



计算机科学

COMPUTER SCIENCE

基于时空间联合去噪的改进差分进化算法

王彬, 张鑫雨, 金海燕

引用本文

王彬, 张鑫雨, 金海燕. 基于时空间联合去噪的改进差分进化算法[J]. 计算机科学, 2024, 51(9): 299-309.

WANG Bin, ZHANG Xinyu, JIN Haiyan. Improved Differential Evolution Algorithm Based on Time-Space Joint Denoising [J]. Computer Science, 2024, 51(9): 299-309.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于演化多目标聚类的SAR图像变化检测](#)

Change Detection in SAR Images Based on Evolutionary Multi-objective Clustering

计算机科学, 2024, 51(9): 140-146. <https://doi.org/10.11896/jsjcx.230800014>

[基于伪标签依赖增强与噪声干扰消减的小样本图像分类](#)

Few-shot Image Classification Based on Pseudo-label Dependence Enhancement and Noise Interference Reduction

计算机科学, 2024, 51(8): 152-159. <https://doi.org/10.11896/jsjcx.230500066>

[基于残差密集卷积自编码的高噪声图像去噪方法](#)

Residual Dense Convolutional Autoencoder for High Noise Image Denoising

计算机科学, 2024, 51(6A): 230400073-7. <https://doi.org/10.11896/jsjcx.230400073>

[基于MCC的后端优化方法及其在ORB-SLAM2中的应用](#)

MCC-based Back-end Optimization Method and Its Application in ORB-SLAM2

计算机科学, 2024, 51(6A): 230600081-7. <https://doi.org/10.11896/jsjcx.230600081>

[有序标签噪声的鲁棒估计与过滤方法](#)

Robust Estimation and Filtering Methods for Ordinal Label Noise

计算机科学, 2024, 51(6): 144-152. <https://doi.org/10.11896/jsjcx.230700115>

基于时空间联合去噪的改进差分进化算法

王彬¹ 张鑫雨¹ 金海燕^{1,2}

1 西安理工大学计算机科学与工程学院 西安 710048

2 西安理工大学陕西省网络计算与安全技术重点实验室 西安 710048

摘要 在工程问题的优化求解过程中,对个体的适应度评价可能会受到环境噪声的干扰,进而影响对种群进行合理的优胜劣汰操作,造成算法性能下降。为了对抗噪声环境的影响,提出了一种基于时空间联合去噪的改进差分进化算法(SEDADe)。根据适应度排名将种群划分成两个子种群,对评价较差个体组成的子种群用分布估计算法(EDA)进化,采用高斯分布建模解空间,利用解空间中多个个体噪声的随机性抵消噪声影响;对评价较好个体组成的子种群用差分进化算法(DE)进化,并且引入基于时间的停滞重采样机制去噪,提高收敛精度。对时空间混合进化得到的两个子种群进行基于概率选择的EDA信息利用操作,利用EDA搜索得到的全局信息引导DE的搜索方向,避免陷入局部最优。在实验中使用了被零均值高斯噪声干扰的基准函数,可以发现SEDADe相比其他算法更具有竞争性,此外通过消融实验验证了所提算法包含的3个机制的有效性和合理性。

关键词: 差分进化;分布估计;噪声;重采样;混合进化;信息利用

中图分类号 TP301

Improved Differential Evolution Algorithm Based on Time-Space Joint Denoising

WANG Bin¹, ZHANG Xinyu¹ and JIN Haiyan^{1,2}

1 School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

2 Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an University of Technology, Xi'an 710048, China

Abstract In the optimization process of solving engineering problems, the evaluation of individual fitness may be affected by environmental noise, so as to affect the reasonable survival of the fittest operation on the population, and result in a decline in algorithm performance. In order to combat the impact of noise environment, an improved differential evolution algorithm (SEDADe) based on joint temporal and spatial denoising is proposed. The population is divided into two subpopulations according to fitness ranking, and the subpopulations composed of poorly evaluated individuals are evolved using a distribution estimation algorithm (EDA). Gaussian distribution is used to model the solution space, using the randomness of multiple individual noises in the solution space to offset the noise impact. Differential evolution algorithm (DE) is used to evolve subpopulations with better evaluated individual composition, and a time-based stagnation resampling mechanism is introduced to denoise to improve convergence accuracy. The EDA information utilization operation based on probability selection is performed on the two subpopulations derived from time-space mixed evolution, and the global information obtained from EDA search is used to guide the search direction of DE to avoid falling into local optimization. In the experiment, a benchmark function interfered by zero mean Gaussian noise is used, and it is found that SEDADe is competitive with other algorithms. In addition, the effectiveness and rationality of the proposed mechanism are verified through ablation experiments.

Keywords Differential evolution, Distribution estimation, Noise, Resampling, Hybrid evolution, Information utilization

1 引言

进化算法(Evolutionary Algorithms, EAs)是一种受生物进化启发的全局优化方法,能够在不同性质的复杂优化问题中找到对应的最优解,具有较高的鲁棒性和普适性^[1]。进化算法旨在通过模仿适者生存的达尔文原理来解决复杂的优化问题,生物适应度的概念在进化计算中通过适应度函数来

体现。优化问题的潜在候选解,通过交叉变异等操作进化,在竞争选择阶段保留优秀候选解到下一代。依据候选解的适应度函数值进行优胜劣汰是进化的一个重要步骤,其从候选解种群中过滤出高质量的解决方案,同时丢弃较差的解决方案。进化算法在模式识别、图像处理、人工智能等众多领域都获得了较为成功的应用,如在通信网络优化、飞机外形设计、图像处理以及电力系统优化设计中^[2],进化算法在无噪声情况下,

到稿日期:2023-06-08 返修日期:2023-11-19

基金项目:国家自然科学基金(62272383,62372369)

This work was supported by the National Natural Science Foundation of China(62272383,62372369).

通信作者:金海燕(jinhaiyan@xaut.edu.cn)

表现出了较高的鲁棒性和普适性。

一般情况下,适应度函数通常是确定性和无噪声的^[3],但是,其很难被应用于现实世界中^[4]。在实际工程优化问题的建模过程中,噪声来源多样,对优化求解会产生较大的影响^[5],因此,图像去噪在计算机视觉领域得到了广泛研究^[6]。在噪声环境下的优化问题中,进化算法会受到噪声的影响。当存在噪声时,进化选择过程可能变得不稳定^[7],搜索过程的收敛性也可能会受到不利影响。一般称这类问题的优化为噪声问题优化(Noisy Optimization Problems, NOPS)^[8-9]。

噪声问题优化难以求解的原因如下:

- (1)较好的候选解由于低估而被抛弃;
- (2)较差的候选解由于高估而被保留。

噪声问题优化的主要关注点是:在适应度环境中存在噪声的情况下,根据竞争决策方案的适应度估计措施,降低向下一代推广劣质解决方案的风险^[10]。

噪声问题的优化,很大程度受噪声模型和噪声强度的影响,噪声环境下的进化算法大多采用的是高斯噪声。目前用于解决噪声问题的主要方法有3种^[11]:显式平均法、隐式平均法、改变选择算子。

(1)显示平均法是对每个候选解进行多次适应度值估计,用平均值作为目标值的期望值。重采样是处理噪声最简单的方法,通过平均逼近真实适应度值。事实上,增加样本量相当于减少估计适应度的方差。因此,理想情况下,无限的样本量会将适应度估计中的不确定性减少到零,从而将问题转化为无噪声问题^[12]。该方法的不足之处在于重采样会增加适应度值的评价次数。

(2)隐式平均法是扩大种群规模,当种群规模足够大时,将会得到更多的类似候选解,噪声的影响随着算法反复搜索同一片区域而被削弱,但这会增加更多的计算成本。

(3)改进选择算子是通过在父代个体和子代个体之间建立一种选择关系,来选择较为合适的个体并保留到下一代。阈值法是经典的改进选择算子,也被称为“阈值法去噪”^[13]。阈值法是在选择步骤中根据阈值更新候选解。只有当子代的适应度值比父代的适应度值的改进差值大于阈值 τ 时,才能用子代替换父代。该方法的不足之处是需要寻找一个最优的静态值或修正的 τ 适应规则。

近年来,研究者对差分进化算法进行了相关改进,以解决噪声问题。文献[14]提出了DE-RSF-TS(随机缩放因子阈值选择算法),在该算法中,用于加权差分向量的缩放因子是随机的^[15],缩放因子在 $[0.5, 1]$ 范围内随机变化,使得DE更加随机化,这种方案称为DE-RSF。在此基础上,又将阈值选择法结合到DE-RSF中,只有当子代的适应度值比父代的适应度值的改进差值大于阈值 τ 时,才能用子代替换父代,阈值 τ 与噪声强度或者方差成比例,即 $\tau = k * \sigma^2$ 。文献[16]提出了一种自适应控制参数的差分进化算法(JDE),该算法比正常差分进化算法的收敛性更好。文献[17]提出了ODE算法,在该算法中,基于对立学习来进行种群初始化、世代跳跃以发现种群的最佳个体,从而更好地逼近候选解。对立学习引入了第二次机会去感知最优解的偏移。文献[18]提出了NRDE算法,该算法在DE/rand/1和DE/best/1经典变异策略下

实现等概率切换,使用二项式交叉和混合交叉,最后通过基于阈值的选择机制允许不太合适的候选解偶尔存活到下一代,从而对抗噪声函数。文献[19]提出了一种双环境PSO算法(DEPSO),其通过基于top- k 精英粒子的加权搜索中心来引导群体。

进化算法应该同时利用优质解的局部信息和种群的全局信息。局部信息有助于开发,而全局信息可以指导探索有希望的领域。分布估计算法(Estimation of Distribution Algorithm, EDA)中的搜索主要基于全局信息,而差分进化算法(Differential Evolution, DE)基于距离和方向,更擅于挖掘局部信息。目前,有些学者已经开展了将EDA与其他算法结合的研究,文献[20]通过改进分布估计算法,通过粒子群优化算法(PSO)每次迭代的位置更新,找到每次迭代的最优种群。文献[21]提出了一种并行混合进化算法(EDAs/PSO),采用分布估计算法和粒子群算法两种模型,在保持分布估计算法较强的全局搜索能力的基础上,还保持了粒子群算法较强的局部搜索能力。文献[22]提出了DE和EDA相结合的算法,将二者获取的信息相结合用以创建出更好的候选解。

针对噪声环境,本文提出了一种基于时空联合去噪的改进差分进化算法(SEDADE),具体包括3个机制:基于时间的停滞重采样去噪机制、基于空间的分布估计模型去噪机制和基于概率选择的EDA信息利用机制。

2 相关理论

2.1 差分进化算法

差分进化算法(Differential Evolution, DE)是一种高效的全局优化算法^[23-25],其原理是通过向量之间的差异扰动整个种群以达到求解问题的最优解的目的,具有较高的鲁棒性和普适性。算法从随机生成初始种群开始,然后通过从每一代群体中实现变异、交叉和选择操作来进化群体。当满足某些预定义的条件时,算法终止。

2.1.1 种群初始化

在算法开始时,使用特定的方法生成初始种群。最常用的初始化方法如式(1)所示:

$$X_{i,j} = X_j^L + (X_j^U - X_j^L) * rand \quad (1)$$

其中, $X_{i,j}$ 是个体 i 的第 j 维的初始值, X_j^L 是第 j 维的下界, X_j^U 是第 j 维的上界, $rand$ 是服从 $[0, 1]$ 均匀分布的随机数。

2.1.2 变异

对于每一个向量 $\mathbf{X}_{i,G}$,都会通过变异策略生成变异向量 $V_{i,G}$,以下是经典的变异策略公式:

$$\begin{aligned} & \text{DE/rand/1:} \\ \mathbf{V}_{i,G+1} &= \mathbf{X}_{r1,G} + F * (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G}) \end{aligned} \quad (2)$$

$$\begin{aligned} & \text{DE/rand/2:} \\ \mathbf{V}_{i,G+1} &= \mathbf{X}_{r1,G} + F * (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G}) + F * (\mathbf{X}_{r4,G} - \mathbf{X}_{r5,G}) \end{aligned} \quad (3)$$

$$\begin{aligned} & \text{DE/best/1:} \\ \mathbf{V}_{i,G+1} &= \mathbf{X}_{\text{best},G} + F * (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G}) \end{aligned} \quad (4)$$

$$\begin{aligned} & \text{DE/best/2:} \\ \mathbf{V}_{i,G+1} &= \mathbf{X}_{\text{best},G} + F * (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G}) + F * (\mathbf{X}_{r4,G} - \mathbf{X}_{r5,G}) \end{aligned} \quad (5)$$

其中, F 是缩放因子, $\mathbf{X}_{\text{best},G}$ 是第 G 代种群中最好的个体, r_1, r_2, r_3, r_4 和 r_5 是在 $[1, N]$ 中随机选择的索引 ($r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$), N 是种群规模。

2.1.3 交叉

得到变异向量之后,将交叉算子应用于父代向量 $\mathbf{X}_{i,G}$ 和变异向量 $\mathbf{V}_{i,G}$, 以便获得试验向量 $\mathbf{U}_{i,G}$ 。交叉运算常使用式(6)表示:

$$\mathbf{U}_{i,G} = \begin{cases} \mathbf{V}_{i,G}, & \text{if } \text{rand} \leq CR \text{ or } j = j_{\text{rand}} \\ \mathbf{X}_{i,G}, & \text{otherwise} \end{cases} \quad (6)$$

其中, $j=1, 2, \dots, D$ (D 是优化问题的维度); CR 是交叉概率, 范围为 $[0, 1]$; j_{rand} 是从 $[1, 2, \dots, D]$ 中随机选取的。

2.1.4 选择

利用当前的噪声 DE 算法,一旦获得试验向量,就通过选择算子从父代向量和试验向量中选择更好的一个,该试验向量将存活到下一代。选择计算式如式(7)所示:

$$\mathbf{X}_{i,G+1} = \begin{cases} \mathbf{U}_{i,G}, & \text{if } f(\mathbf{U}_{i,G}) \leq f(\mathbf{X}_{i,G}) \\ \mathbf{X}_{i,G}, & \text{otherwise} \end{cases} \quad (7)$$

其中, $f(\mathbf{U}_{i,G})$ 是试验向量的目标函数值, $f(\mathbf{X}_{i,G})$ 是父代向量的目标函数值。

2.2 分布估计算法

分布估计算法(EDA)^[26-29]是一种基于概率统计方法的进化算法,呈现出与传统进化算法不同的特征,其采用的是概率模型的学习和采样。

分布估计算法通过一个概率模型描述候选解在空间的分布,采用统计学习手段从群体宏观的角度建立一个描述解分布的概率模型,然后对概率模型随机采样产生新的种群,如此反复进行,实现种群的进化,直到终止条件。由于其采用了

统计的方式建模,自然具有对抗随机噪声的潜在能力。

分布估计算法基本步骤包括:(1)初始化种群;(2)选择优势群体;(3)建立概率模型;(4)随机采样;(5)生成新群体;(6)判断终止条件是否满足,若满足,则输出优化结果,否则,执行第(2)步。

种群增量学习(Population-Based Incremental Learning, PBIL)算法是 EDA 领域中一种典型的分支算法,其采用的概率模型是高斯模型。

2.3 进化过程的环境噪声

在实际应用中,噪声源来源多样,且大多数噪声源满足相互独立假设。当噪声源数量足够多时,此时根据中心极限定理可知,这些噪声源累加的结果服从高斯分布。为了模拟噪声环境,在实验中对基准函数加入了零均值的高斯噪声。

在噪声环境下,根据文献[30-33]中的描述,通常情况下候选解的噪声目标函数表示为:

$$f_{\text{noisy}}(\vec{x}) = f(\vec{x}) + N(0, \sigma^2) \quad (8)$$

其中, $f(\vec{x})$ 是真实目标函数值, $f_{\text{noisy}}(\vec{x})$ 是包含噪声的目标函数值, $N(0, \sigma^2)$ 是均值为 0、方差为 σ^2 的正态分布。

噪声环境下,真实适应度会受到噪声的干扰,在模拟实验中,通常在适应度函数上添加加性噪声,得到噪声适应度函数。本文假设随机噪声函数在重采样时间上具有不变性,在个体空间分布上具有一致性。

3 SEDADE 算法

3.1 算法框架设计

本文提出了一种基于时空间联合去噪的改进差分进化算法(SEDAD E),算法框架结构如图 1 所示。

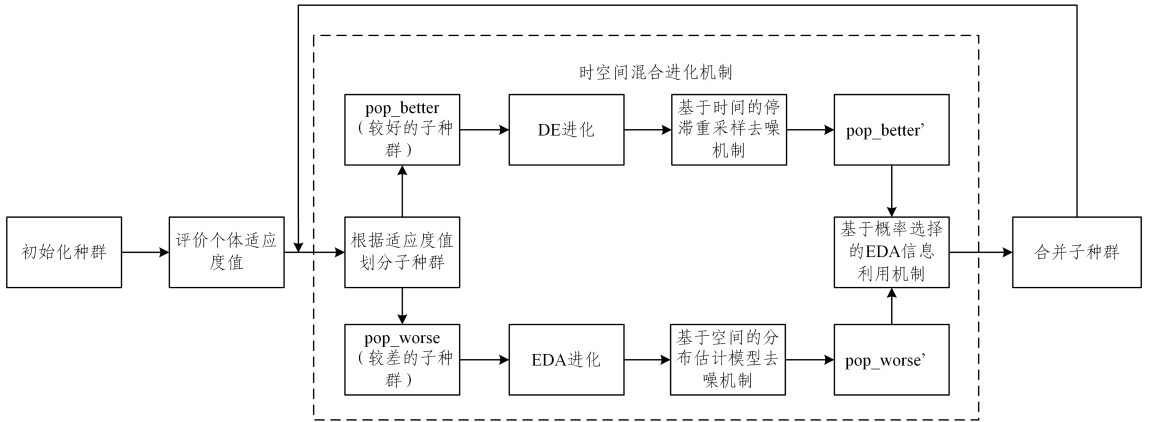


图 1 SEDADE 的框架结构图

Fig. 1 Structure diagram of SEDADE

(1)根据适应度值排名将种群均匀划分成两个子种群 pop_better 和 pop_worse , 两个子种群大小均为 $0.5 * N$ (N 是种群规模)。

(2)排名前 50% 的种群个体 (pop_better 子种群)用 DE 算法进化,并且引入基于时间的停滞重采样机制。当父代与子代停止更新的次数等于指定停滞次数时,对子代进行重采样,在时间的概念上避免种群受噪声的严重影响而陷入停滞进化状态,避免个体选择操作停滞导致的计算资源浪费,进一步减弱噪声的影响。

(3)排名后 50% 的种群个体 (pop_worse 子种群)用 EDA 算法进化,使用基于空间的分布估计模型去噪机制,通过在解空间上的探索生成概率模型,根据概率模型随机采样生成个体,在空间的概念上对个体进化并实现个体去噪。

(4)对时空间混合进化得到的两个子种群进行基于概率选择的 EDA 信息利用操作,从 DE 进化得到的子种群中选出较差的 $10% * N$ 个个体组成 DEP 子种群,从 EDA 进化得到的子种群中选出较好的 $10% * N$ 个个体组成 $EDAP$ 子种群。当满足概率条件时,从 $EDAP$ 子种群中随机选择两个

个体,对 *DEP* 子种群中的个体进行信息利用操作,利用 EDA 搜索空间的全局信息为 DE 的进一步搜索提供有用信息。

(5)最终将两个子种群混合在一起,形成完整种群,进行下一次进化,一直迭代,最终得到最优解。

图 1 中“基于时间的停滞重采样去噪机制”“基于空间的分布估计模型去噪机制”和“基于概率选择的 EDA 信息利用机制”是算法克服噪声影响的主要手段。

3.2 基于时间的停滞重采样去噪机制

本节介绍基于时间的停滞重采样机制。对 *pop_better* 子种群使用 DE 算法进化得到 *pop_better'*,在进化过程中的选择操作中,当父代和子代不进行更新的次数达到一定值时,对子代个体进行重采样操作。在一个时间段内,对个体进行重采样,缩短劣质解在进化种群中的生命周期,可以保证在节约评价次数的同时,减弱噪声对个体适应度值的影响。具体操作的伪代码描述见算法 1。

算法 1 基于时间的停滞重采样机制

参数:最大停滞次数 MaxStopnum;个体的初始停滞次数 sn;父代个体 X(P);子代个体 X(F);父代个体适应度值 f(P);子代个体适应度值 f(F);

1. if sn > MaxStopnum
2. 使用式(11)计算适应度值;
3. sn = 0;
4. else
5. 使用式(10)计算适应度值;
6. end if
7. if f(P) > f(F)
8. X(P) = X(F);
9. f(P) = f(F);
10. else
11. sn = sn + 1;
12. end if

基于时间的停滞重采样机制的优点包括:

(1)提高算法准确性。当算法运行时间过长或者陷入局部最优而导致算法停滞时,算法可能会因为计算能力达到极限而出现误差。如果在算法停滞时进行重采样操作,可以提高算法的准确性,减少噪声对适应度函数的影响。

(2)降低算法资源消耗。在算法停滞时进行重采样操作,可以避免算法停滞所导致的计算资源浪费。

(3)增加种群的多样性。在算法停滞的阶段,重采样可以减少噪声的影响,并增加种群的多样性,从而提高算法的鲁棒性,平衡噪声对候选解的适应度值的偏差。

本文中适应性评价次数 $anum = 10\% * N$ 。

- (1)对当前种群中每个个体进行重采样 $anum$ 次;
- (2)计算适应度均值:

$$\bar{f} = \frac{\sum_{i=1}^{anum} f_i}{anum} \quad (9)$$

3.3 基于空间的分布估计模型去噪机制

本节介绍基于空间的分布估计模型去噪机制。对 *pop_worse* 子种群使用 EDA 算法进化得到 *pop_worse'*,在进化的过程中构建描述解空间的概率模型:通过对种群的评估,选择其中的优秀个体集合,然后采用统计学习的手段构造一个

描述当前解集的概率模型。通过在解空间中的探索生成概率模型,根据概率模型随机采样生成个体,充分利用在空间上的搜索信息对个体进化并实现去噪。本节使用 2.2 节中所述的 PBIL 算法构建概率模型,具体操作见算法 2。分布估计模型去噪流程如下:

- (1)从 *pop_worse* 子种群中选择优势个体;
- (2)根据优势个体构建当前解集的概率模型;
- (3)根据概率模型随机采样生成一个新的子种群。

分布估计算法(EDA)与其他进化算法不同,其将进化算法中的变异和交叉操作进行了替换,具体替换方法为:

- (1)采用统计学习的手段对较优解集的概率分布进行统计估计;
- (2)根据统计估计得到的概率分布模型对解空间进行采样,得到新的一组解集。

分布估计算法的进化策略通过上述方法避免了传统进化算法中交叉算子和变异算子的适应度评价带来的随机性和盲目性,提高了进化搜索的效率。

算法 2 EDA 算法

1. 随机产生 M 个个体作为初始种群;(本算法使用划分的子种群 *pop_worse* 作为初始种群)
2. 计算 M 个个体的适应度值,如果符合终止条件,算法结束,否则继续进行;
3. 选择最优的 K 个个体用来更新概率向量 $P(x)$, $K \leq M$,更新方式为:

$$p_{i+1} = (1-LR)p_i(x) + LR \frac{1}{K} \sum_{k=1}^K x_i^k$$

4. 由新的概率模型采样 M 次,得到新一代群体 *pop_worse'*,返回第 2 步。

本文算法采用了时空间混合进化机制,通过适应度值排名对种群进行划分。对于较好的个体,利用基于时间去噪的 DE 进化,使得较好的个体向着最优解进化,试图逐渐找到最优解;对于种群中较差的个体,利用概率模型在搜索空间上的采样信息,使得较差的个体向着有希望的解空间进化。

3.4 基于概率选择的 EDA 信息利用机制

本节介绍基于概率选择的 EDA 信息利用机制。从基于空间去噪的 EDA 进化得到的子种群 *pop_worse'* 中选出较好的 $cronum$ 个个体组成 EDAP 子种群,从基于时间去噪的 DE 进化得到的子种群 *pop_better'* 中选出较差的 $cronum$ 个个体组成 DEP 子种群。使用基于概率选择的 EDA 信息利用策略,当满足概率条件时,从 EDAP 中随机选择两个个体,对 DEP 子种群个体进行信息利用操作。进行信息利用操作的目的是利用 EDA 进化得到的种群中较好的个体信息以及概率模型在解空间上的采样信息,帮助 DE 进化得到的种群中受噪声影响严重的个体进行二次进化,从而克服噪声的影响。当随机概率 $rand$ 小于信息交叉概率 p 时,执行式(10)的操作:

$$DEP_i = INF * DEP_i + (1-INF) * (EDAP_{r_1} - EDAP_{r_2}) \quad (10)$$

其中,INF 是信息交换因子, r_1 和 r_2 是在 $[1, cronum]$ 中随机选择的索引($r_1 \neq r_2 \neq i$)。

3.5 算法描述

算法 3 较为完整地描述了本文提出的 SEDADE 算法。

算法 3 SEDADE 算法

输入:种群规模 N ; $F=0.5$, $CR=0.9$; 问题维度 D ; 信息交叉概率 $p=0.5$, 交叉数量 $cronum=0.1 * N$, 重采样次数 $cn=0.1 * N$, 信息交换因子 $INF=0.5$, 最大停滞次数 $MaxStopnum=0.1 * N$, 最大函数评价次数 $MAXFES$

输出:噪声函数下最优解

1. 根据式(1)在给定的搜索空间内随机生成初始种群 POP
2. 利用式(10)计算个体的适应度函数值 $f(\mathbf{X}_i)$, 其中 $i=1, 2, \dots, N$
3. 初始化停滞次数 $sn_i=0, i=1, 2, \dots, N$
4. While not terminate do
 5. 根据适应度值排名将种群均匀划分为 pop_better 和 pop_worse
 6. 使用基于空间去噪的 EDA 算法对 pop_worse 子种群进化得到 pop_worse' (具体操作见算法 2)
 7. 使用基于时间去噪的 DE 算法对 pop_better 子种群进化得到 pop_better' (具体操作见算法 3 第 8—25 行)
 8. for $i=1$ to $0.5 * N$
 9. 利用式(2)计算 $V_{i,G}$
 10. for $j=1$ to D
 11. 利用式(6)计算 $U_{i,j,G}$
 12. end for
 13. if $sn_i > MaxStopnum$
 14. 使用式(11)对 $U_{i,j,G}$ 进行 cn 次重采样
 15. $sn_i=0$
 16. else
 17. 利用式(10)计算 $U_{i,G}$ 个体的适应度函数值 $f(U_i)$
 18. end if
 19. if $f(\mathbf{X}_i) > f(U_i)$
 20. $\mathbf{X}_{i,G+1} = U_i$
 21. $f(\mathbf{X}_{i,G+1}) = f(U_i)$
 22. else
 23. $sn_i = sn_i + 1$
 24. end if
 25. end for
 26. 从进化后的 pop_better' 和 pop_worse' 中分别选出较差的和较好的 $cronum$ 个个体组成 DEP 和 EDAP
 27. if $rand < p$
 28. for $i=1$ to $cronum$
 29. 利用式(12)计算 DEP _{i}
 30. end for
 31. end if
 32. 将新得到的 DEP 放回 pop_better'
 33. 将 pop_better' 和 pop_worse' 合并成一个新种群 POP
 34. end While

与 DE 方法对比,该算法在基于空间的去噪步骤中需要根据个体分布计算概率模型和个体采样,增加了计算环节;在基于时间的去噪步骤中需要计算某个个体多次评价的均值,额外消耗了评价次数。SEDADE 算法的时间、空间复杂度有所增加,但从第 4 章中的实验结果来看,算法复杂度的提升有助于算法性能的提高。

4 实验与结果分析**4.1 噪声测试基准函数**

为了评估所提出的 SEDADE 算法的性能,本文选择使用

由 16 个标准函数组成的噪声优化基准函数^[18],根据 30 次独立运行的平均最佳适应度值比较算法性能。

为了模拟噪声环境,对基准函数加入了零均值的高斯噪声,噪声基准函数定义见 2.3 节式(8)。

4.2 对比的算法

为了评估 SEDADE 算法的有效性,将本文提出的 SEDADE 算法与其他算法进行了比较,算法的种群规模大小统一设置为 $N=100$,对比算法详情如下:

(1)一种标准的 DE/rand/1/bin 算法^[23]。

(2)DE-RSF-TS^[14]:一种采用随机比例因子和基于阈值的选择的改进 DE 算法。该算法中使用的比例因子是随机的,可以在 $0.5 \sim 1$ 之间变化,交叉率 $CR=0.3$ 。该方法采用基于阈值的选择机制,该机制依赖于一个常量值。

(3)JDE^[16]:一种参数自适应设置的 DE 变体算法。

(4)一种改进的参数和策略控制的抗噪声差分进化算法^[18],该算法在两种可选的变异策略、可选的混合交叉和基于阈值的选择机制之间实现等概率切换。

(5)EDA^[30]:一种基于 PBIL 的分布估计算法。

(6)DE-EDA^[22]:DE/EDA 混合进化的全局优化进化算法。

(7)DEPSO^[19]:一种双环境 PSO 算法。

4.3 消融实验

本节通过消融实验验证 SEDADE 算法中每个机制在噪声环境下起到的作用,分别在 DE 的基础上加入相关机制来验证机制的改进效果,具体设定如下:

(1)3.2 节中基于时间的停滞重采样机制,记为 TDR;

(2)3.3 节中基于空间的分布估计模型去噪机制,记为 SEDA;

(3)3.2 和 3.3 节机制结合的时空混合进化机制,记为 TDR+SEDA;

(4)3.4 节中基于概率选择的 EDA 信息利用机制,记为 IUM。

表 1 和表 2 分别列出了在 16 个测试函数下且噪声方差为 0.4 和 0.8 的噪声环境下的实验对比结果,对所有算法的每个问题进行 30 次独立运行,并给出最终结果的平均值和标准偏差。对于所有对比算法,16 个测试函数的最大函数评价数量限制为 $D * 10^4$ 。使用 Friedman test 对所有算法进行对比,Avg Rank 是 Friedman test 的平均排名,Avg Rank 值越小表示该算法性能越好,表中字体加粗的数据表示对应函数下最优的结果。

本节设置了 6 种算法,其中 DE 是标准的差分进化算法;SEDA 是基于空间的分布估计模型去噪算法;DE+TDR 是在 DE 的基础上加入基于时间的停滞重采样机制;DE+SEDA 是在 DE 的基础上加入基于空间的分布估计模型去噪机制,二者混合进化;DE+TDR+SEDA 是在 DE 的基础上加入基于时间的停滞重采样机制和基于空间的分别估计模型去噪机制;DE+TDR+SEDA+IUM 即为本文提出的 SEDADE 算法,在 DE+TDR+SEDA 的基础上加入了基于概率选择的 EDA 信息利用机制。

表 1 噪声方差 $\sigma^2=0.4$ 时加性噪声测试函数下的 COST \pm STANDARDTable 1 COST \pm STANDARD under additive noise test function with noise variance $\sigma^2=0.4$

Function	DE	SEDA	DE+TDR	DE+SEDA	DE+TDR+SEDA	DE+TDR+SEDA+IUM
F1	1.58 (2.96×10^{-1})	4.45×10^{-1} (1.65×10^{-1})	6.85×10^{-1} (1.92×10^{-1})	1.20×10^{-1} (9.29×10^{-2})	1.19×10^{-1} (1.29×10^{-1})	1.06×10^{-1} (4.92×10^{-2})
F2	2.79 (3.81×10^{-1})	9.64×10^{-1} (3.24×10^{-1})	2.57 (5.11×10^{-1})	2.90×10^{-1} (6.33×10^{-2})	2.45×10^{-1} (4.96×10^{-2})	2.34×10^{-1} (7.87×10^{-2})
F3	6.89×10^{-1} (1.50×10^{-1})	3.95×10^{-1} (1.10×10^{-1})	6.80×10^{-1} (1.57×10^{-1})	1.14×10^{-1} (5.99×10^{-2})	1.01×10^{-1} (2.35×10^{-2})	1.39×10^{-1} (6.63×10^{-2})
F4	7.44×10^{-1} (2.14×10^{-1})	5.80×10^{-1} (1.32×10^{-1})	6.78×10^{-1} (2.49×10^{-1})	1.36×10^{-1} (8.15×10^{-2})	9.99×10^{-2} (1.93×10^{-2})	1.12×10^{-1} (1.27×10^{-1})
F5	1.39 (3.23×10^{-1})	5.95×10^{-1} (1.61×10^{-1})	6.82×10^{-1} (2.16×10^{-1})	1.29×10^{-1} (1.15×10^{-1})	1.34×10^{-1} (5.92×10^{-2})	1.10×10^{-1} (1.04×10^{-1})
F6	1.53×10^2 (6.96)	3.21×10 (1.01×10)	1.34×10^2 (2.27×10)	2.70 (9.23×10^{-1})	2.49 (9.13×10^{-1})	2.77×10^{-1} (1.19×10^{-1})
F7	0.00 (0.00)	0.00 ($0.00 \times 10^{+00}$)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
F8	1.13 (4.56×10^{-1})	6.55×10^{-1} (2.41×10^{-1})	8.90×10^{-1} (3.32×10^{-1})	7.26×10^{-1} (3.72×10^{-1})	5.05×10^{-1} (2.85×10^{-1})	6.87×10^{-1} (4.81×10^{-1})
F9	2.28×10 (1.28)	1.66×10 (5.42×10^{-1})	$2.22 \times 10^{+01}$ (1.26)	1.76×10 (1.38)	1.80×10 (1.69)	1.59×10 (3.80×10^{-1})
F10	1.35×10^2 (2.67×10)	2.46×10 (8.18)	1.34×10^2 (2.74×10)	2.35 (1.38)	3.74 (1.79)	3.70×10^{-1} (1.27×10^{-1})
F11	2.66×10 (9.82×10^{-1})	2.79×10 (2.49×10^{-1})	2.65×10 (7.59×10^{-1})	2.89×10 (2.51×10^{-1})	2.85×10 (3.28×10^{-1})	2.78×10 (4.38×10^{-1})
F12	2.38 (3.74×10^{-1})	1.54 (1.40×10^{-1})	1.71 (2.45×10^{-1})	1.14 (7.01×10^{-2})	1.12 (5.62×10^{-2})	1.10 (2.77×10^{-1})
F13	8.98×10^{-2} (1.11×10^{-1})	3.23×10^{-2} (3.40×10^{-2})	7.08×10^{-2} (6.20×10^{-2})	3.65×10^{-2} (3.18×10^{-2})	4.15×10^{-2} (3.42×10^{-2})	1.41×10^{-1} (1.11×10^{-1})
F14	4.70×10^{-1} (1.04×10^{-1})	5.92×10^{-1} (8.64×10^{-2})	4.67×10^{-1} (8.83×10^{-2})	4.48×10^{-1} (1.28×10^{-1})	4.11×10^{-1} (1.45×10^{-1})	4.08×10^{-1} (3.38×10^{-2})
F15	1.33×10^{-1} (1.00×10^{-1})	3.38×10^{-2} (1.42×10^{-2})	1.08×10^{-1} (1.36×10^{-1})	1.21×10^{-1} ($1. \times 10 \times 10^{-1}$)	3.94×10^{-2} (2.70×10^{-2})	7.66×10^{-2} (6.11×10^{-2})
F16	1.52 (3.57×10^{-1})	9.43×10^{-1} (1.36×10^{-1})	1.41 (3.84×10^{-1})	3.44×10^{-1} (1.16×10^{-1})	3.97×10^{-1} (1.43×10^{-1})	3.43×10^{-1} (7.22×10^{-2})
Avg Rank	5.44	3.46	4.50	3.09	2.46	2.03

表 2 噪声方差 $\sigma^2=0.8$ 时加性噪声测试函数下的 COST \pm STANDARDTable 2 COST \pm STANDARD under additive noise test function with noise variance $\sigma^2=0.8$

Function	DE	SEDA	DE+TDR	DE+SEDA	DE+TDR+SEDA	DE+TDR+SEDA+IUM
F1	8.93×10^{-1} (2.48×10^{-1})	6.95×10^{-1} (1.14×10^{-1})	8.43×10^{-1} (2.96×10^{-1})	3.68×10^{-1} (3.71×10^{-2})	1.01×10^{-1} (4.50×10^{-2})	2.32×10^{-1} (2.33×10^{-1})
F2	4.03 (3.75×10^{-1})	2.40 (6.34×10^{-1})	3.94 (6.14×10^{-1})	6.55×10^{-1} (8.22×10^{-2})	3.60×10^{-1} (1.22×10^{-1})	4.11×10^{-1} (2.29×10^{-1})
F3	9.69×10^{-1} (1.69×10^{-1})	8.04×10^{-1} (2.28×10^{-1})	8.98×10^{-1} (2.43×10^{-1})	4.68×10^{-1} (3.05×10^{-1})	1.83×10^{-1} (6.88×10^{-2})	1.20×10^{-1} (4.79×10^{-2})
F4	9.61×10^{-1} (2.94×10^{-1})	6.72×10^{-1} (1.17×10^{-1})	8.05×10^{-1} (1.61×10^{-1})	2.97×10^{-1} (1.29×10^{-1})	1.85×10^{-1} (5.00×10^{-2})	1.70×10^{-1} (1.12×10^{-1})
F5	1.02 (1.16×10^{-1})	6.32×10^{-1} (8.88×10^{-2})	8.83×10^{-1} (3.68×10^{-1})	2.84×10^{-1} (1.65×10^{-01})	1.74×10^{-1} (1.11×10^{-1})	1.69×10^{-1} (9.66×10^{-2})
F6	1.29×10^2 (1.40×10)	2.80×10 (8.38)	1.21×10^2 (1.83×10)	2.02 (1.56)	1.60 (2.68)	4.74×10^{-1} (2.65×10^{-1})
F7	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
F8	7.92×10^{-1} (1.91×10^{-1})	8.05×10^{-1} (3.55×10^{-1})	7.52×10^{-1} (3.55×10^{-1})	5.94×10^{-1} (3.73×10^{-1})	6.44×10^{-1} (3.40×10^{-1})	3.66×10^{-1} (3.19×10^{-1})
F9	2.35×10 (1.34)	1.57×10 (1.19×10^{-1})	2.27×10 (3.07×10^{-1})	1.86×10 (2.27)	1.75×10 (1.41)	1.63×10 (1.08×10^{-1})
F10	1.41×10^2 (2.58×10)	2.81×10 (2.51)	1.33×10^2 (3.41×10)	2.44 (1.49)	2.01 (9.22×10^{-1})	4.11×10^{-1} (1.52×10^{-1})
F11	2.69×10 (6.74×10^{-1})	2.82×10 (3.26×10^{-1})	2.67×10 (7.60×10^{-1})	2.81×10 (1.29×10^{-1})	2.81×10 (1.37×10^{-1})	2.89×10 (3.94×10^{-1})
F12	2.03 (2.31×10^{-1})	1.62 (1.53×10^{-1})	1.90 (1.76×10^{-1})	1.20 (2.27×10^{-1})	1.19 (9.33×10^{-2})	1.28 (1.59×10^{-1})
F13	6.84×10^{-2} (5.27×10^{-2})	5.88×10^{-2} (3.36×10^{-2})	9.45×10^{-2} (5.92×10^{-2})	1.19×10^{-1} (1.25×10^{-1})	9.20×10^{-2} (9.14×10^{-2})	4.14×10^{-2} (3.98×10^{-2})
F14	4.93×10^{-1} (1.64×10^{-2})	4.91×10^{-1} (2.31×10^{-2})	5.20×10^{-1} (3.80×10^{-2})	5.45×10^{-1} (1.05×10^{-1})	5.02×10^{-1} (1.76×10^{-2})	4.53×10^{-1} (1.13×10^{-1})
F15	2.62×10^{-1} (2.78×10^{-1})	6.94×10^{-2} (4.31×10^{-2})	6.48×10^{-2} (3.58×10^{-2})	2.33×10^{-1} (1.94×10^{-1})	2.01×10^{-1} (1.30×10^{-1})	1.84×10^{-1} (1.62×10^{-1})
F16	2.22 (3.57×10^{-1})	1.41 (2.55×10^{-1})	1.82 (4.16×10^{-1})	6.72×10^{-1} (2.79×10^{-1})	5.00×10^{-1} (2.30×10^{-1})	5.94×10^{-1} (1.59×10^{-1})
Avg Rank	5.12	3.59	4.37	3.50	2.43	1.96

根据表 1 和表 2 的实验数据结果可以看出,DE 是 6 种算法中效果最差的,DE+TDR+SEDA+IUM 算法效果最佳,SEDA 的 Avg Rank 值小于 DE 的 Avg Rank 值,说明 SEDA 优于 DE,证明了基于空间的分布估计模型去噪效果优于标准的差分进化算法。DE+TDR 优于 DE,说明在 DE 的基础上加入基于时间的停滞重采样机制对噪声有一定的抑制作用。DE+SEDA 优于 DE,SEDA 和 DE+TDR,说明基于空间的分布估计模型去噪机制和 DE 混合进化能够进一步增强去噪效果。DE+TDR+SEDA 也就是在 DE 的基础上加入时空混合进化机制,DE+TDR+SEDA 的 Avg Rank 值小于上述 4 种算法,说明 DE+TDR+SEDA 优于 DE,SEDA,DE+TDR 和 DE+SEDA,证明了时空混合进化机制的有效性,也验证了在 DE+SEDA 的基础上加入基于时间的停滞重采样机制对算法的性能有一定的提升作用。DE+TDR+SEDA+IUM 算法即本文提出的 SEDADE 算法在 DE+TDR+SEDA 的基础上又加入了基于概率选择的 EDA 信息利用机制,DE+TDR+

SEDA+IUM 优于前面 5 种算法,在 16 个测试函数里取得了较好的效果,说明了基于概率选择的 EDA 信息利用机制在一定程度上能够提高算法的性能。因此,可以推断出本文提出的算法机制是合理的、有效的。

4.4 噪声环境下算法的对比效果

在本节中,对 3 个标准噪声强度(0.04,0.5 和 0.75)进行模拟实验,因此对于 16 个基准问题,总共有 64 个测试用例。将 SEDADE 算法的结果与 4.2 节中描述的所有算法进行比较。对所有算法的每个问题和每个噪声强度进行 30 次独立运行,并给出最终结果的平均值和标准偏差。对于所有对比算法,这 16 个测试函数的最大函数评价数量限制为 $D * 10^4$ 。使用 Friedman test 对所有算法进行比较,Avg Rank 是 Friedman test 的平均排名,Avg Rank 值越小表示该算法性能越好。

表 3、表 4 和表 5 分别是测试函数在噪声方差为 0.04,0.5 和 0.75 下的算法对比结果。

表 3 噪声方差 $\sigma^2=0.04$ 时加性噪声测试函数下的 COST \pm STANDARD

Table 3 COST \pm STANDARD under additive noise test function with noise variance $\sigma^2=0.04$

Function	DE	DE-RSF-TS	JDE	NRDE	EDA	DE-EDA	DEPSO	SEDADE
F1	4.90 $\times 10^{-1}$ (8.89 $\times 10^{-2}$)	1.97 $\times 10^{-1}$ (6.48 $\times 10^{-2}$)	2.42 $\times 10^{-1}$ (5.36 $\times 10^{-2}$)	2.75 $\times 10^{-1}$ (8.18 $\times 10^{-2}$)	1.52 $\times 10^{-1}$ (3.91 $\times 10^{-2}$)	1.97 $\times 10^{-1}$ (5.47 $\times 10^{-2}$)	7.63 $\times 10^{-2}$ (3.25 $\times 10^{-2}$)	7.23$\times 10^{-2}$ (2.96$\times 10^{-2}$)
F2	2.33 (2.61 $\times 10^{-1}$)	3.37 $\times 10^{-1}$ (5.96 $\times 10^{-2}$)	3.18 $\times 10^{-1}$ (5.24 $\times 10^{-2}$)	1.02 (5.55 $\times 10^{-1}$)	2.50 $\times 10^{-1}$ (4.00 $\times 10^{-2}$)	4.58 $\times 10^{-1}$ (2.59 $\times 10^{-1}$)	1.98 $\times 10^{-1}$ (5.17 $\times 10^{-2}$)	1.35$\times 10^{-1}$ (3.33$\times 10^{-2}$)
F3	1.57 $\times 10^{+01}$ (1.51)	2.14 $\times 10^{-1}$ (7.63 $\times 10^{-2}$)	2.00 $\times 10^{-1}$ (6.29 $\times 10^{-2}$)	2.17 $\times 10^{-1}$ (6.53 $\times 10^{-2}$)	1.45 $\times 10^{-1}$ (4.06 $\times 10^{-2}$)	2.26 $\times 10^{-1}$ (7.50 $\times 10^{-2}$)	1.05$\times 10^{-1}$ (4.13$\times 10^{-2}$)	1.08 $\times 10^{-1}$ (3.98 $\times 10^{-2}$)
F4	4.67 $\times 10^{-1}$ (1.09 $\times 10^{-1}$)	2.18 $\times 10^{-1}$ (6.98 $\times 10^{-2}$)	2.10 $\times 10^{-1}$ (6.57 $\times 10^{-2}$)	2.64 $\times 10^{-1}$ (7.80 $\times 10^{-2}$)	1.46 $\times 10^{-1}$ (3.38 $\times 10^{-2}$)	2.14 $\times 10^{-1}$ (5.76 $\times 10^{-2}$)	8.16 $\times 10^{-2}$ (2.84 $\times 10^{-2}$)	8.11$\times 10^{-2}$ (4.54$\times 10^{-2}$)
F5	4.51 $\times 10^{-1}$ (1.00 $\times 10^{-1}$)	2.26 $\times 10^{-1}$ (6.24 $\times 10^{-2}$)	2.11 $\times 10^{-1}$ (5.36 $\times 10^{-2}$)	2.19 $\times 10^{-1}$ (5.63 $\times 10^{-2}$)	1.70 $\times 10^{-1}$ (4.52 $\times 10^{-2}$)	2.19 $\times 10^{-1}$ (5.45 $\times 10^{-2}$)	8.09 $\times 10^{-2}$ (2.67 $\times 10^{-2}$)	8.09$\times 10^{-2}$ (3.19$\times 10^{-2}$)
F6	1.53 $\times 10^{+02}$ (8.98)	1.33 $\times 10^2$ (2.23 $\times 10$)	5.22 (3.33)	3.00 (1.69)	2.75 $\times 10$ (5.57)	2.92 $\times 10$ (9.79 $\times 10^{+00}$)	2.97 $\times 10$ (2.55 $\times 10$)	5.55$\times 10^{-1}$ (7.39$\times 10^{-1}$)
F7	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	3.00 $\times 10^{-1}$ (6.75 $\times 10^{-1}$)	0.00 (0.00)	2.00 $\times 10^{-1}$ (4.14 $\times 10^{-1}$)	0.00 (0.00)	0.00 (0.00)
F8	6.57 $\times 10^{-1}$ (2.60 $\times 10^{-1}$)	6.55 $\times 10^{-1}$ (2.48 $\times 10^{-1}$)	6.18 $\times 10^{-1}$ (3.18 $\times 10^{-1}$)	6.57 $\times 10^{-1}$ (3.03 $\times 10^{-1}$)	5.47 $\times 10^{-1}$ (2.83 $\times 10^{-1}$)	6.11 $\times 10^{-1}$ (3.05 $\times 10^{-1}$)	4.98 $\times 10^{-01}$ (3.18 $\times 10^{-1}$)	4.77$\times 10^{-1}$ (2.60$\times 10^{-1}$)
F9	4.37 $\times 10^{+03}$ (3.41 $\times 10^3$)	2.24 $\times 10$ (1.68)	1.80 $\times 10$ (6.99 $\times 10^{-1}$)	1.88 $\times 10$ (2.19)	1.56 $\times 10$ (3.21 $\times 10^{-1}$)	1.75 $\times 10$ (1.87)	2.52 $\times 10$ (1.38)	1.50$\times 10$ (2.92$\times 10^{-1}$)
F10	1.51 $\times 10^2$ (7.95)	1.32 $\times 10^2$ (2.39 $\times 10$)	5.00 (3.33)	3.55 (1.41)	2.92 $\times 10$ (7.06)	2.77 $\times 10$ (8.89)	3.47 $\times 10$ (2.88 $\times 10$)	3.92$\times 10^{-1}$ (5.37$\times 10^{-1}$)
F11	2.66 $\times 10$ (3.47 $\times 10^{-1}$)	1.55$\times 10$ (2.22)	2.22 $\times 10$ (1.21 $\times 10^{+00}$)	2.73 $\times 10$ (1.19)	2.73 $\times 10$ (6.13 $\times 10^{-1}$)	2.75 $\times 10$ (1.58)	2.70 $\times 10$ (1.95 $\times 10^{-1}$)	2.61 $\times 10$ (3.75 $\times 10^{-1}$)
F12	1.46 (9.24 $\times 10^{-2}$)	1.23 (7.20 $\times 10^{-2}$)	1.21 (3.78 $\times 10^{-2}$)	1.23 (7.14 $\times 10^{-2}$)	1.15 (4.54 $\times 10^{-2}$)	1.19 (6.51 $\times 10^{-2}$)	1.08 (2.31 $\times 10^{-2}$)	1.07 (3.49$\times 10^{-2}$)
F13	2.61 $\times 10^{-2}$ (2.79 $\times 10^{-2}$)	3.70 $\times 10^{-2}$ (4.52 $\times 10^{-2}$)	3.96 $\times 10^{-2}$ (4.62 $\times 10^{-2}$)	3.11 $\times 10^{-2}$ (3.46 $\times 10^{-2}$)	8.27$\times 10^{-3}$ (7.82$\times 10^{-3}$)	3.36 $\times 10^{-2}$ (4.04 $\times 10^{-2}$)	1.06 $\times 10^{-2}$ (9.82 $\times 10^{-3}$)	4.10 $\times 10^{-2}$ (3.49 $\times 10^{-2}$)
F14	1.85 $\times 10^{-1}$ (1.68 $\times 10^{-1}$)	1.14 $\times 10^{-1}$ (1.10 $\times 10^{-1}$)	1.15 $\times 10^{-1}$ (9.51 $\times 10^{-2}$)	1.05 $\times 10^{-1}$ (7.25 $\times 10^{-2}$)	7.17 $\times 10^{-2}$ (1.08 $\times 10^{-1}$)	2.88 $\times 10^{-1}$ (1.99 $\times 10^{-1}$)	6.92 $\times 10^{-2}$ (1.24 $\times 10^{-1}$)	6.78$\times 10^{-2}$ (5.92$\times 10^{-2}$)
F15	3.29 $\times 10^{-2}$ (4.45 $\times 10^{-2}$)	3.79 $\times 10^{-2}$ (3.82 $\times 10^{-2}$)	3.58 $\times 10^{-2}$ (4.15 $\times 10^{-2}$)	3.06 $\times 10^{-2}$ (3.48 $\times 10^{-2}$)	2.30 $\times 10^{-2}$ (1.63 $\times 10^{-2}$)	5.56 $\times 10^{-2}$ (4.91 $\times 10^{-2}$)	5.51$\times 10^{-3}$ (5.49$\times 10^{-3}$)	2.88 $\times 10^{-2}$ (2.62 $\times 10^{-2}$)
F16	1.03 (1.33 $\times 10^{-1}$)	4.74 $\times 10^{-1}$ (7.62 $\times 10^{-2}$)	4.28 $\times 10^{-1}$ (7.83 $\times 10^{-2}$)	4.19 $\times 10^{-1}$ (8.69 $\times 10^{-2}$)	3.20 $\times 10^{-1}$ (5.87 $\times 10^{-2}$)	4.75 $\times 10^{-1}$ (9.22 $\times 10^{-2}$)	3.43 $\times 10^{-1}$ (6.07 $\times 10^{-2}$)	1.70$\times 10^{-1}$ (4.05$\times 10^{-2}$)
Avg Rank	6.87	5.53	4.46	5.37	3.13	5.68	3.00	1.93

表 4 噪声方差 $\sigma^2=0.5$ 时加性噪声测试函数下的 COST \pm STANDARD

Table 4 COST \pm STANDARD under additive noise test function with noise variance $\sigma^2=0.5$

Function	DE	DE-RSF-TS	JDE	NRDE	EDA	DE-EDA	DEPSO	SEDADE
F1	1.77 (3.88 $\times 10^{-1}$)	7.53 $\times 10^{-1}$ (1.97 $\times 10^{-1}$)	7.57 $\times 10^{-1}$ (2.14 $\times 10^{-1}$)	8.12 $\times 10^{-1}$ (2.29 $\times 10^{-1}$)	5.20 $\times 10^{-1}$ (1.60 $\times 10^{-1}$)	7.34 $\times 10^{-1}$ (2.19 $\times 10^{-1}$)	2.80 $\times 10^{-1}$ (1.56 $\times 10^{-1}$)	2.80$\times 10^{-1}$ (1.27$\times 10^{-1}$)
F2	1.94 $\times 10$ (7.76 $\times 10^{-1}$)	3.38 (4.91 $\times 10^{-1}$)	3.21 (3.29 $\times 10^{-1}$)	3.32 (4.49 $\times 10^{-1}$)	1.30 (5.98 $\times 10^{-1}$)	5.06 (4.54)	1.16 (3.39 $\times 10^{-1}$)	4.59$\times 10^{-1}$ (1.51$\times 10^{-1}$)

(续表)

Function	DE	DE-RSF-TS	JDE	NRDE	EDA	DE-EDA	DEPSO	SEDADE
F3	1.67×10 (1.49)	7.15×10 ⁻¹ (2.56×10 ⁻¹)	6.74×10 ⁻¹ (1.99×10 ⁻¹)	7.16×10 ⁻¹ (2.31×10 ⁻¹)	5.86×10 ⁻¹ (1.45×10 ⁻¹)	8.35×10 ⁻¹ (2.78×10 ⁻¹)	3.74×10 ⁻¹ (1.36×10 ⁻¹)	3.17×10⁻¹ (1.26×10⁻¹)
F4	1.63 (4.51×10 ⁻¹)	7.61×10 ⁻¹ (2.37×10 ⁻¹)	7.11×10 ⁻¹ (1.89×10 ⁻¹)	8.43×10 ⁻¹ (2.83×10 ⁻¹)	5.76×10 ⁻¹ (1.60×10 ⁻¹)	6.73×10 ⁻¹ (2.36×10 ⁻¹)	2.99×10 ⁻¹ (1.02×10 ⁻¹)	2.94×10⁻¹ (1.25×10⁻¹)
F5	1.65 (3.77×10 ⁻¹)	7.81×10 ⁻¹ (1.87×10 ⁻¹)	6.68×10 ⁻¹ (1.90×10 ⁻¹)	8.65×10 ⁻¹ (2.34×10 ⁻¹)	5.49×10 ⁻¹ (1.35×10 ⁻¹)	7.21×10 ⁻¹ (2.58×10 ⁻¹)	2.98×10 ⁻¹ (9.41×10 ⁻²)	2.78×10⁻¹ (9.89×10⁻²)
F6	1.55×10 ² (9.19)	1.28×10 ² (2.35×10)	1.17×10 (2.81)	1.06×10 (1.73)	2.93×10 (7.16)	2.68×10 (7.22)	2.47×10 (2.12×10)	1.14 (1.07)
F7	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	7.00×10 ⁻¹ (1.06)	0.00 (0.00)	6.67×10 ⁻² (2.58×10 ⁻¹)	0.00 (0.00)	0.00 (0.00)
F8	1.36 (3.82×10 ⁻¹)	8.69×10 ⁻¹ (3.30×10 ⁻¹)	9.79×10 ⁻¹ (3.13×10 ⁻¹)	8.04×10 ⁻¹ (2.94×10 ⁻¹)	8.57×10 ⁻¹ (2.87×10 ⁻¹)	8.45×10 ⁻¹ (3.46×10 ⁻¹)	5.95×10⁻¹ (3.20×10⁻¹)	6.35×10 ⁻¹ (2.87×10 ⁻¹)
F9	4.66×10 ³ (3.15×10 ³)	2.27×10 (2.02)	2.03×10 (9.00×10 ⁻¹)	2.10×10 (1.44)	1.62×10 (6.09×10 ⁻¹)	2.20×10 (2.13)	2.55×10 (1.75)	1.59×10 (3.02×10⁻¹)
F10	1.51×10 ² (8.67)	1.40×10 ² (2.50×10)	1.20×10 (2.73)	1.04×10 (1.31)	2.57×10 (8.14)	3.00×10 (1.09×10)	2.20×10 (1.64×10)	1.42 (1.33)
F11	2.88×10 (5.84×10 ⁻¹)	2.71×10 (8.05×10 ⁻¹)	2.70×10 (7.21×10⁻¹)	2.88×10 (1.15)	2.79×10 (3.49×10 ⁻¹)	2.88×10 (1.01)	3.56×10 (4.25×10)	2.84×10 (6.21×10 ⁻¹)
F12	2.60 (4.40×10 ⁻¹)	1.80 (2.41×10 ⁻¹)	1.73 (2.17×10 ⁻¹)	1.74 (1.94×10 ⁻¹)	1.55 (1.54×10 ⁻¹)	1.69 (1.38×10 ⁻¹)	1.32 (1.36×10 ⁻¹)	1.32 (1.46×10⁻¹)
F13	1.55×10 ⁻¹ (1.33×10 ⁻¹)	9.51×10 ⁻² (8.37×10 ⁻²)	1.11×10 ⁻¹ (1.24×10 ⁻¹)	1.02×10 ⁻¹ (1.09×10 ⁻¹)	3.39×10 ⁻² (2.80×10 ⁻²)	1.31×10 ⁻¹ (1.17×10 ⁻¹)	3.20×10⁻² (2.39×10⁻²)	5.15×10 ⁻² (9.69×10 ⁻²)
F14	4.64×10 ⁻¹ (1.03×10 ⁻¹)	4.87×10 ⁻¹ (7.03×10 ⁻²)	4.83×10 ⁻¹ (5.70×10 ⁻²)	4.56×10 ⁻¹ (1.02×10 ⁻¹)	3.95×10⁻¹ (1.47×10⁻¹)	4.89×10 ⁻¹ (4.30×10 ⁻²)	4.94×10 ⁻¹ (7.50×10 ⁻²)	4.92×10 ⁻¹ (4.37×10 ⁻¹)
F15	1.20×10 ⁻¹ (1.10×10 ⁻¹)	1.24×10 ⁻¹ (1.96×10 ⁻¹)	1.08×10 ⁻¹ (1.06×10 ⁻¹)	1.53×10 ⁻¹ (1.58×10 ⁻¹)	4.72×10 ⁻² (4.38×10 ⁻²)	8.63×10 ⁻² (8.45×10 ⁻²)	2.03×10⁻² (1.76×10⁻²)	9.62×10 ⁻² (1.63×10 ⁻¹)
F16	3.49 (5.43×10 ⁻¹)	1.75 (3.11×10 ⁻¹)	1.44 (2.57×10 ⁻¹)	1.47 (3.82×10 ⁻¹)	1.08 (1.82×10 ⁻¹)	1.64 (3.53×10 ⁻¹)	1.36 (2.00×10 ⁻¹)	5.92×10⁻¹ (1.59×10⁻¹)
Avg Rank	7.15	5.59	4.21	5.18	3.09	5.37	3.18	2.18

表 5 噪声方差 $\sigma^2=0.75$ 时加性噪声测试函数下的 COST±STANDARDTable 5 COST±STANDARD under additive noise test function with noise variance $\sigma^2=0.75$

Function	DE	DE-RSF-TS	JDE	NRDE	EDA	DE-EDA	DEPSO	SEDADE
F1	2.08 (3.80×10 ⁻¹)	1.02 (3.19×10 ⁻¹)	8.83×10 ⁻¹ (1.95×10 ⁻¹)	1.07 (2.65×10 ⁻¹)	6.46×10 ⁻¹ (1.56×10 ⁻¹)	9.21×10 ⁻¹ (2.78×10 ⁻¹)	3.37×10 ⁻¹ (1.45×10 ⁻¹)	3.33×10⁻¹ (1.18×10⁻¹)
F2	2.02×10 (6.14×10 ⁻¹)	3.97 (3.67×10 ⁻¹)	4.25 (3.97×10 ⁻¹)	4.02 (6.07×10 ⁻¹)	2.12 (7.45×10 ⁻¹)	1.30×10 (8.23)	1.97 (4.53×10 ⁻¹)	5.45×10⁻¹ (1.97×10⁻¹)
F3	1.64×10 (1.58)	9.11×10 ⁻¹ (3.42×10 ⁻¹)	8.71×10 ⁻¹ (2.18×10 ⁻¹)	8.85×10 ⁻¹ (2.67×10 ⁻¹)	6.17×10 ⁻¹ (1.51×10 ⁻¹)	1.11 (3.92×10 ⁻¹)	5.73×10 ⁻¹ (3.03×10 ⁻¹)	3.61×10⁻¹ (1.54×10⁻¹)
F4	2.00 (4.27×10 ⁻¹)	9.83×10 ⁻¹ (2.81×10 ⁻¹)	8.58×10 ⁻¹ (2.53×10 ⁻¹)	1.02 (2.64×10 ⁻¹)	5.84×10 ⁻¹ (1.43×10 ⁻¹)	8.46×10 ⁻¹ (2.41×10 ⁻¹)	3.66×10 ⁻¹ (1.57×10 ⁻¹)	2.94×10⁻¹ (1.55×10⁻¹)
F5	2.02 (3.99×10 ⁻¹)	8.75×10 ⁻¹ (2.25×10 ⁻¹)	8.54×10 ⁻¹ (2.31×10 ⁻¹)	1.04 (3.18×10 ⁻¹)	6.87×10 ⁻¹ (1.50×10 ⁻¹)	9.34×10 ⁻¹ (1.97×10 ⁻¹)	3.63×10 ⁻¹ (1.70×10 ⁻¹)	3.49×10⁻¹ (1.11×10⁻¹)
F6	1.52×10 ² (8.90)	1.35×10 ² (2.26×10)	1.41×10 (2.04)	1.32×10 (1.95)	3.00×10 (6.81)	2.79×10 (7.42)	2.66×10 (1.50×10)	1.24 (1.46)
F7	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	5.00×10 ⁻¹ (9.72×10 ⁻¹)	0.00 (0.00)	1.00×10 ⁻¹ (3.16×10 ⁻¹)	0.00 (0.00)	0.00 (0.00)
F8	1.41 (6.26×10 ⁻¹)	9.63×10 ⁻¹ (3.80×10 ⁻¹)	1.02 (4.14×10 ⁻¹)	8.14×10 ⁻¹ (2.92×10 ⁻¹)	9.12×10 ⁻¹ (3.34×10 ⁻¹)	8.49×10 ⁻¹ (3.55×10 ⁻¹)	6.99×10⁻¹ (2.38×10⁻¹)	7.08×10 ⁻¹ (3.12×10 ⁻¹)
F9	4.72×10 ³ (3.24×10 ³)	2.29×10 (1.51)	2.10×10 (1.46)	2.17×10 (1.60)	1.62×10 (4.48×10⁻¹)	2.18×10 (3.76)	2.61×10 (1.53)	1.63×10 (3.99×10 ⁻¹)
F10	1.49×10 ² (9.61)	1.40×10 ² (2.27×10)	1.49×10 (2.60)	1.31×10 (1.91)	2.77×10 (6.13)	3.12×10 (1.10×10)	2.30×10 (1.58×10)	1.43 (1.84)
F11	2.94×10 (7.17×10 ⁻¹)	2.74×10 (8.19×10 ⁻¹)	2.72×10 (7.45×10⁻¹)	2.88×10 (7.31×10 ⁻¹)	2.81×10 (3.81×10 ⁻¹)	2.86×10 (9.78×10 ⁻¹)	2.79×10 (3.38×10 ⁻¹)	2.85×10 (4.85×10 ⁻¹)
F12	2.89 (4.95×10 ⁻¹)	1.85 (2.40×10 ⁻¹)	1.88 (2.91×10 ⁻¹)	2.02 (2.54×10 ⁻¹)	1.65 (1.43×10 ⁻¹)	1.94 (2.78×10 ⁻¹)	1.41 (1.88×10 ⁻¹)	1.35 (1.22×10⁻¹)
F13	1.20×10 ⁻¹ (1.25×10 ⁻¹)	1.32×10 ⁻¹ (1.08×10 ⁻¹)	1.20×10 ⁻¹ (8.00×10 ⁻²)	1.06×10 ⁻¹ (1.98×10 ⁻¹)	4.09×10⁻² (5.72×10⁻²)	1.41×10 ⁻¹ (1.18×10 ⁻¹)	4.94×10 ⁻² (3.28×10 ⁻²)	6.58×10 ⁻² (7.02×10 ⁻²)
F14	5.01×10 ⁻¹ (1.30×10 ⁻¹)	4.71×10 ⁻¹ (8.98×10 ⁻²)	4.99×10 ⁻¹ (5.68×10 ⁻²)	4.90×10 ⁻¹ (9.65×10 ⁻²)	4.92×10 ⁻¹ (4.61×10 ⁻¹)	4.96×10 ⁻¹ (1.60×10 ⁻²)	4.68×10 ⁻¹ (1.27×10 ⁻¹)	4.39×10⁻¹ (1.19×10⁻¹)
F15	1.73×10 ⁻¹ (1.62×10 ⁻¹)	1.38×10 ⁻¹ (1.46×10 ⁻¹)	1.52×10 ⁻¹ (1.48×10 ⁻¹)	2.22×10 ⁻¹ (2.53×10 ⁻¹)	5.61×10 ⁻² (8.84×10 ⁻²)	1.51×10 ⁻¹ (1.58×10 ⁻¹)	3.05×10⁻² (3.05×10⁻²)	1.39×10 ⁻¹ (1.17×10 ⁻¹)
F16	4.17 (4.66×10 ⁻¹)	2.14 (3.35×10 ⁻¹)	1.77 (2.82×10 ⁻¹)	1.74 (3.09×10 ⁻¹)	1.37 (2.39×10 ⁻¹)	2.06 (4.84×10 ⁻¹)	1.65 (2.52×10 ⁻¹)	6.52×10⁻¹ (1.69×10⁻¹)
Avg Rank	7.50	5.15	4.50	5.25	3.28	5.81	2.66	1.84

当噪声方差为 0.04 时,DE-RSF-TS 在 F11 上的性能最好,EDA 在 F13 上性能最好。DE-EDA 在 F9, F11 和 F12 上与 SEDADE 的性能相差不大, NRDE 在 F9, F10, F11 和 F12 上与 SEDADE 性能接近, DEPSO 在 F3 和 F15 上的性能最好。

当噪声方差为 0.5 时, JDE 在 F11 上的性能最好, EDA 在 F14 上的性能最好, EDA 和 SEDADE 在 F9 上性能差异不大, DEPSO 在 F8, F13, F15 上的性能最好。

当噪声方差为 0.75 时, JDE 在 F11 上性能最好, EDA 在 F9 和 F13 上的性能较好, NRDE 和 SEDADE 在 F10 和 F11 上性能差异不大, DEPSO 在 F8 和 F15 上的性能最好。

总体来看, EDA 和 DEPSO 算法在个别函数中具有优势,

说明 EDA 方法具有一定的抗噪能力, 此外, 将 DE 和 PSO 结合有利于算法适应进化的不同阶段。SEDADE 算法综合借鉴了这些算法的优点, 在大部分函数上性能优于其他算法, 并且根据平均排名可以看出, 在每种噪声强度下 SEDADE 算法排名均优于其他算法。

此外, 为了更直观地比较对比算法的收敛效果以及更详细地说明 SEDADE 的性能, 图 2 展示了 8 种算法在噪声方差为 0.25 时的对比收敛图, 选取每种噪声强度下呈现了 12 个测试函数的收敛对比图。从统计上看, SEDADE 算法在选取的 12 个测试函数中获得了较好的结果, 并且在收敛速度或最终解上优于其他算法, 特别是在 F6 和 F10 上改进效果明显。

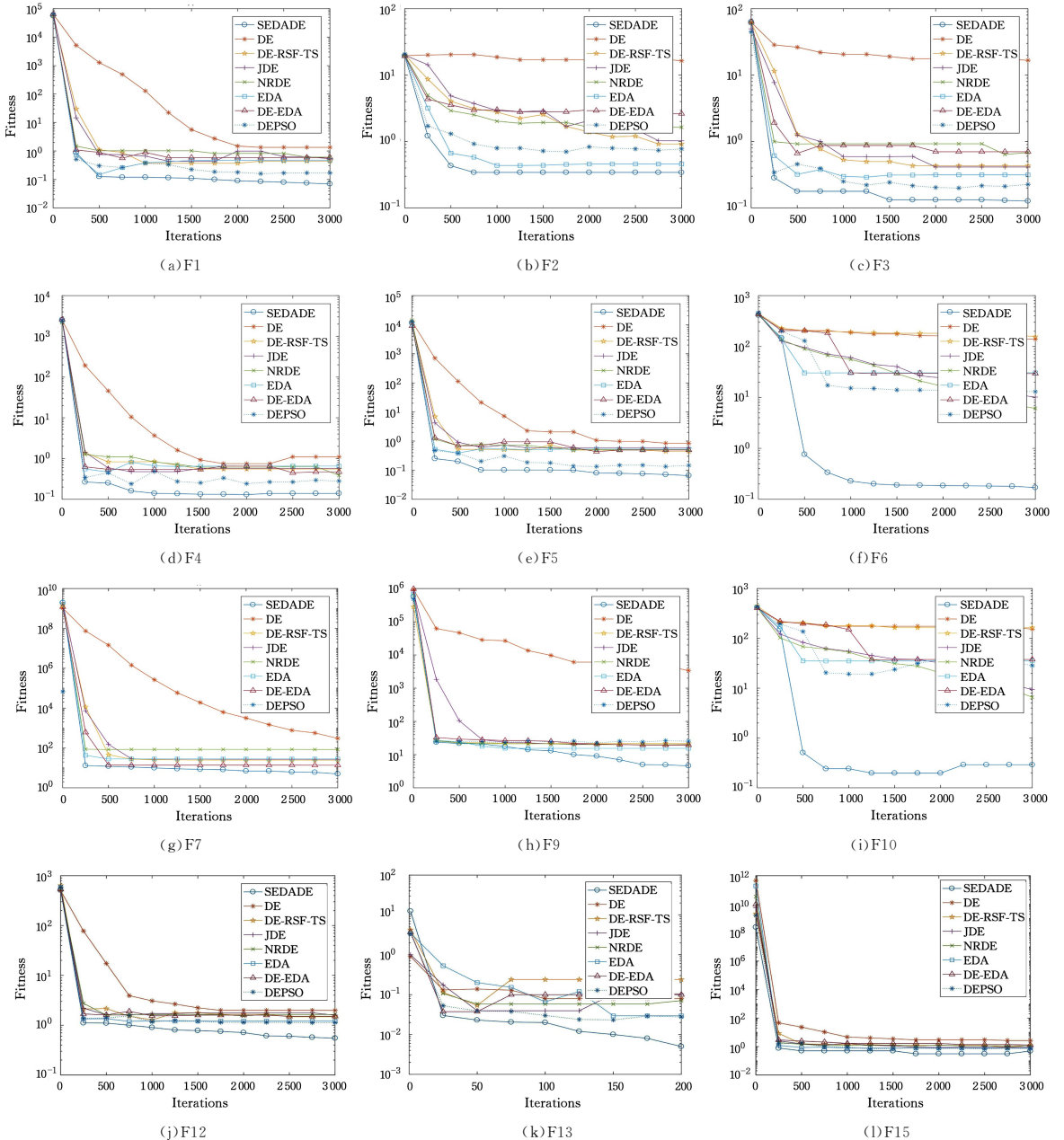


图 2 $\sigma^2 = 0.25$ 时测试实例的平均函数值与迭代次数

Fig. 2 Average function value and iteration number of test cases with noise intensity $\sigma^2 = 0.25$

为了更客观地比较两个算法的性能, 本文用 0.05 显著水平的 Wilcoxon 秩和检验对实验结果进行统计和比较。从表 6—表 8 的秩和检验结果可以看出, P-value 值均小于 0.05, 因此,

在显著水平为 0.05 的情况下, SEDADE 算法与其他对比算法有显著性差异。SEDADE 算法的性能优于其他对比算法, 在性能上具有一定的优势。

表6 噪声方差 $\sigma^2=0.04$ 时的秩和检验结果Table 6 Wilcoxon test results with noise variance $\sigma^2=0.04$

算法对比	R^+	R^-	P -value	0.05
SEDADE vs DE	118.0	2.0	0.000892	YES
SEDADE vs DE-RSF-TS	106.0	14.0	0.008266	YES
SEDADE vs JDE	106.0	14.0	0.008266	YES
SEDADE vs NRDE	134.0	2.0	0.000585	YES
SEDADE vs EDA	115.0	5.0	0.001621	YES
SEDADE vs DE-EDA	135.0	1.0	0.000482	YES
SEDADE vs DEPSO	110.5	25.5	0.026184	YES

表7 噪声方差 $\sigma^2=0.5$ 时的秩和检验结果Table 7 Wilcoxon test results with noise variance $\sigma^2=0.5$

算法对比	R^+	R^-	P -value	0.05
SEDADE vs DE	118.0	2.0	0.000892	YES
SEDADE vs DE-RSF-TS	108.0	12.0	0.005876	YES
SEDADE vs JDE	108.0	12.0	0.005876	YES
SEDADE vs NRDE	135.0	1.0	0.000482	YES
SEDADE vs EDA	102.0	18.0	0.015785	YES
SEDADE vs DE-EDA	133.0	3.0	0.000707	YES
SEDADE vs DEPSO	107.5	28.5	0.038607	YES

表8 噪声方差 $\sigma^2=0.75$ 时的秩和检验结果Table 8 Wilcoxon test results with noise variance $\sigma^2=0.75$

算法对比	R^+	R^-	P -value	0.05
SEDADE vs DE	120.0	0.0	0.000590	YES
SEDADE vs DE-RSF-TS	109.0	11.0	0.004932	YES
SEDADE vs JDE	109.0	11.0	0.004932	YES
SEDADE vs NRDE	136.0	0.0	0.000397	YES
SEDADE vs EDA	101.0	19.0	0.018421	YES
SEDADE vs DE-EDA	136.0	0.0	0.000397	YES
SEDADE vs DEPSO	96.0	24.0	0.038167	YES

结束语 进化算法已在各种工程场景中得到应用^[34],为了消除噪声对优化性能的影响,基于噪声在时间和空间分布的不变性和一致性假设,本文提出了SEDADE算法,一方面利用时间上某个体的多次重采样抵消噪声,另一方面利用解空间中不同个体的随机噪声信息互相抵消噪声,较好地地提高了算法对抗噪声的能力。具体地:(1)通过适应度值排名将种群划分为两个子种群,分别用基于空间去噪的EDA算法和基于时间去噪的DE算法进行进化;(2)对于由评价较差个体组成的子种群采用EDA算法,分布估计算法采用统计建模方式,在没有增加评价次数的情况下较好地抵消了噪声,同时较差个体的多样性有利于算法的探索;(3)对于由评价较好个体组成的子种群采用DE算法,引入基于时间的停滞重采样机制,它虽然增加了评价次数,但降低了噪声干扰,增加了种群多样性,缩短了劣质解在进化种群中的生命周期;(4)通过基于概率选择的EDA信息利用机制,利用EDA搜索到的全局信息,为DE的进一步搜索提供参考,避免陷入局部最优,提高最优解的精度。在噪声环境下,SEDADE算法对比其他优秀的进化算法表现出了一定的优势,通过消融实验验证了提出的机制的有效性。

后续还会考虑复杂噪声的情况,如乘性噪声、泊松噪声以及混合噪声等情况。另外,如果噪声分布随时间和空间变化,则其对过程的影响会更加复杂化,这些都是值得深入研究的问题。

参考文献

- [1] VIKHAR P A. Evolutionary algorithms: A critical review and its future prospects[C]// International Conference on Global Trends in Signal Processing, Information Computing and Communication(ICGTSPIC). IEEE, 2016: 261-265.
- [2] TAU Y H S. Optimization of power delivery design and methodologies[C]// International Conference on Electronic Materials and Packaging. IEEE, 2006: 1-3.
- [3] LI Z, ZHANG S, CAI X, et al. Noisy Optimization by Evolution Strategies With Online Population Size Learning [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(9): 5816-5828.
- [4] CHIU S Y, LIN C N, LIU J, et al. Differential evolution for strongly noisy optimization: Use 1.01 n resamplings at iteration n and reach the 1/2 slope[C]// IEEE Congress on Evolutionary Computation(CEC). IEEE, 2015: 338-345.
- [5] GUO D, NIE Z, YAN L. The application of noise-tolerant ZD design formula to robots' kinematic control via time-varying nonlinear equations solving[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 48(12): 2188-2197.
- [6] JING B B, GUO J, WANG L Q, et al. Image double blind denoising algorithm combining with denoising convolutional neural network and conditional generative adversarial net[J]. Journal of Computer Applications, 2021, 41(6): 1767-1774.
- [7] RAKSHIT P, KONAR A, DAS S, et al. Uncertainty Management in Differential Evolution Induced Multiobjective Optimization in Presence of Measurement Noise[J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2017, 44(7): 922-937.
- [8] MERELO J J, LIBERATORE F, ARES A F, et al. There is Noisy Lunch: A Study of Noise in Evolutionary Optimization Problems[C]// International Joint Conference on Computational Intelligence. IEEE, 2016.
- [9] JIN Y, BRANKE J. Evolutionary optimization in uncertain environments—a survey [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(3): 303-317.
- [10] CAPONIO A, NERI F. Differential Evolution with Noise Analyzer[C]// Workshops on Applications of Evolutionary Computation. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 715-724.
- [11] MININNO E, NERI F. A memetic Differential Evolution approach in noisy optimization [J]. Memetic Computing, 2010, 2(2): 111-135.
- [12] DI PIETRO A, WHILE L, BARONE L. Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions[C]// Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753). IEEE, 2004: 1254-1261.
- [13] MARKON S, ARNOLD D V, BACK T, et al. Thresholding—a selection operator for noisy ES[C]// Proceedings of the 2001 Congress on Evolutionary Computation. IEEE, 2001: 465-472.
- [14] DAS S, KONAR A, CHAKRABORTY U K. Improved differential evolution algorithms for handling noisy optimization problems[C]// 2005 IEEE Congress on Evolutionary Computation.

- IEEE,2005;1691-1698.
- [15] DAS S,KONAR A. An Improved Differential Evolution Scheme for Noisy Optimization Problems[C]//International Conference on Pattern Recognition & Machine Intelligence. Springer-Verlag,2005.
- [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] RAHNAMEYAN S,TIZHOOSH H R,SALAMA M M A. Opposition-based differential evolution for optimization of noisy problems[C]//2006 IEEE International Conference on Evolutionary Computation. IEEE,2006;1865-1872.
- [18] GHOSH A,DAS S,PANIGRAHI B K,et al. A noise resilient differential evolution with improved parameter and strategy control[C]//IEEE congress on evolutionary computation(CEC). IEEE,2017;2590-2597.
- [19] ZHANG J,ZHU X,WANG Y,et al. Dual-Environmental Particle Swarm Optimizer in Noisy and Noise-Free Environments [J]. IEEE Transactions on Cybernetics, 2019, 49 (6): 2011-2021.
- [20] ZHAO Q,GAO Y. A new Algorithm based on the Gbest of Particle Swarm Optimization algorithm to improve Estimation of Distribution Algorithm[C]//2018 International Conference on Smart Computing and Electronic Enterprise(ICSCEE). IEEE, 2018;1-5.
- [21] QIU L Q,LIANG Y Q,FAN J C. A Dynamic Fusion Parallel Hybrid Evolutionary Algorithm EDAs/PSO [J]. Computer Application and Software,2014,31(6):271-274.
- [22] SUN J,ZHANG Q,TSANG E P K. DE/EDA: A new evolutionary algorithm for global optimization [J]. Information Sciences,2005,169(3/4):249-262.
- [23] STORN R,PRICE K. Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces[J]. Journal of Global Optimization,1995,23(1):1-12.
- [24] DAS S,SUGANTHAN P N. Differential evolution: A survey of the state-of-the-art [J]. IEEE Transactions on Evolutionary Computation,2010,15(1):4-31.
- [25] DAS S,MULLICK S S,SUGANTHAN P N. Recent advances in differential evolution-an updated survey[J]. Swarm and Evolutionary Computation,2016,27:1-30.
- [26] MUHLENBEIN H. The equation for response to selection and its use for prediction[J]. Evolutionary Computation,1997,5(3): 303-346.
- [27] OCHOA A. EBBA — Evolutionary best basis algorithm[C]// Proceedings of the Second International Symposium on Adaptive Systems(ISAS 99),Havana,Cuba. 1999;93-98.
- [28] LARRANAGA P,LOZANO J A. Estimation of distribution algorithms: A new tool for evolutionary computation[M]. Boston: Kluwer Press,2002.
- [29] ZHANG B T. A Bayesian Framework for Evolutionary Computation[C]//Proceedings of the 1999 Congress on Evolutionary Computation(CEC99). 1999;722-728.
- [30] BALUJA S. Population-based incremental learning, a method for integrating genetic search based function optimization and competitive learning[R]. Carnegie-Mellon Univ Pittsburgh Pa Dept of Computer Science,1994.
- [31] RAKSHIT P,KONAR A,DAS S. Noisy evolutionary optimization algorithms—a comprehensive survey[J]. Swarm and Evolutionary Computation,2017,33:18-45.
- [32] KRINK T,FILIPIC B,FOGEL G B. Noisy optimization problems—a particular challenge for differential evolution? [C]// Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753). IEEE,2004,1:332-339.
- [33] GHOSH A,DAS S,MALLIPEDDI R,et al. A modified differential evolution with distance-based selection for continuous optimization in presence of noise[J]. IEEE Access,2017,5:26944-26964.
- [34] GUO X,YANG Q,ZHENG H,et al. Optimization of power distribution for electrothermal anti-icing systems by differential evolution algorithm [J]. Applied Thermal Engineering, 2023, 221:119875.



WANG Bin, born in 1971, associate professor, master supervisor, is a senior member of CCF(No. 14802S). His main research interests include evolutionary computation and artificial intelligence.



JIN Haiyan, born in 1976, professor, Ph. D supervisor, is an outstanding member of CCF (No. 74449M). Her main research interests include artificial intelligence theory and its applications and so on.

(责任编辑:何杨)