控制与决策
Control and Decision

基于邻域交叉的双变异差分进化算法求解非线性方程组

赵世杰, 赵秋丽, 陈淼, 崔倩倩

引用本文:
赵世杰, 赵秋丽, 陈淼, 等. 基于邻域交叉的双变异差分进化算法求解非线性方程组[J]. 控制与决策, 2025, 40(2): 546-552.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2024.0194

基于邻域交叉的双变异差分进化算法求解非线性方程组

赵世杰[1,2†]，赵秋丽[1]，陈　淼[1]，崔倩倩[1]

(1. 辽宁工程技术大学 理学院，辽宁 阜新 123000；
2. 辽宁工程技术大学 运筹与优化研究院，辽宁 阜新 123000)

摘　要: 非线性方程组问题的求解难点在于多根联解的同步解出, 针对邻域拥挤差分进化算法存在的多根解出不完整、丢根及易陷入局部最优等问题, 提出一种基于邻域交叉的双变异差分进化算法. 双变异策略基于个体适应度值综合学习邻域和全局的进化信息, 以提高种群多样性并同步增强其局部最优规避性能; 邻域交叉策略通过种群分组与不同交叉操作实现进化个体的差异性引导, 以规避多根的联解丢失并改善计算资源的利用效率. 实验结果表明, 所提算法能够有效实现非线性方程组的多根联解, 且在找根率和成功率指标上表现优异.

关键词: 智能优化算法；非线性方程组；差分进化算法；双变异策略；邻域交叉策略；种群多样性；多根联解

中图分类号: TP18　　　**文献标志码:** A

DOI: 10.13195/j.kzyjc.2024.0194

引用格式: 赵世杰, 赵秋丽, 陈淼, 等. 基于邻域交叉的双变异差分进化算法求解非线性方程组 [J]. 控制与决策, 2025, 40(2): 546-552.

Solving nonlinear equation systems with neighborhood crossover-based dual-mutation differential evolution algorithm

ZHAO Shi-jie[1,2†]，*ZHAO Qiu-li*[1]，*CHEN Miao*[1]，*CUI Qian-qian*[1]

(1. College of Science，Liaoning Technical University，Fuxin 123000，China；2. Institute for Optimization and Decision Analytics，Liaoning Technical University，Fuxin 123000，China)

Abstract: The difficulty of solving nonlinear equation systems (NESs) lies in how to achieve the synchronous parsing of multiple-root joint solution. Since neighborhood-based crowding differential evolution algorithm has many problems such as incomplete solution of multiple roots, loss of roots and easy to fall into local optimality, a neighborhood crossover-based dual-mutation differential evolution algorithm is proposed. The dual-mutation strategy is based on individual fitness values to learn both neighborhood and global evolutionary information, which results in improving population diversity and simultaneously enhancing the local optimum avoidance performance. The neighborhood crossover strategy employs the population grouping mechanism and different crossover operations to achieve differential guidance of evolutionary individuals, which contributes to avoiding the loss of joint solution of multiple roots and improving the computing resources utilization efficiency. The experimental results show that the proposed algorithm can effectively realize the multi-root joint solution of NESs and has outstanding capacity on the index of root rate and success rate.

Keywords: intelligence optimization algorithm；nonlinear equation systems；differential evolution algorithm；dual-mutation strategy；neighborhood crossover strategy；population diversity；multiple-root joint solution

0 引　言

非线性方程组 (nonlinear equation systems, NESs)[1] 问题广泛存在于数据挖掘、机械制造、模式识别等领域, 其多根联解是一个重要的研究课题. 目前 NESs 求解算法[2] 可分为梯度型算法和智能优化算法. 囿于牛顿法[3]、同伦延拓法[4] 等梯度型算法求解 NESs 时往往存在初值敏感、单次运行只可寻得一个根等局限, 智能优化算法 (如遗传算法) 因其初值不敏感性及良好并行寻优能力[5], 较适于 NESs 问题的多根联解且已成为当前 NESs 智能求解的热点议题.

相关的 NESs 智能求解研究中, 差分进化算法 (differential evolution, DE) 的应用最为广泛. 但囿于其选择压力的存在, 需要融入排斥、小生境等多样性

收稿日期: 2024-02-27；**录用日期:** 2024-07-12.
基金项目: 辽宁省教育厅基本科研项目 (JYTMS20230802)；辽宁省自然科学基金面上项目 (2023-MS-317)；辽宁省研究生教育教学改革研究资助项目 (LNYJG2023119).
†通讯作者. E-mail: zhaoshijie@lntu.edu.cn.

保持机制, 以实现 DE 算法对 NESs 问题的同步多根联解能力. Gong 等[6] 和 Liao 等[7] 分别结合加法排斥和动态排斥策略提出针对 NESs 问题的改进型差分进化算法; He 等[8] 改进邻域策略并提出一种基于模糊邻域的定向差分进化算法 (fuzzy neighborhood-based differential evolution with orientation, FNODE); Liao 等[9] 联合分解、子种群控制策略提出基于分解的重新初始化差分进化算法; Qu 等[10] 将邻域变异策略融入拥挤差分进化算法提出基于邻域拥挤的差分进化算法 (neighborhood-based crowding DE, NCDE); Wu 等[11] 融入小生境与聚类策略构建基于 *K*-means 物种形成的差分进化算法 (*K*-means species-based DE, KSDE); Li 等[12] 结合种群多样性动态捕获等机制提出资源再分配自适应双小生境差分进化算法.

由于 NESs 问题的多根复杂性, 小生境型 DE 算法虽已取得一些成效, 但仍需针对特定问题展开特定研究. 如 NCDE 算法[10], 作为一种高效的多模态优化算法, 虽然可用于 NESs 多根求解, 但却存在种群多样性欠佳、求根个数不完整、丢根、易陷入局部最优等问题. 据此, 本文提出一种基于邻域交叉的双变异差分进化算法, 并用于 NESs 问题的智能求解. 实验结果表明, 所提算法有效改善了 NESs 问题的多根联解性能.

1 非线性方程组

NESs 一般由n个变量和m个方程$f_j(\cdot)$组成, 且至少有一个非线性方程, 其定义式为

$$\begin{cases} f_1(x_1,\dots,x_n)=0,\\ f_2(x_1,\dots,x_n)=0,\\ \quad\vdots\\ f_m(x_1,\dots,x_n)=0. \end{cases} \tag{1}$$

其中: $x=(x_1,\dots,x_n)\in S$表示n维决策向量, $S\subseteq \mathbb{R}^n$表示决策空间, 且$S=\prod_{i=1}^{n}[L_i,U_i]$, L_i和U_i分别表示第i维变量x_i的下界和上界; $f_{j\in 1,\dots,m}(x_1,\dots,x_n)=0$表示方程组的第$j$个方程.

在利用智能优化算法对 NESs 问题进行求根时, 参考文献 [13] 将其转换为最小化单目标优化问题

$$\min \varGamma=\sum_{j=1}^{m} f_j^2(x), \tag{2}$$

其中$f_j(x)$表示 NESs 问题中第j个方程的函数值.

2 基于邻域交叉的双变异差分进化算法

NESs 多根智能求解要求智能优化算法在种群多样性保持的同时协同提高计算资源利用效率. NCDE 算法[10] 的个体变异时只学习了邻域局部信息而非全局信息, 导致其种群多样性弱化, 并诱发了对 NESs 多根锚定概率的降低; 同时, NCDE 因其特定邻域结构导致陷入局部极值的个体难以跳出局域, 不仅无益于 NESs 多根联解, 且必然会造成计算资源的浪费. 因此, 本文为同步协调种群多样性和计算资源利用效率, 适定性构建双变异策略和邻域交叉策略, 并融合性提出一种基于邻域交叉的双变异差分进化算法 (neighborhood crossover-based dual-mutation differential evolution algorithm, NCD_DE), 以改善算法求解 NESs 问题的多根联解性能.

2.1 双变异策略

为较好平衡 NCDE 算法并行寻优过程中的探索和开采能力, 提高其种群多样性及跳出局部最优的性能, 基于适应度信息在变异策略 DE/rand/1[10] 基础上提出一种综合学习邻域和全局信息的双变异策略.

设个体适应度值的临界判定阈值为δ, 记第i个个体的适应度值为$\varGamma_i$, 参照文献 [14] 设置$\delta=0.5$.

1) 当$\varGamma_i\geqslant\delta$时, 变异中间体$\nu_i$的计算式为

$$\nu_i(g+1)=x_{r_1}(g)+F\cdot(x_{r_2}(g)-x_{r_3}(g)). \tag{3}$$

其中: $x_{r_1}\in \mathrm{pop}$, pop表示当前整体种群; $F\in[0,2]$表示缩放因子; $x_{r_2}, x_{r_3}\in \mathrm{subpop}_i$, subpop_i表示当前个体x_i的邻域子群; $r_1\neq r_2\neq r_3\neq i$.

2) 当$\varGamma_i<\delta$时, 为增强个体x_i的收敛效率及跳出局部收敛域性能, 变异中间体ν_i的计算式为

$$\nu_i(g+1)=x_i(g)+F\cdot(x_{r_1}(g)-x_{r_2}(g))+ \\ \varGamma_i\cdot(x_{r_3}(g)-x_{r_4}(g)). \tag{4}$$

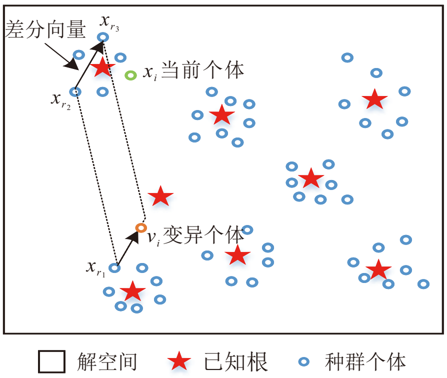
其中: $x_{r_1}(g), x_{r_2}(g)\in \mathrm{subpop}_i$; $x_{r_3}(g), x_{r_4}(g)\in \mathrm{pop}$; $r_1\neq r_2\neq r_3\neq r_4\neq i$. 注意, 当$x_i(g)$的邻域子群逼近全局最优位置$x^*$时, $\varGamma_i$逼近于 0, 故式 (4) 中项$\varGamma_i\cdot(x_{r_3}(g)-x_{r_4}(g))$近乎可忽略不计, 此时种群趋于收敛; 当$x_i(g)$的邻域子群临近局部最优位置时, $\varGamma_i$并未趋于 0, 项$\varGamma_i\cdot(x_{r_3}(g)-x_{r_4}(g))$的作用不可忽略, 且因$x_{r_3}(g)$, $x_{r_4}(g)\in \mathrm{pop}$, 此时个体倾向于搜索整个可行域而具备跳出局部最优的能力.

综上可得基于适应度差异判定的双变异策略为

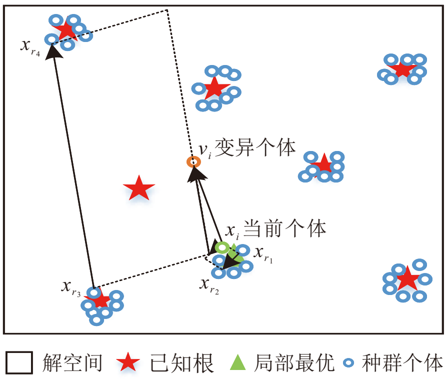
$$v_i(g+1)= \\ \begin{cases} x_{r_1}(g)+F\cdot(x_{r_2}(g)-x_{r_3}(g)),\ \varGamma_i\geqslant\delta;\\ x_i(g)+F\cdot(x_{r_1}(g)-x_{r_2}(g))+\\ \varGamma_i\cdot(x_{r_3}(g)-x_{r_4}(g)),\ \varGamma_i<\delta. \end{cases} \tag{5}$$

图 1 为基于双变异策略的变异中间体v_i进化示意. 由图 1(a) 可知, 变异中间体v_i同时学习个体x_i的邻域差分信息和全局随机个体的位置信息, 以增强进化种群多样性及多根联解性能; 由图 1(b) 可知, 进化种群随迭代越趋聚集于多个根附近, 变异中间体

v_i越保持对邻域和全局差分信息的概率性学习, 以实现算法多模同步收敛性及跳出局部极值的能力.



(a) 进化前期时v_i的位置更新



(b) 陷入局部最优时v_i的位置更新

图1 基于双变异策略的变异中间体v_i进化示意

2.2 邻域交叉策略

为进一步强化进化种群的多样性及计算资源的同步协调性, 提出基于种群分组与不同交叉操作的邻域交叉策略. 种群根据适应度升序, 按比例ω分为优良、一般和较差的群组, 并记为SP(SuperiorPop)、MP(MiddlePop)、IP(InferiorPop), 3 个群组内的个体x_i分别继承其对应变异中间体ν_i或邻域内最优个体$x^{\text{in}}_{\text{best}}$、随机个体$x^{\text{in}}_{\text{rand}}$及邻域外随机个体$x^{\text{out}}_{\text{rand}}$的维度信息, 依概率交叉进化并产生子代个体$u_i$.

1) 若$x_i \in \text{SP}$, 则子代个体u_i继承变异中间体ν_i与邻域最优个体$x^{\text{in}}_{\text{best}}$的维度信息, 计算式为

$$u_{i,j}(g+1)=\begin{cases}v_{i,j}(g+1),\ r\leqslant \text{CR or } j=j_{\text{Rand}};\\ x^{\text{in}}_{\text{best},j}(g),\ \text{otherwise}.\end{cases}\tag{6}$$

其中: r为区间$[0,1]$内的一个随机数; $\text{CR}\in[0,1]$为交叉概率; $j_{\text{Rand}}\in\{1,2,\ldots,n\}$为一个随机整数, 以保证$u_i$至少有一个维度继承自$v_i$. 该交叉模式以邻域最优信息的依概率学习, 改善算法的收敛效率.

2) 若$x_i \in \text{MP}$, 则子代个体u_i继承变异中间体ν_i与邻域随机个体$x^{\text{in}}_{\text{rand}}$的维度信息, 计算式为

$$u_{i,j}(g+1)=\begin{cases}v_{i,j}(g+1),\ r\leqslant \text{CR or } j=j_{\text{Rand}};\\ x^{\text{in}}_{\text{rand},j}(g),\ \text{otherwise}.\end{cases}\tag{7}$$

其中: $x^{\text{in}}_{\text{rand}}\in \text{subpop}_i$, 且$\text{rand}\neq i$. 该交叉模式继承邻域随机个体信息, 兼顾算法收敛性和种群多样性.

3) 若$x_i \in \text{IP}$, 则子代个体u_i继承变异中间体ν_i与邻域外随机个体$x^{\text{out}}_{\text{rand}}$的维度信息, 计算式为

$$u_{i,j}(g+1)=\begin{cases}v_{i,j}(g+1),\ r\leqslant \text{CR or } j=j_{\text{Rand}};\\ x^{\text{out}}_{\text{rand},j}(g),\ \text{otherwise}.\end{cases}\tag{8}$$

其中$x^{\text{out}}_{\text{rand}}\in \text{pop}-\text{subpop}_i$. 该交叉模式继承邻域外随机个体的全局信息, 可较好增加其种群多样性.

综上可得, 基于适应度分组的邻域交叉策略为

$$u_{i,j}(g+1)=\begin{cases}\begin{cases}v_{i,j}(g+1),\ r\leqslant \text{CR or } j=j_{\text{Rand}};\\ x^{\text{in}}_{\text{best},j}(g),\ \text{otherwise};\end{cases}\\ \quad \text{if } x_i\in\text{SP}.\\ \begin{cases}v_{i,j}(g+1),\ r\leqslant \text{CR or } j=j_{\text{Rand}};\\ x^{\text{in}}_{\text{rand},j}(g),\ \text{otherwise};\end{cases}\\ \quad \text{if } x_i\in\text{MP}.\\ \begin{cases}v_{i,j}(g+1),\ r\leqslant \text{CR or } j=j_{\text{Rand}};\\ x^{\text{out}}_{\text{rand},j}(g),\ \text{otherwise};\end{cases}\\ \quad \text{if } x_i\in\text{IP}.\end{cases}\tag{9}$$

此外, 为有效增强算法的局部开采性能, 当个体适应度值$\varGamma_i<\delta$时, 选用式 (6) 的邻域交叉模式.

2.3 基于邻域交叉的双变异差分进化算法

将双变异策略和邻域交叉策略融入 NCDE 算法, 提出用于 NESs 问题的 NCD_DE 算法, 其伪代码如下.

输入: 种群规模NP、评估次数Max_FES、决策空间S、变量维数n;

输出: 最终找到的 NESs 问题的多个根向量 $\boldsymbol{x}$*.

1) // 随机生成NP个种群个体

2) // 计算种群个体的适应度值

3) While FES < Max_FES

4) //种群基于适应度值升序排列

5) for $i=1:\text{NP}$

6) // 种群按分组比例分为 SP、MP、IP 三个群组

7) if $x_i\in\text{SP}$ or $\varGamma_i<\delta$

8) // 个体x_i选择式 (6) 的交叉模式

9) else if $x_i\in\text{MP}$

10) // 个体x_i选择式 (7) 的交叉模式

11) else

12)　　// 个体x_i选择式 (8) 的交叉模式

13)　　end if

14)　　// 个体x_i与欧氏距离最近的$M-1$个个体组成邻域子群subpop$_i$, $M=\lfloor 5+5\cdot((\text{Max\_FES}-\text{FES})/\text{Max\_FES})\rfloor$, $\lfloor\cdot\rfloor$表示向下取整

15)　　if $\Gamma_i<\delta$

16)　　// 个体x_i选择式 (4) 的变异模式

17)　　else

18)　　// 个体x_i选择式 (3) 的变异模式

19)　　end if

20)　　// 个体x_i按变异交叉模式生成子代u_i

21)　end for

22)　for $i=1$ to NP

23)　// 计算子代u_i适应度值

24)　　FES = FES + 1

25)　end for

26)　for $i=1$ to NP

27)　// 计算u_i与当前种群pop所有个体的欧氏距离, 找到与u_i欧氏距离最近的父代个体 (记为x) 并与u_i比较适应度, 选择优者更新父代个体x

28)　// 若x满足根要求, 存入$X_{\rm save}$并重新初始化

29)　end for

30) end while

31) //满足根条件的种群个体与存储的$X_{\rm save}$合并, 再按 3.2 节个体满足条件筛定最终根并组成 ***x****

32) return ***x****

2.4　NCD_DE 算法的理论分析

1) 时间复杂度分析.

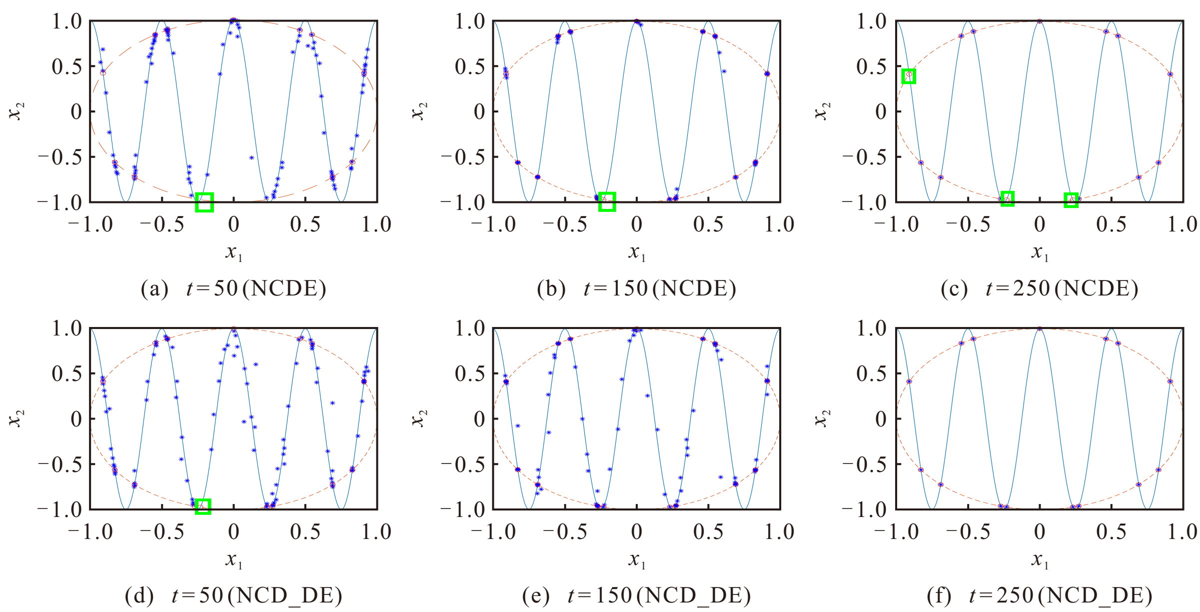
NCD_DE 算法的复杂度主要受种群规模NP、最大迭代次数T、变量维度n影响. 复杂度运算主要有初始化$O(\text{NP}\cdot n)$, 计算初始适应度$O(\text{NP})$, 种群升序排列$O(T\cdot\text{NP}\cdot\log\text{NP})$, 选择交叉模式$O(T\cdot\text{NP})$, 组成邻域子群的计算、排序和确定$O(T\cdot\text{NP}\cdot(\text{NP}\cdot n+\text{NP}\cdot\log\text{NP}+M-1))$, 选择变异算子$O(T\cdot\text{NP})$, 变异$O(T\cdot\text{NP}\cdot n)$, 交叉生成子代$O(T\cdot\text{NP}\cdot n)$, 计算子代适应度$O(T\cdot\text{NP})$, 计算欧氏距离并确定最近父代个体及其适应度优者更新$O(T\cdot\text{NP}\cdot(\text{NP}\cdot n+\text{NP}+1))$, 父代个体的根判断、存储及重新初始化$O(T\cdot\text{NP}\cdot N\cdot n+H\cdot(1+n))$($H$为存储的个体数), 循环迭代后种群个体的与根距离计算、最近确定和保存$O(N\cdot(\text{NP}\cdot n+\text{NP})+K)$ (N为已知根数目, $K\leqslant N$为满足根条件的个体数), 合并候选根的个体满足计算、最近确定和最终根筛定$O(N\cdot((K+H)\cdot n+(K+H))+Q)$ (Q为最终筛定的根数目).

综上所述, 在忽略低阶项时, NCD_DE 算法的时间复杂度为$O(T\cdot\text{NP}^2\cdot\max\{n,\log\text{NP}\})$, 且与改进相对的 NCDE 算法的时间复杂度一致.

2) 寻根有效性分析.

为测验 NCD_DE 算法对 NESs 多根联解性能的有效性, 以 3.1 节F_2方程组为例, 分别记录 NCDE 算法和 NCD_DE 算法的寻根演化过程, 对比见图 2 (圆圈为F_2的根, 星号为种群个体, 二者重合表示算法找到的根, 带方框的圆圈表示算法未找到的根).

由图 2 可知: 迭代演化初期, NCDE 和 NCD_DE 算法的进化种群已呈现向 NESs 多个根附近的聚集现象, 且均有一个根未成功锚定, 但 NCD_DE 的进



(a)　$t=50$ (NCDE)　(b)　$t=150$ (NCDE)　(c)　$t=250$ (NCDE)

(d)　$t=50$ (NCD_DE)　(e)　$t=150$ (NCD_DE)　(f)　$t=250$ (NCD_DE)

图2　NCDE 算法和 NCD_DE 算法对 NESs 问题的寻根演化过程对比示意 (t表示进化次数)

化个体在搜索空间内散布较均匀, 且具备良好种群多样性. 迭代演化中期, NCDE 算法的进化个体几乎已全部聚集到多个根附近, 但却并未捕捉到演化初期的未锚定根; NCD_DE 算法不仅成功锚定并聚集在 NESs 所有根的附近, 且进化个体仍持续保持在搜索空间的散布和全局探索能力. 迭代演化末期 (仅绘制最终寻得根的情况), NCDE 算法不仅仍未找到先前的未锚定根, 甚至出现了两个中期聚集根的丢根现象, NCD_DE 算法则成功找到 NESs 问题的全部 15 个根, 这得益于双变异策略的多样性保持机制、邻域交叉策略的计算资源良好利用以及历史所得根的存储机制, 有效验证了所提 NCD_DE 算法的良好寻根性能.

3 实验与结果

3.1 测试函数描述

为测验 NCD_DE 算法多根联解性能, 以 6 个不同特征的 NESs 测试函数[7] 进行实验, 信息如下.

$$F_1:\ \begin{cases} x_1^2-x_2^2=0; \\ 1-|x_1-x_2|=0. \end{cases}$$

其中: $x_i\in[-3,3]\,, i=1,2$. 该方程组有 2 个根.

$$F_2:\ \begin{cases} x_1-\cos(4\pi x_2)=0; \\ x_1^2+x_2^2-1=0. \end{cases}$$

其中: $x_i\in[-1,1]\,, i=1,2$. 该方程组有 15 个根.

$$F_3:\ \begin{cases} 0.04\Big(\dfrac{11}{15}-x_1\Big)\mathrm{e}^{10x_1/(1+x_1/100)}-x_1=0; \\ 0.04(2.2-2x_1-3x_2)\mathrm{e}^{10x_2/(1+x_2/100)}+ \\ x_1-3x_2=0. \end{cases}$$

其中: $x_i\in[0,1]\,, i=1,2$. 该方程组有 7 个根.

$$F_4:\ x_i-\cos\Big(2x_i-\sum_{j=1}^{n}x_j\Big)=0,\ i=1,2,\dots,n.$$

其中: $x_i\in[-1,1]\,, n=3$. 该方程组有 7 个根.

$$F_5:\ \begin{cases} 4x_1^3-3x_1-x_2=0; \\ x_1^2-x_2=0. \end{cases}$$

其中: $x_1\in[-5,1.5]\,, x_2\in[0,5]$. 该方程组有 3 个根.

$$F_6:\ \begin{cases} x_1^2-x_1-x_2^2-x_1+x_3^2=0; \\ \sin(x_2-\exp(x_1))=0; \\ x_3-\log|x_2|=0. \end{cases}$$

其中: $x_1\in[0,2]\,, x_2\in[-10,10]\,, x_3\in[-1,1]$. 该方程组有 5 个根.

3.2 评价指标与参数设置

实验以找根率 (root rate, RR)、成功率 (success rate, SR)[6] 作为算法对 NESs 多根联解性能的评价指标, 以 NCDE[10]、KSDE[11]、CADE[13]、FNODE[8]、ANDE[15] 和 self_CCDE[16] 为对比算法, NCDE 参数按照文献 [11-13] 设定, 其他算法参数均按原文献设置. 为保证算法性能评价的公平性, 各算法均采用统一的种群规模 NP $=100$和评价次数 Max_FES (F_1设为 5000, $F_2\sim F_6$设为 25000) 独立运行 30 次实验.

此外, NESs 寻根性能参数设置情况 (个体满足): 1) 候选根精度阈值$\theta=10^{-6}$; 2) 个体x_i与根的距离小于等于$\varepsilon=0.01$, 则认为个体x_i是 NESs 的根.

3.3 实验结果与分析

1) 调控参数的性能影响实验.

NCD_DE 算法的 NESs 多根联解性能受双变异策略的适应度阈值δ和邻域交叉策略的分组比例ω影响. 将仅融入双变异策略和邻域交叉策略的算法记为 NCD_DE-D 和 NCD_DE-N. 当δ较大时, NCD_DE-D 算法难以保持高种群多样性, 并降低了找到多个根的概率; 当δ较小时, 其局部开采性能较低, 且难以定位到高精度的根. 因此, 为平衡 NCD_DE-D 的多根联解与根的高精度锚定性能, 参考文献 [14] 设置$\delta=0.5$. 另一方面, 为测验种群不同分组比例ω(优良、一般和较差) 对 NCD_DE-N 算法的性能影响, 分别设定ω为 1: 1: 1(按 33: 33: 34 设置)、1: 1: 2、1: 1: 3、1: 2: 1、1: 2: 2、1: 3: 1、2: 1: 1、2: 1: 2、2: 2: 1 和 3: 1: 1, 共 10 种情形, 对应算法记为 NCD_DE-1, ..., NCD_DE-10, 各算法独立运行 30 次. 此实验的 Friedman 检验的秩次排名平均结果见表 1 (秩次排名值越小表示对应算法性能越优异).

表1 不同ω比例下 NCD_DE 算法的 Friedman 检验秩次排名

算法	秩次排名(RR)	秩次排名(SR)
NCD_DE-1	4.0000	4.1667
NCD_DE-2	2.6667	2.5000
NCD_DE-3	4.0000	4.0000
NCD_DE-4	3.0000	3.0000
NCD_DE-5	3.8333	4.0000
NCD_DE-6	3.5000	3.8333
NCD_DE-7	3.0000	3.1667
NCD_DE-8	4.1667	4.1667
NCD_DE-9	4.5000	5.1667
NCD_DE-10	**1.6667**	**1.6667**

由表 1 可知, 不同ω比例对 NCD_DE-N 的性能影响有所差异, 以 NCD_DE-10 算法的秩次排名双最优, 故后续实验均按$\omega=3$: 1: 1 设定 NCD_DE 算法.

2) 改进策略的有效性实验.

为验证所提 NCD_DE 算法的两种改进策略及其融合有效性, 以 6 组 NESs 函数且以 NCDE 算法为对比进行消融实验, 30 次实验的对比统计结果见

表 2 和表 3. 其中: * 表示改进策略优于 NCDE; 粗体表示指标值绝对占优, +/–/ = 分别表示所提算法的评价指标值优于/劣于/等于对比算法的指标.

表2　各算法 RR 指标的对比统计结果

函数	NCDE	NCD_DE-D	NCD_DE-N	NCD_DE
F_1	0.4500	0.4500	0.7167*	**0.7667**
F_2	0.8844	0.9600*	0.9511*	**0.9822**
F_3	0.8762	0.9952*	0.9619*	**1.0000**
F_4	0.9667	0.9952*	0.9952*	**1.0000**
F_5	0.8444	**1.0000**	0.9889*	**1.0000**
F_6	0.9933	0.9933	0.9867	**1.0000**
+/–/ =		4/0/2	5/1/0	6/0/0

表3　各算法 SR 指标的对比统计结果

函数	NCDE	NCD_DE-D	NCD_DE-N	NCD_DE
F_1	0.2000	0.2667*	0.5000*	**0.6000**
F_2	0.2000	0.5000*	0.5333*	**0.7333**
F_3	0.4000	0.9667*	0.7333*	**1.0000**
F_4	0.7667	0.9667*	0.9667*	**1.0000**
F_5	0.5333	**1.0000**	0.9667*	**1.0000**
F_6	0.9667	0.9667	0.9333	**1.0000**
+/–/ =		5/0/1	5/1/0	6/0/0

由表 2 可知: 两种改进策略的 RR 指标大部分优于传统 NCDE 算法, 其融合型 NCD_DE 算法指标值均显著优于 NCDE 算法, 表 3 中 SR 指标亦可得到相似占优结果; 特别是 NCD_DE 算法在 4/6 个方程组上均能够成功找到 NESs 的所有根, 验证了两种改进策略对 NESs 多根联解性能的有效性, 以及所提 NCD_DE 算法对 NESs 优异的多根锚定能力.

3) 算法性能的对比分析实验.

为进一步验证 NCD_DE 算法的优异寻根性能, 以 3.2 节的 6 种对比算法及相应参数设置进行 30 次独立实验, 所得 RR 和 SR 指标的对比结果见表 4 和表 5.

表4　NCD_DE 算法与对比算法的 RR 指标对比

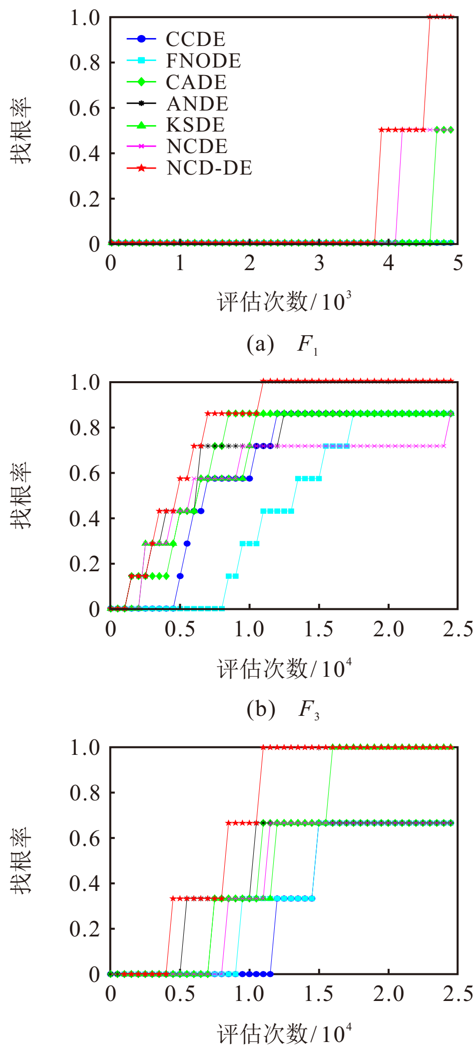
函数	NCD_DE	NCDE	KSDE	ANDE	FNODE	CADE	self_CCDE
F_1	**0.7667**	0.4500	0.0333	0	0	0.2667	0.0500
F_2	**0.9822**	0.8844	0.9622	0.5156	0.8244	0.9622	0.9511
F_3	**1.0000**	0.8762	0.9476	0.8714	0.9286	0.9238	0.9476
F_4	**1.0000**	0.9667	0.9952	0.4952	**1.0000**	0.9857	0.9143
F_5	**1.0000**	0.8444	**1.0000**	0.6778	0.8889	0.9333	0.8778
F_6	**1.0000**	0.9933	0.9867	0.2667	0.4933	0.9933	0.7333
Avg.	**0.9582**	0.8358	0.8208	0.4711	0.6892	0.8442	0.7457
+/–/ =		6/0/0	5/0/1	6/0/0	5/0/1	6/0/0	6/0/0

由表 4 可知, NCD_DE 算法相较于其他 6 种对比算法, 对 6 组 NESs 问题均获得更高的 RR 指标值且其平均 RR 指标值高达 0.9582, 有效佐证了所提算法相对更强的多根搜寻和锚定性能; 表 5 中 NCD_DE 的 SR 指标仍保持多根同步捕获能力的高竞争性且成功率高达 0.8889, 综合验证了 NCD_DE 算法相对更为优异的 NESs 多根联解性能.

表5　NCD_DE 算法与对比算法的 SR 指标对比

函数	NCD_DE	NCDE	KSDE	ANDE	FNODE	CADE	self_CCDE
F_1	**0.6000**	0.2000	0	0	0	0.0667	0
F_2	**0.7333**	0.2000	0.5667	0	0.0333	0.5000	0.3667
F_3	**1.0000**	0.4000	0.6333	0.1333	0.5000	0.5333	0.6667
F_4	**1.0000**	0.7667	0.9667	0	**1.0000**	0.9000	0.5333
F_5	**1.0000**	0.5333	**1.0000**	0.2000	0.6667	0.8000	0.6333
F_6	**1.0000**	0.9667	0.9333	0	0	0.9667	0.1000
Avg.	**0.8889**	0.5111	0.6833	0.0556	0.3667	0.6278	0.3833
+/–/ =		6/0/0	5/0/1	6/0/0	5/0/1	6/0/0	6/0/0

为直观展示 NCD_DE 算法优异的 NESs 多根联解性能, 绘制不同算法的找根率对比曲线见图 3, 各分图图例同图 3(a).



(a)　F_1

(b)　F_3

(c)　F_5

图3　NCD_DE 算法与 6 种对比算法的(单次)找根率对比曲线

由图 3 可知: 随着种群的进化迭代, 各算法的找根率均呈现出较明显的阶段性跃升态势, 且以 NDC_DE 算法的性能表现最优, 究其原因是进化个体随迭代进程逐渐聚集于多个根附近, 通过种群多样性保持

机制和跳出局部最优策略依概率搜寻并定位到 NESs 的更多根. 在迭代过程中, NDC_DE 算法能以相对更小的评估次数动态锚定更多的根, 即以相对更少的计算资源获得相对更高的找根率.

各算法 30 次独立实验的 Friedman 检验秩次排名结果见表 6. 由表 6 分析可知, 所提 NCD_DE 算法在 RR 和 SR 指标上的秩次排名均显著优于其他对比算法, 进一步验证了所提 NCD_DE 算法在 NESs 多根联解性能上的有效性和优越性.

表6 各算法的 Friedman 检验秩次排名

算法	秩次排名(RR)	秩次排名(SR)
NCD_DE	**1.0000**	**1.0000**
NCDE	3.6667	4.0000
KSDE	2.5000	2.5000
ANDE	6.0000	5.8333
FNODE	3.8333	4.0000
CADE	2.6667	2.8333
self_CCDE	3.6667	3.8333

4 结 论

本文针对 NESs 问题智能求解中存在的多根析出不完整、易陷入局部最优等问题, 提出一种基于邻域交叉的双变异差分进化算法 (NCD_DE), 以改善算法的种群多样性和计算资源利用效率. 实验结果表明所提 NCD_DE 算法对 NESs 问题的智能求解表现出相对更优的找根率和成功率, 验证了其优异多根联解性能. 后续将重点研究群组个体自适应排斥、聚合技术以进一步增强 NESs 问题的多根搜寻能力.

参考文献 (References)

[1] Liao Z W, Zhu F Y, Mi X Y, et al. A neighborhood information-based adaptive differential evolution for solving complex nonlinear equation system model[J]. Expert Systems with Applications, 2023, 216: 119455.

[2] Li S J, Gong W Y, Lim R, et al. Evolutionary multitasking for solving nonlinear equation systems[J]. Information Sciences, 2024, 660: 120139.

[3] Sharma J R, Guha R K, Sharma R. An efficient fourth order weighted-Newton method for systems of nonlinear equations[J]. Numerical Algorithms, 2013, 62(2): 307-323.

[4] Mehta D. Finding all the stationary points of a potential-energy landscape via numerical polynomial-homotopy-continuation method[J]. Physical Review E, Statistical, Nonlinear, and Soft Matter Physics, 2011, 84(2): 025702.

[5] 赵世杰, 高雷阜, 于冬梅, 等. 融合能量周期性递减与牛顿局部增强的改进 HHO 算法[J]. 控制与决策, 2021, 36(3): 629-636.
(Zhao S J, Gao L F, Yu D M, et al. Improved Harris Hawks optimization coupling energy cycle decline mechanism and Newton local enhancement strategy[J]. Control and Decision, 2021, 36(3): 629-636.)

[6] Gong W Y, Wang Y, Cai Z H, et al. Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(4): 1499-1513.

[7] Liao Z W, Gong W Y, Yan X S, et al. Solving nonlinear equations system with dynamic repulsion-based evolutionary algorithms[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(4): 1590-1601.

[8] He W, Gong W Y, Wang L, et al. Fuzzy neighborhood-based differential evolution with orientation for nonlinear equation systems[J]. Knowledge-Based Systems, 2019, 182: 104796.

[9] Liao Z W, Gong W Y, Wang L, et al. A decomposition-based differential evolution with reinitialization for nonlinear equations systems[J]. Knowledge-Based Systems, 2020, 191: 105312.

[10] Qu B Y, Suganthan P N, Liang J J. Differential evolution with neighborhood mutation for multimodal optimization[J]. IEEE Transactions on Evolutionary Computation, 2012, 16(5): 601-614.

[11] Wu J Y, Gong W Y, Wang L. A clustering-based differential evolution with different crowding factors for nonlinear equations system[J]. Applied Soft Computing, 2021, 98: 106733.

[12] Li S J, Gong W Y, Gu Q, et al. Adaptive dual niching-based differential evolution with resource reallocation for nonlinear equation systems[J]. Neural Computing and Applications, 2023, 35(16): 11917-11936.

[13] 王开, 龚文引. 求解非线性方程组系统的改进差分进化算法[J]. 控制与决策, 2020, 35(9): 2121-2128.
(Wang K, Gong W Y. Solving nonlinear equations system with an improved differential evolution[J]. Control and Decision, 2020, 35(9): 2121-2128.)

[14] Liao Z W, Gong W Y, Wang L. Memetic niching-based evolutionary algorithms for solving nonlinear equation system[J]. Expert Systems with Applications, 2020, 149: 113261.

[15] Wang Z J, Zhan Z H, Lin Y, et al. Automatic niching differential evolution with contour prediction approach for multimodal optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2020, 24(1): 114-128.

[16] Gao W F, Yen G G, Liu S Y. A cluster-based differential evolution with self-adaptive strategy for multimodal optimization[J]. IEEE Transactions on Cybernetics, 2014, 44(8): 1314-1327.

作者简介

赵世杰 (1987−), 男, 副教授, 博士, 博士生导师, 主要研究方向为智能优化与计算、机器学习与数据解析、水下导航与智能定位, E-mail: zhaoshijie@lntu.edu.cn;

赵秋丽 (1998−), 女, 硕士生, 主要研究方向为进化计算、非线性方程组智能求解, E-mail: zhao3346231586@163.com;

陈淼 (1999−), 女, 硕士生, 主要研究方向为进化计算、约束多目标优化, E-mail: 18342851718@163.com;

崔倩倩 (1998−), 女, 硕士生, 主要研究方向为进化计算、多目标特征选择, E-mail: 15939408618@163.com.