E-07	0.00E+00	4.47E+02	<b>0.00E+00</b>	0.00E+00	0.00E+00
F13	Mean	<b>9.01E+00</b>	2.60E+01	3.53E+03	3.60E+01	1.01E+07	1.49E+02	8.15E+01	1.64E+02
	SR	<b>100%</b>	100%	100%	100%	0%	100%	100%	100%
	viol	<b>0.00E+00</b>	0.00E+00	4.00E-08	0.00E+00	3.58E+02	0.00E+00	0.00E+00	0.00E+00
F14	Mean	<b>1.45E+00</b>	1.91E+00	1.55E+00	1.56E+00	1.70E+01	1.47E+00	1.53E+00	1.88E+00
	SR	<b>100%</b>	100%	100%	100%	0%	100%	100%	100%
	viol	<b>0.00E+00</b>	0.00E+00	2.00E-06	0.00E+00	2.54E+03	0.00E+00	0.00E+00	0.00E+00
F15	Mean	<b>8.64E+00</b>	1.29E+01	2.41E+01	9.02E+00	1.82E+01	1.17E+01	9.14E+00	2.11E+01
	SR	<b>100%</b>	100%	48%	88%	48%	100%	88%	100%
	viol	<b>0.00E+00</b>	0.00E+00	2.01E+02	0.00E+00	1.85E-04	0.00E+00	2.62E-05	0.00E+00
F16	Mean	9.55E+00	1.43E+02	2.12E+02	<b>7.92E+00</b>	2.11E+02	1.63E+02	8.42E+00	2.01E+02
	SR	100%	100%	16%	<b>100%</b>	16%	100%	100%	92%
	viol	0.00E+00	0.00E+00	8.83E-03	<b>0.00E+00</b>	1.85E-04	0.00E+00	0.00E+00	5.59E-06
F17	Mean	1.02E+00	<b>1.01E+00</b>	1.03E+00	1.03E+00	1.18E+00	1.02E+00	1.02E+00	1.02E+00
	SR	0%	<b>0%</b>	0%	0%	0%	0%	0%	0%
	viol	3.10E+01	<b>1.55E+01</b>	1.73E+01	1.55E+01	3.07E+02	1.55E+01	1.55E+01	1.55E+01
F18	Mean	<b>3.72E+01</b>	4.01E+03	6.71E+03	4.55E+02	9.55E+02	1.63E+03	3.57E+03	8.28E+02
	SR	<b>100%</b>	0%	0%	0%	0%	0%	0%	0%
	viol	<b>0.00E+00</b>	1.33E+06	8.76E+08	3.13E+05	9.84E+05	879.52	3.36E+07	1.03E+07
F19	Mean	0.00E+00	0.00E+00	<b>3.09E-01</b>	4.75E-01	5.60E+01	6.00E-06	3.66E+00	0.00E+00
	SR	0%	0%	<b>0%</b>	0%	0%	0%	0%	0%
	viol	4.27E+04	2.14E+04	<b>1.43E+04</b>	2.14E+04	2.15E+04	2.14E+04	2.14E+04	2.14E+04
F20	Mean	<b>1.37E+00</b>	2.25E+00	2.11E+00	5.17E+00	9.23E+00	5.85E+00	4.58E+00	4.59E+00
	SR	<b>100%</b>	100%	100%	100%	100%	100%	100%	100%
	viol	<b>0.00E+00</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F21	Mean	1.27E+01	2.75E+01	1.33E+01	1.48E+01	7.69E+02	<b>6.68E+00</b>	1.19E+01	1.40E+01
	SR	100%	100%	100%	100%	0%	<b>100%</b>	92%	100%
	viol	0.00E+00	0.00E+00	4.00E-08	0.00E+00	3.46E+02	<b>0.00E+00</b>	1.48E-12	0.00E+00
F22	Mean	3.51E+04	8.44E+02	3.42E+04	1.21E+02	4.80E+07	1.28E+04	<b>9.41E+01</b>	1.80E+03
	SR	36%	100%	72%	100%	0%	72%	<b>100%</b>	96%
	viol	4.05E+00	0.00E+00	2.76E-01	0.00E+00	7.67E+02	4.00E+00	<b>0.00E+00</b>	6.20E-04
F23	Mean	<b>1.41E+00</b>	1.86E+00	1.58E+00	1.46E+00	2.05E+01	1.48E+00	1.43E+00	1.87E+00
	SR	<b>100%</b>	100%	84%	100%	0%	100%	100%	100%
	viol	<b>0.00E+00</b>	0.00E+00	1.60E-04	0.00E+00	1.38E+04	0.00E+00	0.00E+00	0.00E+00
F24	Mean	<b>8.01E+00</b>	1.39E+01	2.09E+01	8.51E+00	1.89E+01	1.21E+01	8.39E+00	1.88E+01
	SR	<b>100%</b>	100%	20%	100%	36%	100%	100%	100%
	viol	<b>0.00E+00</b>	0.00E+00	1.47E+01	0.00E+00	1.47E+02	0.00E+00	0.00E+00	0.00E+00
F25	Mean	4.00E+01	1.41E+02	2.07E+02	<b>1.32E+01</b>	2.36E+02	1.68E+02	1.60E+01	1.84E+02
	SR	100%	100%	32%	<b>100%</b>	28%	100%	100%	96%
	viol	0.00E+00	0.00E+00	2.34E-02	<b>0.00E+00</b>	2.76E+02	0.00E+00	0.00E+00	3.12E-06
F26	Mean	1.02E+00	1.01E+00	<b>1.04E+00</b>	1.03E+00	1.65E+00	1.02E+00	1.02E+00	1.02E+00
	SR	0%	100%	<b>0%</b>	0%	0%	0%	0%	0%
	viol	3.10E+01	0.00E+00	<b>2.15E+01</b>	1.55E+01	1.26E+03	1.55E+01	1.55E+01	1.55E+01

	Mean	<b>8.53E+01</b>	5.23E+04	3.14E+03	1.00E+03	3.54E+03	5.91E+02	1.19E+04	1.70E+03
F27	SR	<b>44%</b>	0%	0%	0%	0%	0%	0%	0%
	viol	<b>6.30E+00</b>	1.08E+08	1.52E+07	6.04E+06	9.44E+05	1.77E+03	3.41E+08	8.18E+06
	Mean	1.37E+02	1.53E+02	<b>1.22E+02</b>	6.62E+01	9.64E+01	1.29E+02	6.43E+01	5.65E+01
F28	SR	0%	0%	<b>0%</b>	0%	0%	0%	0%	0%
	viol	4.30E+04	2.15E+04	<b>1.43E+04</b>	2.14E+04	2.15E+04	2.15E+04	2.14E+04	2.14E+04
+/-/-		-	14/3/10	15/3/9	14/2/12	24/0/4	15/2/11	15/1/12	16/1/11
rank		<b>2.93</b>	4.52	5.61	3.63	7.07	4.25	3.61	4.39

表2展示了EMSDE算法与其他几种算法在30维下的对比结果,EMSDE在C01,C02,C03,C04,C05,C08,C09,C10,C12,C13,C14,C15,C16,C18,C20,C21,C23,C24,C25这19个函数上实现了100%的成功率.最后一行列出了Friedman检测结果,EMSDE以rank值为2.93位居第一,验证了EMSDE算法比其他7种算法具有更好的性能.EMSDE算法至少有测试集中50%的函数比另外7种算法的效果更好.特别是对比PMODE算法,有超过测试集85%的函数比PMODE好.同样地,相较于其它7种算法,EMSDE算法在C01,C02,C09,C10,C13,C14,C15,C18,C20,C23,C24,C27这12个函数上取得了最好的结果.这些函数既包含了可分离约束问题、不可分离问题也包含了旋转问题.实验证明了

EMSDE算法对于CEC2017测试集不同类型的问题具有强有力的竞争力.

## 4 讨论

本节首先讨论了EMSDE算法的收敛性,并与JADE约束优化算法做了适应值收敛性分析.其次,本文做了消融实验,测试了EMSDE算法集成策略的优越性,并与几种仅改变单一组件的算法进行了对比.

### 4.1 收敛性分析

图1展示了EMSDE算法与JADE变异策略的约束优化算法在10维和30维的结果.其中可分函数以F16为代表,不

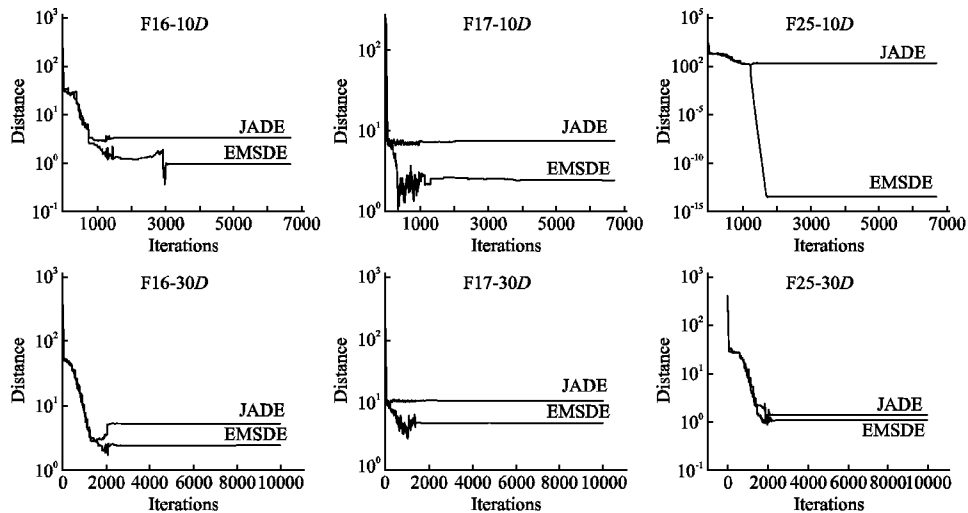


图1 EMSDE和JADE在10维和30维函数F16、F17和F25上的收敛性曲线对比  
Fig. 1 Comparison of the convergence curves of EMSDE and JADE on 10-dimensional and 30-dimensional functions F16, F17 and F25

可分函数以F17为代表,旋转函数以F25为代表.图1反映了种群个体与质心的最大距离( $d_{max}$ ).

在图1中,从F16、F17和F25的10维变化曲线可以看出,JADE和EMSDE在相近的迭代时期时 $d_{max}$ 趋于平稳,并且平稳后EMSDE的 $d_{max}$ 小于JADE.特别地,在F25上,EMSDE的 $d_{max}$ 可以收敛到 $3.72 \times 10^{-14}$ ,而JADE只收敛到1.97.说明在10维问题上,EMSDE算法的收敛性更好.同样,在30维的F16、F17和F25上,EMSDE的 $d_{max}$ 也小于JADE的 $d_{max}$ .综上所述,EMSDE算法与使用JADE变异组件的约束优化算法相比,EMSDE算法种群收敛性更好.

### 4.2 消融实验

为了突出本文集成策略的优势,本文展示了10维和30维问题的消融实验.

消融实验将EMSDE算法(EMSDE)分别与使用JADE策略的约束方法(JADE)、使用Takahama和Sakai提出的 $\epsilon$ 约束优化方法(EC)和单变异策略方法(SMS)进行对比.在使用JADE策略的约束方法中,保持其约束方法与EMSDE的约束方法相同,但是比例系数 $F$ 和交叉概率 $CR$ 的更新策略采用的是公式(13)、(14)、(20)、(21).单变异方法仅使用公式(22)中迭代超过后的变异策略.以上这些对比实验,每次仅改变EMSDE混合策略中的一种策略,可以检测EMSDE混合策略的有效性.

图2展示了EMSDE、JADE、EC和SMS的成功率柱状图.在图2中,在10维问题中,EMSDE在F3、F11、F15和F18上的成功率均为100%,而其他算法在这些函数上的成功率都未达到100%.可以看出,EMSDE算法具有比其他算法更

好的性能. 在 30 维问题的 F11 函数上, 仅有 EMSDE 算法的成功率不为 0, 其他算法的成功率都为 0. 并且在 30 维的 F3,

F15, F18, 只有 EMSDE 能达到 100% 的成功率.

对于 10 维问题, EMSDE 算法在其中的 21 个函数中都达

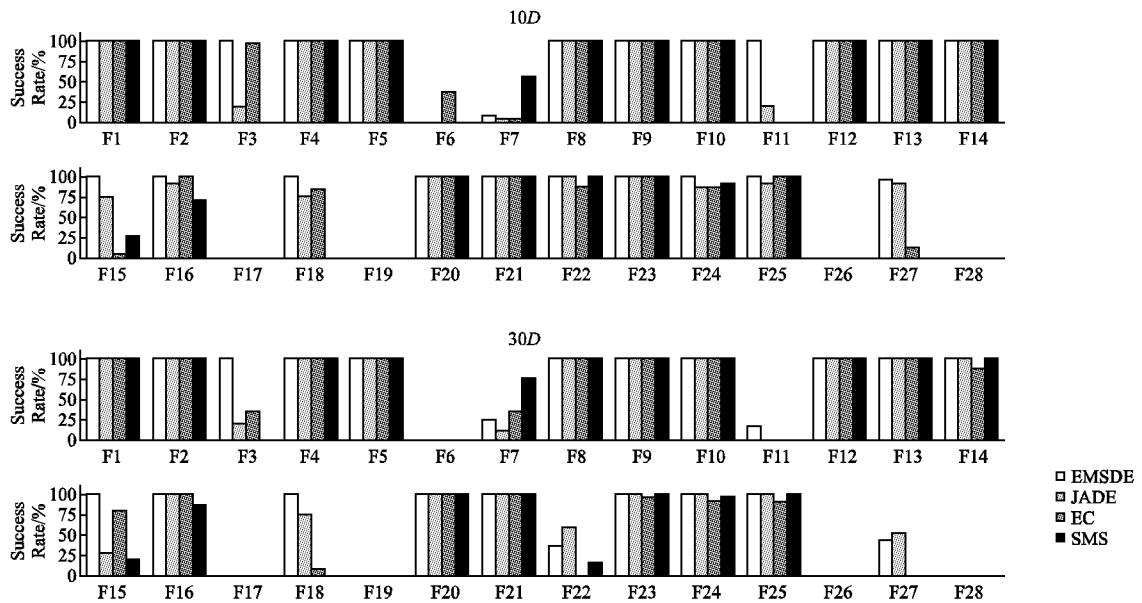


图 2 在 CEC2017 测试集 10 维和 30 维基准函数上消融实验的成功率柱状图  
Fig. 2 Histogram of the success rate of the ablation experiment on the 10-dimensional and 30-dimensional benchmark functions of the CEC2017 test set

到了 100% 的成功率, JADE 只在其中 14 个函数上达到了 100% 的成功率, 而 EC 和 SMS 只在其中 15 个函数上达到了 100% 的成功率. 由此可见, 在 10 维问题上 EMSDE 达到 100% 成功率的函数数量最多.

对于 30 维问题, EMSDE 在其中的 19 个函数上达到了 100% 的成功率, JADE 在其中 16 个函数上达到了 100% 的成功率, EC 和 SMS 分别在其中 12 个和 14 个函数上达到了 100% 的成功率. EMSDE 算法同样比另外几种算法的效果更好.

表 3 在 CEC2017 测试集 10 维基准函数上消融实验的 Friedman 检验

Table 3 Friedman test of ablation experiments on the 10-dimensional benchmark function of the CEC2017 test set

对比实验	rank
EMSDE	2.13
JADE	2.43
只用 $\epsilon$ 约束 (EC)	2.84
单变异策略 (SMS)	2.61

表 4 在 CEC2017 测试集 30 维基准函数上消融实验的 Friedman 检验

Table 4 Friedman test of ablation experiments on the 30-dimensional benchmark function of the CEC2017 test set

对比实验	rank
EMSDE	1.96
JADE	2.18
只用 $\epsilon$ 约束 (EC)	3.14
单变异策略 (SMS)	2.71

表 3 和表 4 分别展示了 EMSDE 算法与几种控制算子的算法对比结果. 使用 Friedman 检验方法得出的在 CEC2017 测

试集 10 维和 30 维的 28 个基准函数上适应度均值的 rank 值. 在 10 维和 30 维的条件下, EMSDE 算法的效果最优, 这表明了 EMSDE 算法的集成策略非常有效.

综上所述, 对于 10 维问题和 30 维问题, EMSDE 算法都比另外 3 种仅使用单一策略的算法结果要好, 实验结果验证了集成多策略的优越性.

### 5 结论

本文提出了求解复杂约束优化问题集成多策略的差分进化算法 EMSDE. EMSDE 将罚函数归一化用于调节差分进化算法中 F 和 CR 的权重, 并结合历史存档策略方法用于自适应生成参数. 最后, 将种群分为前后两个阶段进行变异操作, 种群在迭代的前 50% 阶段采用带有外部存档集的 DE/current-to-pbest/1 的变异策略, 在迭代的后 50% 阶段采用具有外部存档集的另一种新的变异策略. 两阶段变异策略有利于种群跳出局部最优区域, 提高种群多样性. 在约束处理技术方面, 将  $\epsilon$  作为罚函数中冲突项的系数, 根据进化代数动态地调整惩罚函数. 通过与经典的几种约束算法进行仿真实验对比, 本文提出的 EMSDE 算法比其他算法具有更好的性能. 此外, 消融实验的结果表明 EMSDE 算法的集成策略比单一组件具有更好的收敛性和性能.

### References:

[ 1 ] Ramamoorthy S, Ravikumar G, Saravana Balaji B, et al. Mcamo: multi constraint aware multi-objective resource scheduling optimization technique for cloud infrastructure services[J]. Journal of Ambient Intelligence Humanized Computing, 2021, 12(6): 5909-5916.

[ 2 ] Peykani P, Nouri M, Eshghi F, et al. A novel mathematical approach for fuzzy multi-period multi-objective portfolio optimization problem under uncertain environment and practical constraints[J]. Jour-

- nal of Fuzzy Extension Applications, 2021, 2(3):191-203.
- [3] Jiang D, Wang Y, Lv Z, et al. An energy-efficient networking approach in cloud services for iiot networks [J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(5):928-941.
- [4] ZHANG Z T, SHENG C Z, YU B G, et al. Multi-sensor vehicle positioning algorithm based on constrained optimization [J]. *Radio Engineering*, 2022, 52(10):1781-1787.
- [5] Zu L, Xu H, Wang H, et al. Design and analysis of filament-wound composite pressure vessels based on non-geodesic winding [J]. *Composite Structures*, 2019, 207:41-52, doi: 10.1016/j.compstruct.2018.09.007.
- [6] LIU G G, HANG Y F, WANG X, et al. Hybrid discrete particle swarm optimization algorithm for x-architecture steiner minimal tree construction with slew constraints [J]. *Chinese Journal of Computers*, 2021, 44(12):2542-2559.
- [7] Xu P, Luo W, Lin X, et al. Constraint-objective cooperative coevolution for large-scale constrained optimization [J]. *ACM Transactions on Evolutionary Learning Optimization*, 2021, 1(3):1-26.
- [8] Zou D, Gong D. Differential evolution based on migrating variables for the combined heat and power dynamic economic dispatch [J]. *Energy*, 2022, 238:121664, doi:10.1016/j.energy.2021.121664.
- [9] Zhao F, Zhao L, Wang L, et al. An ensemble discrete differential evolution for the distributed blocking flowshop scheduling with minimizing makespan criterion [J]. *Expert Systems with Applications*, 2020, 160:113678, doi:10.1016/j.eswa.2020.113678.
- [10] Ali M Z, Awad N H, Suganthan P N, et al. An adaptive multipopulation differential evolution with dynamic population reduction [J]. *IEEE Transactions on Cybernetics*, 2016, 47(9):2768-2779.
- [11] Brest J, Zamuda A, Fister I, et al. Self-adaptive differential evolution algorithm with a small and varying population size [C]//IEEE Congress on Evolutionary Computation, 2012:1-8.
- [12] Storn R, Price K. Minimizing the real functions of the icec'96 contest by differential evolution [C]//Proceedings of IEEE International Conference on Evolutionary Computation, 1996:842-844.
- [13] Ronkkonen J, Kukkonen S, Price K V. Real-parameter optimization with differential evolution [C]//IEEE Congress on Evolutionary Computation, 2005:506-513.
- [14] Gämperle R, Müller S D, Koumoutsakos P. A parameter study for differential evolution [J]. *Advances in Intelligent Systems, Fuzzy-Systems, Evolutionary Computation*, 2002, 10(10):293-298.
- [15] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters [J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1):55-66.
- [16] Brest J, Greiner S, Boskovic B, et al. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems [J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(6):646-657.
- [17] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2008, 13(2):398-417.
- [18] Zhang J, Sanderson A C. Jade: adaptive differential evolution with optional external archive [J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5):945-958.
- [19] Xu P, Luo W, Lin X, et al. Evolutionary continuous constrained optimization using random direction repair [J]. *Information Sciences*, 2021, 566:80-102, doi:10.1016/j.ins.2021.02.055.
- [20] Deb K. An efficient constraint handling method for genetic algorithms [J]. *Computer Methods in Applied Mechanics Engineering*, 2000, 186(2-4):311-338.
- [21] LI Z Y, HUANG T, CHEN S M, et al. Overview of constrained optimization evolution algorithms [J]. *Journal of Software*, 2017, 28(6):1529-1546.
- [22] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2000, 4(3):284-294.
- [23] Trivedi A, Sanyal K, Verma P, et al. A unified differential evolution algorithm for constrained optimization problems [C]//IEEE Congress on Evolutionary Computation (CEC), 2017:1231-1238.
- [24] Takahama T, Sakai S. Constrained optimization by the  $\epsilon$  constrained differential evolution with gradient-based mutation and feasible elites [C]//IEEE International Conference on Evolutionary Computation, 2006:1-8.
- [25] Takahama T, Sakai S. Efficient constrained optimization by the  $\epsilon$  constrained differential evolution with rough approximation using kernel regression [C]//IEEE Congress on Evolutionary Computation, 2013:1334-1341.
- [26] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2000, 4(3):284-294.
- [27] Homaifar A, Qi C X, Lai S H. Constrained optimization via genetic algorithms [J]. *Simulation Trans of the Society for Modeling & Simulation International*, 1994, 62(4):242-253.
- [28] Joines J A, Houck C R. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's [C]//International Conference on Evolutionary Computation, 1994:579-584.
- [29] LI L S, WENG Q Q. Research on adaptive differential evolution algorithm for test data generation [J]. *Journal of Chinese Computer Systems*, 2018, 39(2):292-296.
- [30] Wu G, Mallipeddi R, Suganthan P N. Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization [C]//National University of Defense Technology, Changsha, Hunan, PR China Kyungpook National University, Daegu, South Korea Nanyang Technological University, Singapore, 2017.
- [31] Tvrdík J, Polůková R. A simple framework for constrained problems with application of l-shade44 and ide [C]//IEEE Congress on Evolutionary Computation (CEC), 2017:1436-1443.
- [32] Zamuda A. Adaptive constraint handling and success history differential evolution for cec 2017 constrained real-parameter optimization [C]//IEEE Congress on Evolutionary Computation (CEC), 2017:2443-2450.
- [33] Huang W, Xu T, Li K, et al. Multiobjective differential evolution enhanced with principle component analysis for constrained optimization [J]. *Swarm Evolutionary Computation*, 2019, 50:100571, doi:10.1016/j.swevo.2019.100571.
- [34] Rao R V, Keesari H S, Oclon P, et al. An adaptive multi-team perturbation-guiding jaya algorithm for optimization and its applications [J]. *Engineering with Computers*, 2020, 36(1):391-419.
- [35] Trivedi A, Biswas N, Chakroborty S, et al. Extending unified differential evolution with a new ensemble of constraint handling techniques [C]//IEEE Symposium Series on Computational Intelligence (SSCI), 2017:1-8.
- 附中文参考文献:**
- [4] 张子腾, 盛传贞, 蔚保国, 等. 基于约束优化的多传感器车载定位方法 [J]. *无线电工程*, 2022, 52(10):1781-1787.
- [6] 刘耿耿, 黄逸飞, 王鑫, 等. 基于混合离散粒子群优化的 Slew 约束 X 结构 Steiner 最小树算法 [J]. *计算机学报*, 2021, 44(12):2542-2559.
- [21] 李智勇, 黄滔, 陈少森, 等. 约束优化进化算法综述 [J]. *软件学报*, 2017, 28(6):1529-1546.
- [29] 李龙澍, 翁晴晴. 应用自适应差分进化算法优化测试数据研究 [J]. *小型微型计算机系统*, 2018, 39(2):292-296.