

自适应双坐标系的差分进化算法 求解混合变量优化问题

周新宇¹⁾ 黄君洪^{1),2)} 彭 虎³⁾ 王 晖⁴⁾ 王 峰⁵⁾

¹⁾(江西师范大学计算机信息工程学院 南昌 330022)

²⁾(吉林大学计算机科学与技术学院 长春 130012)

³⁾(九江学院计算机与大数据科学学院 江西 九江 332005)

⁴⁾(南昌工程学院信息工程学院 南昌 330022)

⁵⁾(武汉大学计算机学院 武汉 430072)

摘 要 如何设计求解混合变量优化问题(Mixed-variable Optimization Problems, MVOPs)的相关算法,是计算机算法设计与分析领域中的一个重要研究方向. 该问题的求解难点在于需同时优化连续型和离散型决策变量,目前进化算法(Evolutionary Algorithms, EAs)是求解该问题的一种有效手段. 然而,现有的相关 EAs 忽略了问题变量之间的相关性,导致算法性能还存在一定不足. 为此,本文从变量相关性角度出发,提出了一种自适应双坐标系的差分进化算法来求解 MVOPs. 首先,利用种群的协方差矩阵信息来构建特征坐标系,实现在特征坐标系下执行算法的相关操作,以松弛连续变量与离散变量之间的相关性;其次,为避免种群多样性丢失,仍保留了原坐标系,并设计了一种自适应策略来应用特征坐标系和原坐标系,以发挥双坐标系的优势;最后,为提高离散变量的优化效果,专门设计了一种基于离散变量相关性的局部搜索策略,以增强算法的整体性能. 为验证本文方法性能,在一套包含 28 个测试函数的通用测试集上进行了大量实验,与 5 种求解 MVOPs 的知名 EAs 进行了对比,结果表明本文方法有更好性能. 此外,在一个实际 MVOP 上,即焊接梁设计问题,本文方法能取得目前已知最好解. 本文方法为计算机算法设计和分析领域的相关工作提供了新的思路.

关键词 混合变量;差分进化;变量相关性;特征坐标系;双坐标系

中图法分类号 TP391

DOI 号 10.11897/SP.J.1016.2024.02116

Differential Evolution Algorithm Based on Adaptive Bi-Coordinate Systems for Mixed-Variable Optimization Problem

ZHOU Xin-Yu¹⁾ HUANG Jun-Hong^{1),2)} PENG Hu³⁾ WANG Hui⁴⁾ WANG Feng⁵⁾

¹⁾(School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022)

²⁾(College of Computer Science and Technology, Jilin University, Changchun 130012)

³⁾(School of Computer and Big Data Science, Jiujiang University, Jiujiang, Jiangxi 332005)

⁴⁾(School of Information Engineering, Nanchang Institute of Technology, Nanchang 330022)

⁵⁾(School of Computer Science, Wuhan University, Wuhan 430072)

Abstract How to design algorithms for solving Mixed-variable Optimization Problems (MVOPs) is an important research direction in the field of computer algorithm design and analysis. The difficulty of this problem lies in the need to optimize both continuous and discrete decision variables, and Evolutionary Algorithms (EAs) are an effective means to solve this problem.

收稿日期:2023-08-25;在线发布日期:2024-05-17. 本课题得到国家自然科学基金(62366022,61966019,62166027,62173258)、江西省自然科学基金(20232BAB202048)资助. 周新宇,博士,副教授,中国计算机学会(CCF)会员,主要研究领域为智能计算、机器学习. E-mail:xyzhou@jxnu.edu.cn. 黄君洪(通信作者),博士研究生,主要研究领域为智能优化、边缘计算. E-mail:drhugh9527@163.com. 彭 虎,博士,副教授,主要研究领域为大数据分析、智能计算及应用. 王 晖,博士,教授,主要研究领域为进化计算、群智能、优化调度. 王 峰,博士,教授,主要研究领域为大数据挖掘管理与分析、机器学习与智能交互、智能计算.

However, there exists an issue that the correlation between problem variables is ignored, which deteriorates the algorithm performance to some extent. Therefore, in this work, a differential evolution algorithm based on adaptive bi-coordinate systems is proposed to solve MVOPs from the perspective of variable correlation. Firstly, the covariance matrix information of the population is used to establish the eigen coordinate system, so that the relevant operations of the algorithm can be performed in the eigen coordinate system to relax the correlation between continuous and discrete variables; secondly, to avoid the loss of population diversity, the original coordinate system is retained, while an adaptive strategy is designed to switch the eigen coordinate system and the original coordinate system for taking full advantages of the two coordinate systems; finally, to better optimize the discrete variables, a local search strategy based on the correlation between discrete variables is specially designed, which is helpful for the overall performance of the algorithm. To verify our approach, extensive experiments are conducted on a widely used test suite containing 28 test functions, and the comparative results with five well-known EAs of solving MVOPs confirm that the proposed approach owns better performance. Besides, on a real-world MVOP, i. e., the welded beam design problem, and our approach is able to obtain the ever best-known solution to the problem. Our approach provides a new idea for the related work in the field of computer algorithm design and analysis.

Keywords mixed-variable optimization; differential evolution; variable correlation; eigen coordinate system; bi-coordinate systems

1 引言

在计算机算法设计与分析领域中,如何设计求解混合变量优化问题的相关算法是一个重要研究方向.事实上,混合变量优化问题(Mixed-variable Optimization Problems, MVOPs)在实际的生产和生活中十分常见,例如:压力容器设计问题^[1]、报贩问题^[2]以及焊接梁设计问题等^[3].这类问题的主要特点是目标函数不仅有非凸、多峰、不可微等属性,其决策变量还同时包含了连续型和离散型两类变量.以最小化问题为例, MVOPs 的数学表达式可定义如下:

$$\begin{aligned} & \text{Min } f(\mathbf{X}) \\ \text{s. t. } & \begin{cases} g_j(\mathbf{X}) \leq 0, j = 1, 2, \dots, k \\ h_j(\mathbf{X}) = 0, j = k + 1, k + 2, \dots, q \\ x_i^l \leq x_i \leq x_i^u, i = 1, 2, \dots, D \end{cases} \end{aligned} \quad (1)$$

其中, $g_j(\mathbf{X}) \leq 0$ 、 $h_j(\mathbf{X}) = 0$ 分别表示问题的不等式约束和等式约束. $\mathbf{X} = (\mathbf{X}^m, \mathbf{X}^n) = (x_1, \dots, x_m, x_{m+1}, \dots, x_{m+n})$, 其中 \mathbf{X}^m 、 \mathbf{X}^n 分别为 m 维连续型变量和 n 维离散型变量,且 $m+n=D$; x_i^u 和 x_i^l 分别是第 i 维决策变量的上下界.

相对于连续优化问题(Continuous optimization

problems, COPs)或离散优化问题(Discrete Optimization Problems, DOPs), MVOPs 的搜索空间复杂度更高,求解难度也更大.为求解 MVOPs,研究人员做了一些尝试,设计了分支定界法^[4]、割平面法^[5]以及分解算法^[6]等.然而,随着问题规模不断增大,这些方法因需消耗大量的算力和存储空间,导致算法的求解效率也快速下降.因此,近年来,有相关研究人员开始采用进化算法(Evolutionary Algorithms, EAs)来求解 MVOPs,并取得了一定成效.在计算机算法设计与分析领域中,EAs 是一种高效的随机搜索算法,它不依赖于问题的具体数学特性,可方便地用于求解各类优化问题,特别是对于 NP 难的问题,有非常良好的求解效果.

然而,现有的相关 EAs 几乎都是仅针对 COPs 或 DOPs 设计的,例如:基本的差分进化算法(Differential Evolution Algorithm, DE)^[7]、粒子群优化算法(Particle Swarm Optimization Algorithm, PSO)^[8]、人工蜂群算法(Artificial Bee Colony Algorithm, ABC)^[9]等通常用于求解 COPs,而基本的蚁群算法(Ant Colony Algorithm, ACO)^[10]一般适用于求解 DOPs,鲜有范例能直接用于求解 MVOPs.由于连续变量和离散变量在空间分布特征上存在巨大差异, EAs 往往无法直接用于求解

MVOPs. 因此,采用 EAs 求解 MVOPs 就必须解决两个难点:其一,如何设计出高效的方法来分别处理连续变量和离散变量;其二,如何对混合变量的搜索空间进行高效协同搜索,使得算法能快速定位到全局最优解.

早期,研究人员做了一些初步尝试,例如:Debd 等人^[11]提出了一种遗传自适应的遗传算法、Ocenasek 等人^[12]提出了一种混合贝叶斯优化算法以及 Guo 等人^[13]提出了一种混合群智能方法.但遗憾的是,这些方法仅是简单地将 MVOPs 中的连续变量和离散变量分开进行处理,仍然无法很好地协同处理混合变量,算法效果不尽人意.最近,研究人员在这些早期工作的基础上做了进一步研究,例如:Wang 等人^[2]提出了一种改进的分布估计算法用于求解报贩问题,该算法的特点是采用了两种直方图模型;基于混合变量编码方式,Wang 等人^[14]提出了一种改进的 PSO 算法(PSO_{mv}),设计了两种重组策略来分别处理连续变量和离散变量;Peng 等人^[15]提出了多策略协同进化的 DE 算法($MCDE_{mv}$),他们基于松弛法思路设计了一种混合变量协同优化方法来处理连续变量和离散变量.毫无疑问,这些近期的相关工作有效提高了 EAs 求解 MVOPs 的性能.但需指出的是,这些相关工作也还存在一定不足,它们忽略了连续变量与离散变量之间的相关性,而这种相关性是导致 MVOPs 较难求解的原因之一,会影响算法的搜索方向,降低算法的有效性,特别是对于变量相关性高的 MVOPs 更是如此.

基于上述考虑,本文提出了自适应双坐标系的差分进化算法(Differential Evolution algorithm based on Adaptive Bi-coordinate Systems, $ABSDE_{mv}$)来求解 MVOPs.在 $ABSDE_{mv}$ 中,为松弛连续变量与离散变量之间的相关性,基于种群的协方差矩阵信息,把原坐标系旋转至特征坐标系,在特征坐标下使用搜索策略生成后代个体,有利于引导种群向更优的搜索区域进化.同时,为避免种群的多样性丧失, $ABSDE_{mv}$ 仍保留了原坐标系,但设计了一种自适应策略在进化过程中切换原坐标系和特征坐标系,便于发挥出两种坐标系的各自优势.同时,为提高对离散变量的优化效果,在 $ABSDE_{mv}$ 中专门设计了一种基于离散变量相关性的局部搜索策略,利用离散变量之间的相关性,对多个相关性强的离散变量同时进行优化,以增强算法处理离散变量的性能.值得一提的是,受集成变异策略思路的启发,在 $ABSDE_{mv}$ 中还使用了集成变异策略框架,以提高算

法的整体性能.为验证 $ABSDE_{mv}$,在一套包含 28 个测试函数的通用测试集上进行了大量实验,与 5 种求解 MVOPs 的知名 EAs 进行了对比,实验结果表明 $ABSDE_{mv}$ 有更好性能.此外, $ABSDE_{mv}$ 还被应用于求解一个实际 MVOP,即焊接梁设计问题, $ABSDE_{mv}$ 能得到该问题的目前已知最好解,并且算法性能非常稳定.

本文方法为计算机算法设计与分析领域的相关工作,特别是随机搜索算法的相关工作,提供了新的研究思路,其主要贡献及创新之处可总结如下:

(1)首次从变量相关性的角度设计了求解 MVOPs 的 EAs,通过引入特征坐标系来松弛连续变量、离散变量以及连续变量和离散变量之间的相关性,提升了算法求解性能,为随机搜索算法的相关工作提供了参考.

(2)提出了基于离散变量相关性的局部搜索策略,根据离散变量之间的相关性来确定要优化的决策变量,进一步提高了离散变量的优化性能,提高了算法的搜索效率和收敛速度.

(3)采用了集成变异策略框架对连续变量和离散变量进行协同优化,克服了经典 DE 仅采用单一变异策略和固定参数值的局限性,提高了算法的整体性能和鲁棒性.

本文的章节安排如下:第 2 节简要介绍 DE 算法;第 3 节分别对求解混合变量优化问题的 EAs 和 DE 的相关研究工作进行回顾;第 4 节介绍了本文提出的 $ABSDE_{mv}$,并详细叙述了算法的相关细节;第 5 节给出一系列相关的实验验证及结果分析;最后一节对本文工作进行了总结和展望.

2 差分进化算法

作为一种基于种群的启发式随机搜索算法,DE 是进化算法中效果最好的范例之一,最早由 Storn 和 Price 于 1995 年提出^[16],主要通过变异、交叉和选择三种进化操作来生成后代个体.DE 通过迭代进化一个大小为 NP 的种群以得到最优解,假设问题有 D 维决策变量,则第 g 代种群的第 i 个目标个体可表示为

$$\mathbf{X}_i^g = (x_{i,1}^g, x_{i,2}^g, \dots, x_{i,D}^g) \quad (2)$$

其中 $i \in \{1, 2, \dots, NP\}$, $j \in \{1, 2, \dots, D\}$.当 $g=0$ 时, \mathbf{X}_i^0 的第 j 维决策变量 $x_{i,j}^0$ 可按下式随机生成:

$$x_{i,j}^0 = l_j + \text{rand}(0, 1) \cdot (u_j - l_j) \quad (3)$$

其中, $\text{rand}(0, 1)$ 表示 $[0, 1]$ 内服从均匀分布的随

机数, l_j 和 u_j 分别是第 j 维决策变量的下界和上界.

在初始化种群之后, DE 采用变异、交叉和选择三种操作推动种群迭代进化, 直到算法达到预设的停机条件, 例如: 最大迭代次数 g_{max} 或最大适应度函数评估次数 $MaxFEs$. 另外, 变异操作和交叉操作的组合也常被称为试验个体生成策略. 下面简介 DE 的这三种操作:

(1) 变异操作. 变异操作是 DE 至关重要的一步, 也是 DE 区别于其他 EAs 范例的最大不同点. 以最简单且常用的变异策略“DE/rand/1”为例, 其公式如下:

$$\mathbf{V}_i^g = \mathbf{X}_{r_1}^g + F \cdot (\mathbf{X}_{r_2}^g - \mathbf{X}_{r_3}^g) \quad (4)$$

其中, \mathbf{V}_i^g 为第 g 代目标个体 \mathbf{X}_i^g 对应产生的变异个体; $\mathbf{X}_{r_1}^g, \mathbf{X}_{r_2}^g, \mathbf{X}_{r_3}^g$ 是从种群中随机选择的互不相同的个体, 且它们都不同于目标个体 \mathbf{X}_i^g ; 参数 F 为缩放因子, 是 $[0, 1]$ 内服从均匀分布的随机数, F 值越高, 说明搜索步长越大, 这有利于勘探; 相反, F 值较低意味着将围绕目标个体进行细粒度搜索, 这有利于开采. 此外, 如果 \mathbf{V}_i^g 的第 j 维决策变量超出边界 $[l_j, u_j]$, 则通过如下公式进行修复:

$$v_{i,j}^g = \begin{cases} \min(u_j, 2l_j - v_{i,j}^g), & v_{i,j}^g < l_j \\ \max(l_j, 2u_j - v_{i,j}^g), & v_{i,j}^g > u_j \end{cases} \quad (5)$$

(2) 交叉操作. 在生成变异个体后, DE 为每对 \mathbf{X}_i^g 和 \mathbf{V}_i^g 进行交叉操作, 以得到一个试验个体 $\mathbf{U}_i^g = (u_{i,1}^g, u_{i,2}^g, \dots, u_{i,D}^g)$. DE 中最常用的二项式交叉可表示如下:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & \text{if } \text{rand}(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j}^g, & \text{otherwise} \end{cases} \quad (6)$$

其中, j_{rand} 为 $[1, D]$ 内的随机整数, $\text{rand}(0,1)$ 为 $[0, 1]$ 内均匀分布的随机数. CR 为交叉概率, 对 DE 性能有重要影响, CR 值越大意味着后代个体可从变异个体中继承更多信息; 相反, CR 值越小, 后代个体与父代个体之间的差异越小. 值得注意的是, 条件 $j = j_{rand}$ 可确保试验个体 \mathbf{U}_i^g 与目标个体 \mathbf{X}_i^g 至少在一个维度上存在差异.

(3) 选择操作. DE 的选择操作是目标个体与其对应的试验个体之间的一对一竞争. 对于最小化问题, 如果试验个体的目标函数值小于或等于目标个体的目标函数值, 则试验个体可存活至下一代; 否则, 目标个体将进入下一次迭代. 这一过程正是“适者生存”的体现, 可形式化如下:

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

3 相关工作

3.1 求解混合变量优化问题的 EAs 的相关工作

EAs 求解混合变量优化问题主要有三种方法: 1) 离散化方法, 先将连续变量的搜索范围离散化, 再对离散变量进行优化; 2) 二分方法, 对连续变量和离散变量分别设计相应的处理策略, 以此产生包含混合变量的解; 3) 松弛方法, 先采用连续优化策略对连续变量和离散变量同时进行优化, 之后再采用转化方法, 如: 舍入法, 把连续变量转化为离散值来产生新解. 下面简介这三种方法的相关 EAs:

(1) 基于离散化方法的 EAs

该类算法先将连续变量处理为离散变量, 然后再用离散变量优化策略对问题求解. 例如, Ocenasek 等人^[12]提出了一种混合的贝叶斯优化算法, 构建了带混合决策节点的树使得算法既适用于 COPs, 也适用于 DOPs; 随后, 再将算法与分布估计算法结合用于求解 MVOPs. Shi 等人^[17]提出了一种基于数据驱动模型的自适应分布估计算法(AE-DA), 该算法混合了连续和离散概率分布模型来处理不同类型的变量. 为求解混合变量报贩问题, Wang 等人^[2]提出了一种基于直方图模型的分布估计算法(EDA_{mv}), 其中连续变量采用宽度自适应直方图模型, 离散变量则采用基于学习的直方图模型; 实验结果表明, EDA_{mv} 的性能不仅优于其他相关算法, 而且在某些方面甚至超过了商业软件 Lingo.

(2) 基于二分方法的 EAs

该类算法同时采用连续变量优化策略和离散变量优化策略来生成包含混合变量的解. 例如, Lin 等人^[18]提出了一种混合 DE 框架(DE_{mv}), 该框架混合了基本 DE 和基于集合的 DE 用于分别优化连续变量和离散变量, 与其他代表性算法的对比结果表明 DE_{mv} 是高效的. Wang 等人^[14]提出了一种改进的 PSO 算法(PSO_{mv}), 设计了两种重组策略来分别处理连续变量和离散变量; 此外, 在 PSO_{mv} 中采用了自适应参数调整策略和约束处理方法来提高算法整体性能. 为求解混合整数规划问题, Liu 等人^[19]提出了一种基于双目标优化的两阶段方法在第一阶段, 将该问题转化为约束双目标优化问题, 使种群逼近满足整数约束的 Pareto 最优解, 第二阶段专注于提高算法的收敛精度, 从而方便得到最优解.

(3) 基于松弛方法的 EAs

该类算法先将离散变量松弛转化为连续变量,

然后使用连续变量优化策略对问题求解. 例如, Mashinchi 等人^[20]提出了一种基于禁忌搜索和 Nelder-Mead 搜索策略的混合优化算法; 禁忌搜索可覆盖更广的解空间, 使搜索远离当前解, 从而减少算法陷入局部最优的可能. Liao 等人^[21]提出了一种求解 MVOPs 的 ACO 算法 (ACO_{mv}); 在 ACO_{mv} 中, 优化问题的决策变量可显式地声明为连续变量、序数变量或类别变量; 相应地, 设计了三种后代个体生成机制, 即: 连续优化机制、连续松弛机制、以及分类优化机制. 最近, Peng 等人^[15]提出了多策略协同进化的 DE 算法 (MCDE_{mv}), 他们基于松弛法思路设计了一种混合变量协同进化方法来处理连续变量和离散变量, 该方法的主要特点是结合了不同特征的变异策略和交叉操作, 且对离散变量专门设计了基于统计的局部搜索策略, 能取得较为满意的优化效果.

3.2 DE 的相关工作

目前, DE 主要用于求解 COPs, 但还存在一些不足, 例如: 局部搜索能力较弱、进化后期收敛速度慢等. 为解决这些问题, 研究人员相继提出了多种不同的改进算法, 这些算法可粗略地划分为以下两类:

(1) 改进或提出新的变异策略

变异策略是 DE 区别于其他 EAs 范例的主要特征, 其对于算法性能有重要影响, 因此如何改进或提出新的变异策略是 DE 相关工作中的一个重要类别. Zhang 和 Sanderson^[22]提出了一种自适应 DE, 其亮点是设计了一种新的变异策略“DE/current-to- p best/1”, 其中 p best 表示从当前种群的 $p\%$ 精英个体中随机选择的一个精英个体; 同时, 该算法还使用了外部存档来自适应更新控制参数 F 和 CR . 受 PSO 算法的启发, Das 等人^[23]提出了一种基于邻域搜索的“DE/current-to-best/1”变异策略, 使个体只能与同邻域中的其他个体进行变异操作, 在一定程度上提高了 DE 算法的开采能力. 为减少 DE 对控制参数的依赖, Wang 等人^[24]提出一种基于高斯变异策略的 DE, 其变异个体由预先构建的高斯分布模型来生成, 该分布的均值和标准差由当前个体和全局最优个体计算而来, 此变异策略几乎不需额外的控制参数. Deng 等人^[25]提出了一种基于动态组合变异策略和两级参数自适应策略的 DE; 在该算法中, 新提出的变异策略包含了一个动态基向量, 由当前最优个体和一个精英个体组成, 有利于增强算法的开采能力.

(2) 集成多种变异策略

不同的变异策略往往有不同的特点和性能, 通

常单一策略无法用于求解多种不同类型的问题或复杂问题. 因此, 如何根据问题特点来选择最恰当的变异策略是 DE 的一个研究热点. 该类工作的一个主流思路是采用多种变异策略用于构建策略候选池, 再根据设计好的概率模型从该候选池中自适应地选择策略, 代表性的工作有: Qin 等人^[26]设计了一种变异策略自适应的 DE, 采用四种常见的变异策略来构建候选池, 并根据以往搜索过程中的经验来动态调整每种策略被选中的概率; Mallipeddi 等人^[27]提出了一种集成变异策略的 DE, 其变异策略和参数均从已构建好的策略候选池和参数候选池中选择, 并且根据每组变异策略和参数值的历史表现来设置其选择概率; Gong 等人^[28]提出了一种策略自适应机制, 先为每种策略采用不同的参数自适应方法, 再为每个个体设置了对应的策略选择参数, 最后根据构建的正态分布概率模型来自适应调整参数值; Yi 等人^[29]设计了基于集成种群的自适应 DE, 该算法利用了两组变异策略和交叉操作来生成后代, 以更好地平衡算法的开采和勘探能力.

4 ABSDE_{mv} 算法

4.1 算法动机

对最优化问题而言, 决策变量之间普遍存在相关性, 不仅对于 COPs 或 DOPs 是这样, 对 MVOPs 也是如此. 通常, 在求解最优化问题的情景下, 决策变量间的相关性可视作是把问题的最优解变换至坐标系的非坐标轴位置, 从而无法通过逐维寻优的方式来得到问题的最优解, 即问题是不可分的 (Non-separable). 与此相反, 若问题的最优解恰好位于坐标轴上, 那么可按逐维寻优的方式来较容易地获得最优解, 即该问题是可分的 (Separable). 一般而言, 不可分问题的求解难度要远大于可分问题, 因其决策变量间的相关性会导致求解过程容易产生“连锁反应”, 仅改变解的某一维度无法提高解的质量, 而须同时改变有相关性的某些维度时才能找到更好解, 有“牵一发而动全身”的效果. 当然, 这种相关性给算法的求解也带来了挑战.

事实上, 研究人员在求解 COPs 时就注意到了决策变量间存在相关性的特点, 也相应地提出了一些解决办法. 例如, Guo 等人^[30]指出 DE 算法的交叉操作是旋转可变的, 易受变量间相关性的影响, 为此他们提出了基于特征向量的交叉操作, 把在原坐标系中执行的交叉操作放至特征坐标系中进行, 以

缓解变量间相关性的影响. 类似地, Wang 等人^[31]在累积种群分布信息的基础上, 通过计算出种群的协方差矩阵来建立特征坐标系, 以松弛变量之间的相关性. 这些解决办法对于如何提高算法处理变量间存在相关性的问题起到了积极促进作用.

受上述研究工作的启发, 我们注意到 MVOPs 的变量间同样存在相关性, 这种相关性存在于连续变量之间、离散变量之间以及连续变量和离散变量之间. 然而, 遗憾的是, 目前还未有相关工作从变量间相关性的角度来设计求解 MVOPs 的 EAs. 为此, 本文引入特征坐标来松弛这种相关性, 提高算法求解 MVOPs 的性能. 同时, 为保留原坐标系的优点, 设计了一种自适应双坐标系策略, 在两种坐标系之间自适应切换, 以发挥它们的优势. 为进一步提高离散变量的优化效果, 专门设计了一种基于离散变量相关性的局部搜索策略, 根据离散变量之间的相关性来确定要优化的决策变量.

4.2 解的编码方式

对于 COPs, 决策变量无需做特殊处理, 直接使用真实值即可. 对于 DOPs, 决策变量存在多种类型, 例如: 二进制数、整数, 或从有限候选集中选择的元素. 离散变量进一步可分为序数变量和类别变量. 序数变量是指具有显式顺序的变量, 例如: 表示学生年龄的变量就是序数变量, 因其是在一组具有明确顺序的整数上定义的; 相反, 类别变量就是指无显式顺序的变量, 例如: 在二进制背包问题中, 每个变量都是一个类别变量, 用 0 或 1 来表示是否将相关的物品放进到背包中.

对于 MVOPs, EAs 处理混合变量有离散化方法、二分方法以及松弛方法. 对于离散化方法, 使用二进制字符串对连续变量进行编码, 但这会导致二进制字符串长度限制算法的精确度. 与离散化方法不同的是, 二分方法通常会引入新的策略来处理原算法不能直接处理的决策变量, 并且还一般会嵌入一个局部搜索过程, 以提高算法的搜索效率. 虽然二分方法可根据变量类型的特点分别处理连续变量和离散变量, 但它忽略了变量之间的相关性, 且算法需设计两套优化策略, 策略之间缺失一致性和兼容性. 因此, 在连续变量和离散变量的协同优化过程中, 二分方法可能会带来一定的副作用. 至于松弛方法, 它使用与连续变量相同的进化策略, 以连续值的形式优化离散变量. 这种方法主要有两个优点: 一方面, 它不需要为离散变量设计特定的搜索策略; 另一方面, 它能够保留原算法的特点, 不会破坏不同变量之

间的相关性.

基于上述考虑, 我们采用松弛方法对解进行编码. 例如, 对于第 i 个个体 \mathbf{X}_i , 其编码方式如图 1 所示. \mathbf{X}_i 中前 M 维变量采用连续值进行编码, 剩余 N 维变量则采用离散值编码. 因此, 在算法的优化过程中, 不需要构造不同类型的模型. M 、 N 的值取决于决策变量中的连续变量和离散变量的数量, 此方式也适用于解决具体的工程设计问题. 因此, 在 MVOPs 中, 一个解可由包含连续变量和离散变量的混合编码方式表示. 在优化过程中, 先将离散变量 $\{x_{i,M+1}, x_{i,M+2}, \dots, x_{i,M+N}\}$ 作为连续变量 $\{\tilde{x}_{i,M+1}, \tilde{x}_{i,M+2}, \dots, \tilde{x}_{i,M+N}\}$ 进行优化, 再对连续值进行“向下取整”得到离散变量的值, 由此产生新解. 相较于其他取整方式, 例如: 四舍五入、向上取整等, 向下取整有两方面优点: 1) 简单方便; 2) 可减小信息损失, 若连续值表示某种数量, 向下取整可确保编码后的值不超过原始值, 从而在一定程度上保留了原始信息. 因此, 本文采取向下取整的方式对连续值离散化.

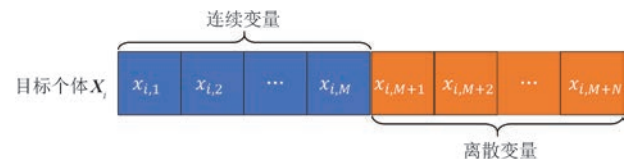


图 1 目标个体 \mathbf{X}_i 的混合编码方式

4.3 自适应双坐标系策略

因 DE 的二项式交叉或指数交叉是旋转可变的, 无论用何种交叉方式, DE 性能对坐标系的旋转都是敏感的, 特别是当目标函数具有强变量相关性时更是如此. 为解决此问题, 研究人员提出了一些算术重组方法, 例如: Price 提出了两种改进 DE (DE/rand/1/ Either-Or^[32] 和 drift-free DE^[33]), 即采用旋转不变的算术重组操作代替原来的二项式交叉操作. 然而, 算术重组方法可能会丢失种群的多样性, 且存在过早收敛的缺陷. 为解决这一问题, Guo 等人^[30]引入特征坐标系来代替原坐标系, 采用种群协方差矩阵的特征向量信息来旋转坐标系, 同时为避免种群多样性丢失, 以概率选择的方式来确定在哪个坐标系下生成试验个体, 报道的实验结果验证了该方法处理 COPs 是有效的, 能取得较好的优化效果.

对于既存在连续变量又存在离散变量的 MVOPs, 虽然变量类型不同, 对应的特点也不同, 但在优化过程中, 变量之间的相关性同样不可忽视, 比如: 连续变量之间、离散变量之间以及连续变量与离散变量

之间的相关性. 若能在一定程度上松弛变量之间的相关性, 则可提高算法求解 MVOPs 的性能. 基于这种考虑, 我们引入特征坐标系来实现 DE 的变异操作, 以缓解变量之间相关性带来的求解困难, 提高 DE 求解 MVOPs 的性能. 在特征坐标系下实现变异操作主要包括三个关键步骤: (1) 计算得到种群的协方差信息; (2) 对协方差矩阵进行特征分解以得到特征坐标系, 再对种群中的个体进行坐标系切换; (3) 在特征坐标系下进行变异、交叉操作, 得到试验个体. 下面对这三个步骤做进一步说明.

第一步: 计算得到种群协方差信息, 以计算第 g 代种群中第 i 维变量与第 j 维变量的协方差为例:

$$\text{cov}(i, j) = \frac{\sum_{k=1}^{NP} (x_{k,i}^g - \bar{x}_i^g)(x_{k,j}^g - \bar{x}_j^g)}{NP - 1} \quad (8)$$

其中 $\bar{x}_i^g = \frac{\sum_{k=1}^{NP} x_{k,i}^g}{NP}$ 为第 i 维变量的均值, \bar{x}_j^g 是第 j 维变量的均值. 协方差矩阵 \mathbf{C}^g 可由协方差定义为

$$\mathbf{C}^g = (c_{i,j}, c_{i,j} = \text{cov}(i, j)) \quad (9)$$

值得注意的是, 在本文算法 ABSDE_{mv} 中计算协方差矩阵的样本为种群中适应度值最好的一半个体, 其目的是减少种群随机性对协方差矩阵计算的干扰.

第二步: 将协方差矩阵进行特征分解以得到特征向量

$$\mathbf{C}^g = \mathbf{Q}^g \mathbf{\Lambda}^g (\mathbf{Q}^g)^{-1} \quad (10)$$

其中 \mathbf{Q}^g 是一个 $D \times D$ 的方阵, 第 i 列是 \mathbf{Q}^g 的特征向量 \mathbf{q}_i^g , $\mathbf{\Lambda}^g$ 是对角矩阵, 它的对角元素是相应的特

征值.

第三步: 在特征坐标系下进行变异、交叉操作, 从而得到试验个体. 以“DE/rand/1”为例, \mathbf{X}_i^g 、 $\mathbf{X}_{r_1}^g$ 、 $\mathbf{X}_{r_2}^g$ 和 $\mathbf{X}_{r_3}^g$ 在特征坐标系下的新位置可以表示为 $\widetilde{\mathbf{X}}_i^g = (\mathbf{X}_i^g \cdot \mathbf{Q}^g)$ 、 $\widetilde{\mathbf{X}}_{r_1}^g = (\mathbf{X}_{r_1}^g \cdot \mathbf{Q}^g)$ 、 $\widetilde{\mathbf{X}}_{r_2}^g = (\mathbf{X}_{r_2}^g \cdot \mathbf{Q}^g)$ 和 $\widetilde{\mathbf{X}}_{r_3}^g = (\mathbf{X}_{r_3}^g \cdot \mathbf{Q}^g)$, 则对应生成的变异个体为 $\widetilde{\mathbf{V}}_i^g$. 然后, 通过二项式交叉方式, 将目标个体与变异个体的某些决策变量进行交换, 产生一个试验个体, 可表示为

$$\widetilde{\mathbf{U}}_i^g = \text{Bin}(\widetilde{\mathbf{X}}_i^g, \widetilde{\mathbf{V}}_i^g) \quad (11)$$

其中, $\text{Bin}(a, b)$ 表示 a 和 b 之间的二项式交叉操作.

以二维问题为例, 图 2 展示了特征坐标系下实现变异、交叉操作的过程, 其中, x'_1, x'_2 为通过公式(10)特征分解得到的特征坐标系. 子图(a)表示根据种群中适应度值最好的一半个体计算协方差矩阵, 所得协方差矩阵展示了子图中虚线所示分布; 子图(b)表示对协方差矩阵进行特征分解得到特征坐标系, 同时选取目标个体; 子图(c)表示在特征坐标系下进行变异操作, 变异个体 $\widetilde{\mathbf{V}}_i^g$ 由紫色点表示; 子图(d)表示在特征坐标系下进行交叉操作, 试验个体 $\widetilde{\mathbf{U}}_i^g$ 由橙黄色点所示.

需指定的是, 我们在构造协方差矩阵时采用了种群中适应度值最好的一半个体作为样本, 这种方法能有效地松弛目标函数中变量与变量之间的相互作用, 但也可能会导致算法的搜索行为变得较为贪婪,

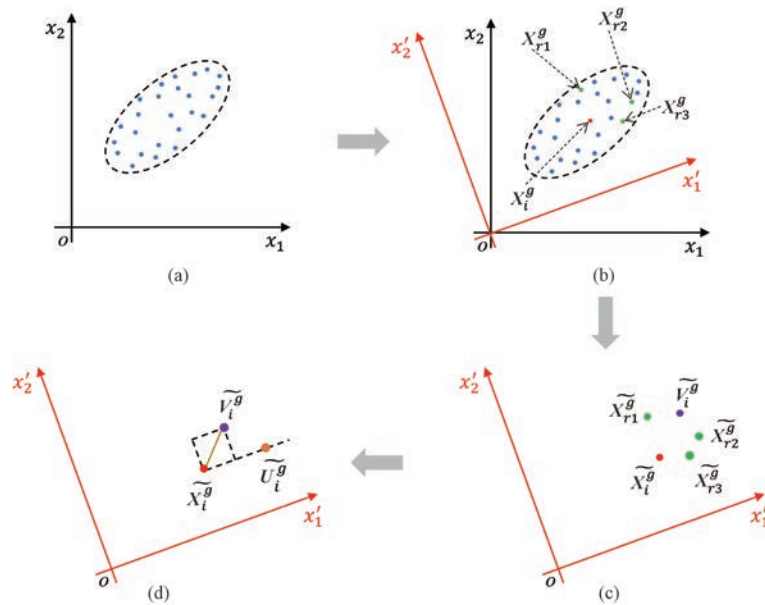


图 2 特征坐标系下的变异、交叉操作

从而容易丢失种群的多样性,导致算法性能下降.为此,我们保留了原坐标系,并自适应地在两种坐标系之间进行切换,以保持种群多样性.自适应切换策略介绍如下:

首先,计算算法在当前迭代中使用的坐标系的成功率.假设第 g 代中有 Num_S 个子代个体优于其父代个体,则该坐标系在第 g 代的使用成功率 $SR(g)$ 可定义为

$$SR(g) = \frac{Num_S}{NP} \quad (12)$$

然后,根据当前成功率 $SR(g)$ 与上一代成功率 $SR(g-1)$ 的关系,确定下一代 $g+1$ 使用何种坐标系,规则如下:

$$CS(g+1) = \begin{cases} CS(g), & \text{if } SR(g) \geq SR(g-1) \cdot \mu \\ \overline{CS(g)}, & \text{otherwise} \end{cases} \quad (13)$$

其中, $CS(g)$ 表示第 g 代使用的坐标系, $\overline{CS(g)}$ 表示与 $CS(g)$ 相反的另一坐标系;参数 $\mu = m + rand(0, 0.1)$ 为区间 $[m, m+0.1]$ 中的随机数,参数 m 为 $[0.1, 0.9]$ 之间的预设值.

最后,若 $CS(g+1) = \overline{CS(g)}$,即坐标系发生改变,则成功率 $SR(g)$ 初始化为 $SR(0)$,表达式如下:

$$SR(g) = \begin{cases} SR(0), & \text{if } CS(g+1) = \overline{CS(g)} \\ SR(g), & \text{otherwise} \end{cases} \quad (14)$$

从上述过程可看出,在迭代过程中,我们通过 $SR(g)$ 与 $SR(g-1) \cdot \mu$ 的关系来确定下一次迭代使用何种坐标系,参数 μ 可视为一个衰减因子.设置该衰减因子的目的是,随着算法的迭代次数增加,种群在进化后期产生更好后代的可能性会越来越小,从而导致坐标系的成功率也会越来越低,如果没有衰减因子的协助,坐标系会非常频繁地切换,在极端情况下,每一次迭代都可能会切换一次坐标系.因此,衰减因子在一定程度上能保持坐标系更平稳地切换.此外,若坐标系发生改变,则此时坐标系的成功率将不再作为下一次迭代的参考值,需按公式(14)重新初始化成功率.在本文中, m 设置为 0.7, $SR(0)$ 设置为 0.6,可使算法性能达到较好效果.关于参数 m 和 $SR(0)$ 的敏感性分析将在实验部分的 5.2 小节中单独给出.

4.4 基于离散变量相关性的局部搜索策略

值得注意的是,如果将离散变量简单地松弛为连续变量,则算法性能会随着离散变量的数量增加而降低.为提高算法在复杂离散空间的优化性能,针对离散变量,我们专门设计了一种基于离散变量相

关性的局部搜索策略 SBA_C ,其详细实现过程如下:

第一步:根据公式(9),以种群中最好的一半个体为样本,计算出离散变量的相关性矩阵,表达式如下:

$$C_d = \begin{bmatrix} cov(M+1, M+1) & \cdots & cov(M+1, N) \\ \vdots & \ddots & \vdots \\ cov(N, M+1) & \cdots & cov(N, N) \end{bmatrix} \quad (15)$$

其中, $cov(M+1, M+2)$ 表示为第 $M+1$ 维离散变量与第 $M+2$ 维离散变量的相关性.

第二步:统计当前种群中每个个体的所有离散变量的每一个可行值出现的频率.第 i 维中,第 j 个可行值的出现频率 $Prob_i(j)$ 可计算如下:

$$Prob_i(j) = \frac{count_{i,j}}{count_i} \quad (16)$$

其中, $count_{i,j}$ 为第 i 维中第 j 个可行值的出现次数, $count_i$ 是第 i 维中所有可行值的出现次数之和.

第三步:从种群中随机选择出 a 个个体,对每个被选中个体,先从其离散变量中随机选出一维,记作 $x_{i,j}$,再根据相关性矩阵 C_d 确定与 $x_{i,j}$ 相关性最高的 $b-1$ 维离散变量;然后,确定 $x_{i,j}$ 中出现频率最高的前 c 个可行值,为包括 $x_{i,j}$ 在内的 b 维离散变量每维独立地从这 c 个可行值中随机选择一个值用于替换,以此作为局部搜索.为更直观地展示这一步骤,以 $a=3, b=2, c=2$ 为例,在图 3 中给出了该步骤的操作过程.图 3(a)表示随机选择了 3 个个体 $\mathbf{X}_{r_1}, \mathbf{X}_{r_2}, \mathbf{X}_{r_3}$;图 3(b)表示确定的 2 维离散变量,其中绿色块表示随机选择的维度,紫色块表示与随机选择维度相关性最高的维度;图 3(c)表示从随机选择维度中出现频率最高的前 2 个可行值中随机选择一个作为对应维离散变量的值,例如:在第 $M+1$ 维中,应在可行值 $Num_{M+1,1}$ 和 $Num_{M+1,2}$ 之间随机选择一个用于替换原值.

为更直观地理解 SBA_C 的整个过程,算法 1 展示对应的伪代码. SBA_C 涉及 3 个控制参数: a, b, c . 参数 a 决定了有多少个体能进行局部搜索,也决定了所消耗的适应度值评估次数, a 值过大则消耗的评估次数越多,会影响到算法的整体性能.参数 b 控制了局部搜索的维度数,值越大则更新的维度越多;如果该值设置不恰当,可能会导致收敛速度过快,种群多样性丢失越快.参数 c 影响了所选个体的收敛速度,如果 $c=1$,则只将出现频率最高的可行值作为更新值,如果 c 值接近所有可行值的数量,则

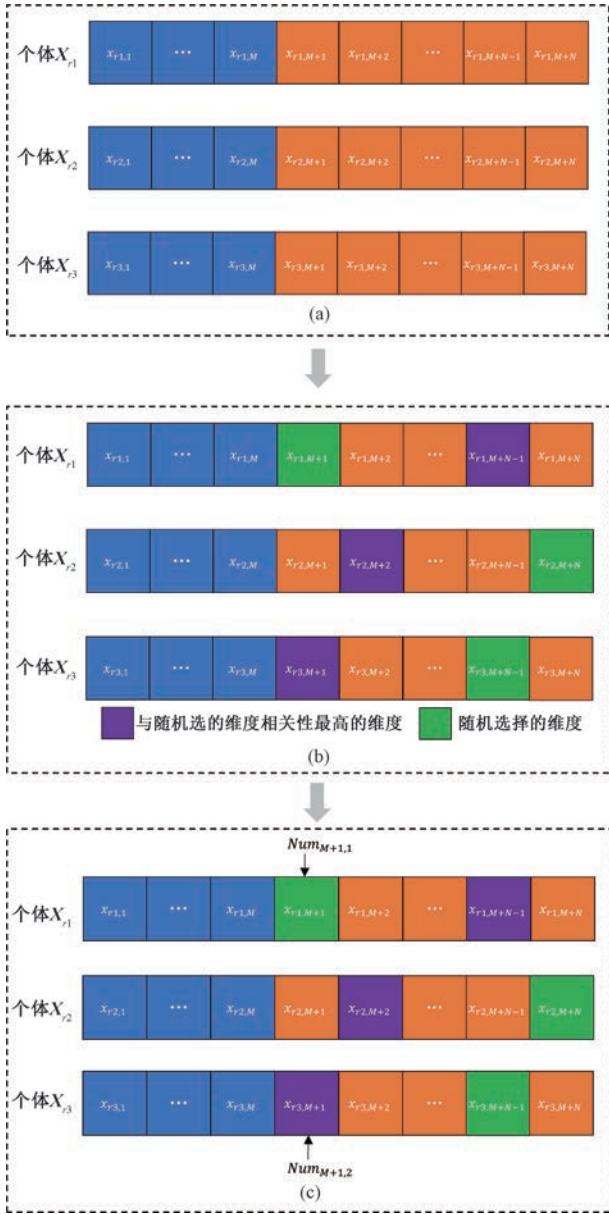


图3 基于离散变量相关性的局部搜索过程

可行值的选择趋近于随机。此处,我们将 a, b, c 分别设置为 $\{5, 3, 5\}$, 可使得算法性能达到较好效果。关于这 3 个参数的敏感性分析将在实验部分的 5.2 小节中给出。

算法 1. SBA_C 伪代码.

输入: 种群 P , 种群大小 NP , 参数 $\{a, b, c\}$

输出: 更新后的种群 P

1. 根据目标函数值对种群个体进行排序, 从最好至最差;
2. FOR 种群 P 中选出最好的一半个体, $k = \{1, 2, \dots, NP/2\}$ DO
3. FOR 每一个可行值 $V_j, j = \{1, 2, \dots, s\}$ DO
4. FOR 每一维离散变量 $x_{k,M+i}, i = \{1, 2, \dots, N\}$ DO
5. IF $x_{k,M+i} = V_j$ THEN
6. $count_i = count_i + 1$;

7. END IF
8. END FOR
9. $count_i = count_i + count_{i,j}$;
10. END FOR
11. 按式(16)计算概率 $Prob_i(j)$;
12. END FOR
13. 按式(9)计算种群 P 中最好的一半个体的 N 维离散变量之间的相关性以得到协方差矩阵 C_d ;
14. 从种群 P 中选择 a 个个体;
15. FOR 每一个被选择的个体 $X_i, i = \{1, 2, \dots, a\}$ DO
16. 随机选择一个维度 j , 根据协方差矩阵 C_d 确定 $b-1$ 个相关性最高维度;
17. FOR 每一个被选择的维度 $j_l, l = \{1, 2, \dots, b\}$ DO
18. 对第 j_l 维度上所有可行值的出现频率 $Prob$ 从大到小排序;
19. 从对应的前 c 个可行值中随机选择一个用于替换原值;
20. END FOR
21. END FOR
22. 对新生成的个体执行编码, 再计算其目标函数值 $f(X_i)$;
23. $FES = FES + a$
24. 生成的新个体与父代个体进行比较, 选择更优的个体进入到下一代中;

4.5 集成变异策略框架

一般而言, DE 性能主要取决于两部分: 一是试验个体生成方式, 另一个是控制参数设置, 即种群规模 NP 、缩放因子 F 和交叉概率 CR 如何设置。在解决一些复杂优化问题时, 因仅使用单一搜索策略和固定参数值, DE 性能难以令人满意。为提高 DE 性能, 一些研究人员提出了集成变异策略的思想, 例如: Wang 等人^[34]提出的 CoDE 采用了三种变异策略和三种控制参数设置。此处, 为进一步提高 DE 求解 MVOPs 性能, 受 CoDE 启发, 我们在 ABSDE_{mv} 中也采用了类似的集成变异策略框架对连续变量和离散变量进行协同优化。所选的三种变异策略是:

- (1) DE/rand/1;
- (2) DE/current-to-best/1;
- (3) DE/best/1.

相应地, 使用了如下三组参数值:

- (1) $[F=1.0, CR=0.1]$;
- (2) $[F=1.0, CR=0.9]$;
- (3) $[F=0.8, CR=0.2]$ 。

在该框架中, 为三种变异策略随机搭配一组参数值, 可得到三种不同的试验个体, 注意交叉操作仍采用经典的二项式交叉。不难发现, 每一个目标个体

均会有三个对应的试验个体,若采用目标函数进行评估,则消耗的目标函数评估次数是经典 DE 的 3 倍,即 $3 \cdot NP$ 。为降低评估次数,我们采用欧式距离代替真实的目标函数来完成三个试验个体的评估,具体做法是:首先,计算三个试验个体到种群最优个体 X_{best}^g 的欧式距离;其次,选择欧式距离最小的试验个体作为最终的试验个体;最后,将试验个体与目标个体进行比较,选择两者中较优的一个作为后代个体。需说明的是,对这一方式的有效性验证,我们在实验部分的 5.3 小节进行了相关实验。

4.6 算法框架

与经典 DE 相比,本文提出的 $ABSDE_{mv}$ 主要有如下三点改进之处:

(1)引入了特征坐标系,并设计了一种自适应策略来切换特征坐标系和原坐标系,以发挥这两种坐

标系的各自优势.特征坐标系可松弛变量间的相关性,有利于提高算法求解不可分 MVOPs 的性能,而原坐标系可以避免种群多样性的丢失。

(2)设计了一种基于离散变量相关性的局部搜索策略专门用于优化离散变量.该策略可根据离散变量间的相关性来确定要优化的决策变量,能更加精准地对问题寻优,从而增强离散变量的优化效果。

(3)采用了一种集成变异策略框架对连续变量和离散变量进行协同优化.该框架可克服经典 DE 仅采用单一变异策略和固定参数值的局限性,提高算法的整体性能。

为更直观地理解 $ABSDE_{mv}$ 的主要过程,在图 4 中给出了算法的主要流程,同时在算法 2 中给出算法的伪代码描述。

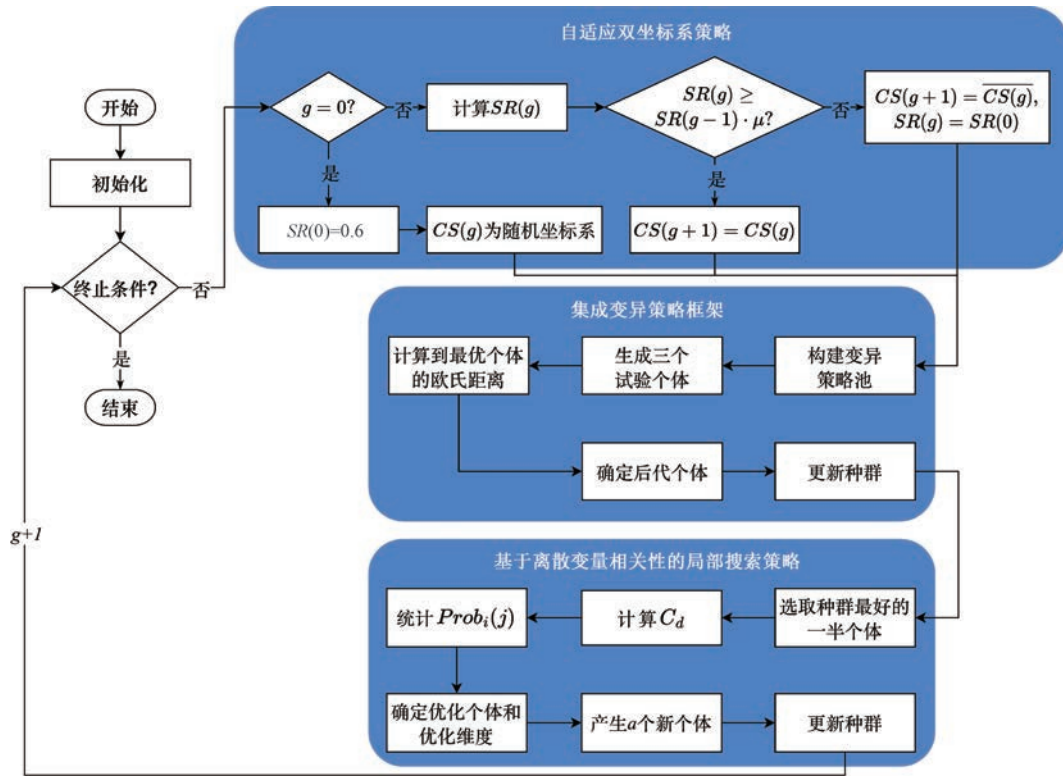


图 4 $ABSDE_{mv}$ 主要流程

算法 2. $ABSDE_{mv}$ 算法伪代码.

输入:种群 P , 种群大小 NP , 缩放因子 F , 交叉概率 CR , 最大适应度函数评估次数 $MaxFEs$, 参数 $\{a, b, c\}$

输出: g_{best} (种群最优个体)

1. 初始化种群 P , $FEs = 0$;
2. 对种群个体进行编码, 评估每个个体的目标函数值;
3. $FEs = FEs + NP$;
4. WHILE $FEs < MaxFEs$ DO
5. 执行自适应双坐标系策略来确定坐标系;

6. 执行集成变异策略框架来生成后代个体;
7. $FEs = FEs + NP$;
8. 执行基于离散变量相关性的局部搜索策略(算法 1);
9. END WHILE

5 实验验证

为验证 $ABSDE_{mv}$ 的有效性,本文在一套通用测

试函数集上进行了大量实验,主要分为以下四个部分:

(1)控制参数的敏感性分析,包括自适应双坐标系所涉及的参数 m 、坐标系使用成功率的初始值 $SR(0)$,以及基于离散变量相关性的局部搜索策略所涉及的三个参数 $\{a, b, c\}$.

(2)策略有效性验证,包括自适应双坐标系策略、基于离散变量相关性的局部搜索策略以及集成变异策略框架中的个体选择方式.

(3)与其他知名的求解 MVOPs 的 EAs 进行性能比较.

(4)求解一个实际 MVOPs,即焊接梁设计问题.

5.1 测试函数与参数设置

实验中采用了一套包含 28 个测试函数的通用测试集,在求解 MVOPs 的相关算法中也被多次使用^[14-15].这些测试函数来自于 CEC2013 测试函数^[35],但其决策变量不再只包含连续变量,而是混合变量,其中 M 维的连续变量仍保持原有特性,而对于 N 维的离散变量则做了离散化处理.对于 M 维连续变量,重构的测试函数维持了 CEC2013 测试函数的原特性,包括决策变量的搜索范围依旧为 $[-100, 100]^2$,全局最优解的偏移为 $(o_1, o_2, o_3, \dots, o_z)$ (随机分布在 $[-80, 80]^z$),旋转矩阵也相同;对于 N 维离散变量,将全局最优解的偏移转换成整数(随机分布在 $[-80, 80]^v$),并保持了原始的搜索范围 $[-100, 100]^z$ 和旋转矩阵.与原 CEC2013 测试集类似,这套重构的测试集的基准函数按其特点也可分为三类:(1)F01~F05 为单峰函数,(2)F06~F20 为基本多峰函数,(3)F21~F28 为复合函数.其中,有 24 个函数的变量间存在相关性,即不可分问题,而剩下的 4 个函数(F01、F05、F11 和 F22)的变量间无相关性,即为可分离问题.需指出的是,因同时包含了连续变量和离散变量,重构的测试集的搜索空间更加复杂,因此求解难度也要高于原测试集.

在后续四个实验部分中,最大的适应度函数

评估次数($MaxFEs$)设置为 $2000 \cdot D$,测试维度 $D = 50$,连续变量和离散变量的维度数量均为 25,种群大小 NP 设置为 30.为尽可能减少随机因素的影响,算法在每个测试函数上独立运行 30 次,并记录其平均值作为最终结果.

5.2 控制参数敏感性分析

5.2.1 参数 m 和 $SR(0)$ 敏感性分析

在自适应双坐标系策略中,参数 μ 决定了成功率 $SR(g)$ 需满足的最小值,而 m 决定了参数 μ 的取值.如果 m 的取值很大,那么成功率 $SR(g)$ 只有较大时才能持续使用当前坐标系.然而,随着迭代次数不断增加,种群生成较好后代个体的机会将不断减少,即成功率 $SR(g)$ 也会逐渐减小,那么两种坐标系之间的切换会过于频繁,这反而不利于算法找到更优解.反之,若 m 的值设置太小,则较小的 $SR(g)$ 则几乎不会切换坐标系,从而大大降低双坐标系的作用.因此,合适的 m 值,有利于选择出适合当前种群搜索状态的坐标系.在发生坐标系切换后,将成功率 $SR(g-1)$ 初始化为 $SR(0)$,可确保坐标系有适合的初始值.本小节中,我们测试了不同的 m 和 $SR(0)$ 取值对 $ABSDE_{mv}$ 性能的影响,从而选择最合适的参数值.

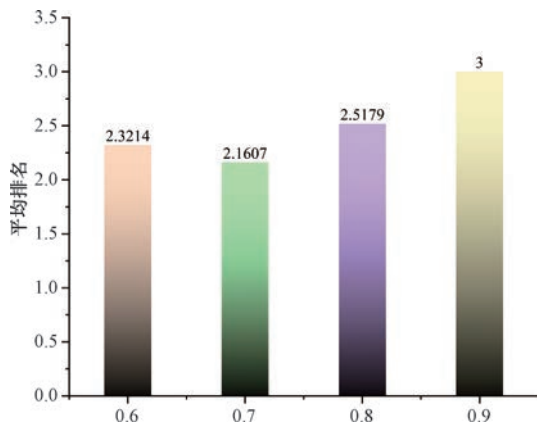
因 m 取值范围为 $[0.1, 0.9]$,所以我们选择了 9 个有代表性的取值进行测试,即 $m = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$ 或 0.9 .表 1 给出了不同 m 值的最终比较结果,其中最好结果加粗显示.同时,为简化表格,表 1 只给出其中 4 个有代表性取值的结果,即 $m = 0.6, 0.7, 0.8, 0.9$.从整体上看, $m = 0.7$ 时的性能最好,能在 12 个函数上取得最好结果, $m = 0.6$ 次之.但是, $m = 0.6$ 的结果情况与 $m = 0.7$ 相差无几,仅少一个函数,即在 11 个函数上取得了最好结果.此外,为比较四种参数值的整体性能,采用 Friedman 检验进行多问题分析.通过显著性水平为 $\alpha = 0.05$ 的 Friedman 检验,可得到四种值的平均排名情况,如图 5 所示.相比于 $m = 0.6, m = 0.7$ 以微弱优势排名第一.因此,根据比较结果, m 值默认设置为 0.7.

表 1 $ABSDE_{mv}$ 使用不同 m 值的比较结果

Func.	$m = 0.6$	$m = 0.7$	$m = 0.8$	$m = 0.9$
F01	1.86E+00±2.19E-02	1.86E+00±2.19E-02	1.86E+00±4.52E-16	1.87E+00±7.30E-02
F02	2.30E+06±7.32E+05	2.04E+06±7.32E+05	2.37E+06±9.30E+05	2.41E+06±8.08E+05
F03	1.78E+08±1.21E+08	1.36E+08±1.21E+08	2.22E+08±2.07E+08	1.56E+08±8.84E+07
F04	1.26E+04±2.37E+03	1.19E+04±2.37E+03	1.25E+04±3.18E+03	1.33E+04±4.10E+03
F05	4.33E-01±1.13E-16	4.33E-01±1.13E-16	4.41E-01±4.18E-02	4.33E-01±1.13E-16

续表

Func.	$m = 0.6$	$m = 0.7$	$m = 0.8$	$m = 0.9$
F06	5.71E+01±2.04E+01	4.68E+01±2.04E+01	4.94E+01±9.83E+00	5.09E+01±1.30E+01
F07	7.76E+01±2.24E+01	6.89E+01±2.24E+01	6.89E+01±2.47E+01	7.00E+01±1.97E+01
F08	2.11E+01±1.08E-01	2.11E+01±1.08E-01	2.11E+01±1.04E-01	2.11E+01±1.11E-01
F09	3.68E+01±5.70E+00	3.72E+01±5.70E+00	3.43E+01±5.21E+00	3.63E+01±7.05E+00
F10	4.63E+00±1.16E+00	4.43E+00±1.16E+00	4.38E+00±1.18E+00	4.84E+00±1.49E+00
F11	2.73E+01±8.41E+00	2.77E+01±8.41E+00	2.69E+01±6.55E+00	2.90E+01±7.37E+00
F12	1.24E+02±2.63E+01	1.32E+02±2.63E+01	1.40E+02±2.72E+01	1.23E+02±2.44E+01
F13	2.36E+02±4.79E+01	2.39E+02±4.79E+01	2.64E+02±5.21E+01	2.44E+02±5.77E+01
F14	9.74E+02±2.52E+02	1.01E+03±2.52E+02	8.33E+02±2.32E+02	1.01E+03±2.78E+02
F15	7.71E+03±1.20E+03	8.24E+03±1.20E+03	8.32E+03±1.01E+03	8.36E+03±1.01E+03
F16	1.23E+00±5.20E-01	1.14E+00±5.20E-01	1.36E+00±5.15E-01	1.24E+00±5.40E-01
F17	8.47E+01±5.65E+00	8.27E+01±5.65E+00	8.39E+01±4.99E+00	8.41E+01±6.62E+00
F18	2.48E+02±2.93E+01	2.63E+02±2.93E+01	2.58E+02±2.66E+01	2.48E+02±2.45E+01
F19	6.64E+00±1.30E+00	6.73E+00±1.30E+00	6.86E+00±1.14E+00	6.84E+00±1.30E+00
F20	2.08E+01±1.26E+00	2.09E+01±1.26E+00	2.10E+01±9.84E-01	2.12E+01±8.41E-01
F21	8.27E+02±3.58E+02	8.06E+02±3.58E+02	8.38E+02±3.36E+02	7.14E+02±4.39E+02
F22	9.75E+02±2.53E+02	9.78E+02±2.53E+02	1.02E+03±2.66E+02	1.08E+03±2.74E+02
F23	8.19E+03±1.26E+03	8.32E+03±1.26E+03	8.25E+03±1.35E+03	8.47E+03±1.11E+03
F24	2.83E+02±1.25E+01	2.80E+02±1.25E+01	2.86E+02±1.38E+01	2.80E+02±1.49E+01
F25	3.33E+02±2.51E+01	3.37E+02±2.51E+01	3.35E+02±1.99E+01	3.43E+02±1.86E+01
F26	3.63E+02±6.64E+01	3.64E+02±6.64E+01	3.55E+02±7.17E+01	3.72E+02±5.98E+01
F27	1.21E+03±1.54E+02	1.20E+03±1.54E+02	1.18E+03±1.20E+02	1.19E+03±1.24E+02
F28	1.23E+03±1.38E+03	4.08E+02±1.38E+03	8.20E+02±1.07E+03	9.15E+02±1.15E+03

图 5 不同 m 值的平均排名情况

同样地,我们选择了 9 个有代表性的 $SR(0)$ 值进行实验,即 $SR(0) = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$ 或 0.9 . 表 2 给出了不同 $SR(0)$ 值的最终比较结果,其中最好的结果也加粗显示. 类似地,表 2 也只给出 4 个具有代表性取值的结果,即 $SR(0) = 0.5, 0.6, 0.7, 0.8$. 从表中可看出, $SR(0) = 0.6$ 时,算法在 28 个测试函数中的 12 个函数上获得了最好结果,而采用其他值时取得最好结果的函数个数均不超过 10 个. 另外,通过 Friedman 检验得到的平均排名如图 6 所示, $SR(0) = 0.6$ 以较大优势排名第一. 因此,根据比较结果, $ABSDE_{mv}$ 中 $SR(0)$ 默认设置为 0.6.

5.2.2 参数 $\{a, b, c\}$ 敏感性分析

基于离散变量相关性的局部搜索策略 (SBA_C) 有三个重要参数: $\{a, b, c\}$, a 值决定了有多少个个体能进行局部搜索, b 值控制了局部搜索的维度数, c 值表示可行值选择的范围,会影响种群的收敛速度. 为确定这三个参数的合理取值,本节进行了参数敏感性分析. 这三个参数的可能取值均为 $[3, 5, 10]$, 因此有 27 种不同的参数值组合方案. 我们测试这 27 组方案对 $ABSDE_{mv}$ 性能的影响,其他参数均保持不变. 表 3 给出了不同组合方案的实验结果,其中最好结果用粗体突显. 为简化表格结果,表 3 只展示 4 组具有代表性组合的结果,即 $\{a, b, c\} = \{3, 3, 3\}, \{3, 3, 5\}, \{3, 3, 10\}, \{5, 3, 5\}$. 另外,为进一步验证 SBA_C 策略的有效性,我们构造了一个去除 SBA_C 策略的对比算法 $SBA_C_{without}$.

从表 3 可看出,四种方案的结果均优于或等于 $SBA_C_{without}$ 算法,这也从侧面验证了 SBA_C 策略是有效的. 在四种方案中, $SBA_C_{5,3,5}$ 取得最佳结果,在 28 个测试函数中有 11 个函数上取得了最好结果,其次是 $SBA_C_{3,3,3}$ 取得了 8 个最好结果. 图 7 展示了这五个算法通过 Friedman 检验得到的平均排名, $SBA_C_{5,3,5}$ 排名第一. 因此,我们将 $\{a, b, c\}$ 的值设置为 $\{5, 3, 5\}$, 下文中的所有实验都按此设置进行.

表 2 ABSDE_{mv} 使用不同 SR(0) 值的比较结果

Func.	SR(0) = 0.5	SR(0) = 0.6	SR(0) = 0.7	SR(0) = 0.8
F01	1.86E+00±4.52E-16	1.86E+00±4.52E-16	1.87E+00±2.74E-02	1.87E+00±2.74E-02
F02	2.29E+06±6.64E+05	2.04E+06±5.60E+05	2.29E+06±5.87E+05	2.40E+06±9.48E+05
F03	1.73E+08±1.33E+08	1.36E+08±6.34E+07	1.55E+08±9.69E+07	1.78E+08±1.60E+08
F04	1.36E+04±3.74E+03	1.19E+04±3.48E+03	1.33E+04±3.57E+03	1.27E+04±4.10E+03
F05	4.33E-01±1.13E-16	4.33E-01±1.13E-16	4.33E-01±1.13E-16	4.33E-01±1.13E-16
F06	5.08E+01±1.27E+01	4.68E+01±3.67E+00	5.09E+01±1.42E+01	5.47E+01±1.84E+01
F07	7.42E+01±2.60E+01	6.89E+01±1.71E+01	7.33E+01±2.49E+01	6.23E+01±1.87E+01
F08	2.11E+01±1.12E-01	2.11E+01±1.03E-01	2.11E+01±8.60E-02	2.11E+01±1.17E-01
F09	3.56E+01±5.54E+00	3.72E+01±6.65E+00	3.77E+01±6.10E+00	3.73E+01±6.01E+00
F10	4.67E+00±1.19E+00	4.43E+00±1.69E+00	5.08E+00±1.56E+00	4.63E+00±1.23E+00
F11	2.85E+01±1.05E+01	2.77E+01±7.69E+00	2.60E+01±6.27E+00	2.64E+01±6.19E+00
F12	1.35E+02±2.68E+01	1.32E+02±2.52E+01	1.36E+02±2.33E+01	1.36E+02±2.15E+01
F13	2.55E+02±4.38E+01	2.39E+02±4.51E+01	2.39E+02±4.60E+01	2.65E+02±5.79E+01
F14	1.02E+03±2.87E+02	1.01E+03±2.54E+02	9.98E+02±2.26E+02	9.70E+02±2.37E+02
F15	7.96E+03±1.32E+03	8.24E+03±9.58E+02	8.17E+03±1.01E+03	8.22E+03±1.02E+03
F16	1.26E+00±6.54E-01	1.14E+00±5.03E-01	1.52E+00±8.57E-01	1.48E+00±7.39E-01
F17	8.13E+01±4.18E+00	8.27E+01±5.84E+00	8.35E+01±4.38E+00	8.31E+01±5.79E+00
F18	2.50E+02±2.97E+01	2.63E+02±2.62E+01	2.44E+02±2.43E+01	2.47E+02±2.73E+01
F19	6.83E+00±1.04E+00	6.73E+00±9.16E-01	6.98E+00±1.23E+00	7.03E+00±1.27E+00
F20	2.12E+01±8.11E-01	2.09E+01±9.04E-01	2.11E+01±8.86E-01	2.11E+01±1.11E+00
F21	8.67E+02±3.19E+02	8.06E+02±3.75E+02	8.66E+02±3.47E+02	8.75E+02±3.50E+02
F22	1.01E+03±3.14E+02	9.78E+02±3.61E+02	1.01E+03±2.63E+02	1.03E+03±3.20E+02
F23	8.50E+03±1.36E+03	8.32E+03±1.33E+03	8.32E+03±1.14E+03	8.54E+03±1.25E+03
F24	2.83E+02±1.71E+01	2.80E+02±1.22E+01	2.79E+02±1.78E+01	2.83E+02±1.26E+01
F25	3.32E+02±2.81E+01	3.37E+02±2.41E+01	3.37E+02±2.54E+01	3.39E+02±2.52E+01
F26	3.64E+02±6.73E+01	3.64E+02±6.63E+01	3.89E+02±1.62E+01	3.58E+02±7.35E+01
F27	1.23E+03±1.34E+02	1.20E+03±1.53E+02	1.20E+03±9.47E+01	1.24E+03±1.19E+02
F28	7.17E+02±9.44E+02	4.08E+02±4.98E-01	6.11E+02±7.72E+02	7.13E+02±9.29E+02

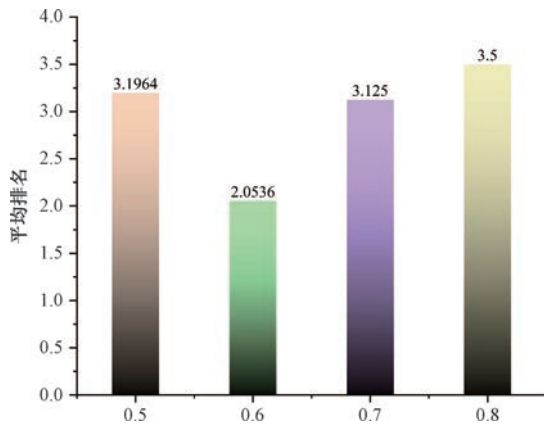


图 6 不同 SR(0) 值的平均排名情况

5.3 策略有效性验证

在本小节中,我们设计了一系列试验来验证自适应双坐标系策略、基于离散变量相关性的局部搜索策略以及集成变异策略框架中所涉及到的个体选择方式的有效性。

5.3.1 自适应双坐标系策略的有效性验证

在此实验中,为验证自适应双坐标系策略的有效性,我们构造了三种对比算法,如下所示:

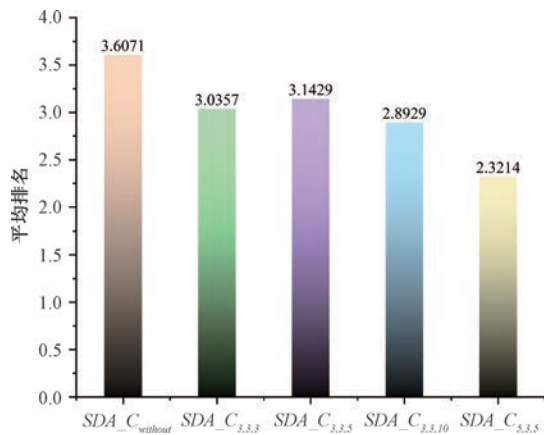
- (1) ABSDE_{mv_E}: 仅使用特征坐标系;
- (2) ABSDE_{mv_I}: 仅使用原坐标系;
- (3) ABSDE_{mv_R}: 按相同概率随机选择特征坐标系或原坐标系。

上述三个对比算法与 ABSDE_{mv} 可分成两类:一类是 ABSDE_{mv_E} 和 ABSDE_{mv_I}, 它们仅使用单一的坐标系;另一类是 ABSDE_{mv_R} 和 ABSDE_{mv}, 它们同时使用了双坐标系,区别在于如何使用这两种坐标系。为公平比较,这三种对比算法的参数设置均与 ABSDE_{mv} 保持一致。通过与这三种对比算法的性能比较,可直观地分析自适应双坐标系策略对 ABSDE_{mv} 的影响。同时,为使对比结果更具统计意义,采用了无参数的 Wilcoxon's 秩和检验,显著性水平 α 设置为 0.05。

表 4 给出了所涉及的四种算法的比较结果,且最好结果加粗显示。Wilcoxon's 秩和检验的结果汇总在表 4 最后一行,符号“+”,“=”和“-”分别表示对比算法的结果好于、相似和差于 ABSDE_{mv}。从表中可看出,ABSDE_{mv} 总体性能在所有对比算法中表现最好,能在绝大多数的测试函数上取得最好结果,

表 3 ABSDE_{mv} 使用不同 $\{a, b, c\}$ 值的比较结果

Func.	SBA_C _{without}	SBA_C _{3,3,3}	SBA_C _{3,3,5}	SBA_C _{3,3,10}	SBA_C _{5,3,5}
F01	2.26E+00±4.21E-01	1.87E+00±4.93E-02	1.86E+00±4.52E-16	1.89E+00±7.64E-02	1.86E+00±4.52E-16
F02	1.98E+06±4.39E+05	2.32E+06±8.62E+05	2.39E+06±9.34E+05	2.27E+06±6.54E+05	2.04E+06±5.60E+05
F03	1.78E+08±1.01E+08	2.35E+08±2.72E+08	2.35E+08±1.87E+08	1.81E+08±1.72E+08	1.36E+08±6.34E+07
F04	1.05E+04±3.07E+03	1.08E+04±3.01E+03	1.06E+04±3.01E+03	1.15E+04±3.00E+03	1.19E+04±3.48E+03
F05	6.61E-01±2.10E-01	4.36E-01±1.79E-02	4.33E-01±1.13E-16	4.85E-01±9.46E-02	4.33E-01±1.13E-16
F06	4.95E+01±9.94E+00	5.24E+01±1.60E+01	5.23E+01±1.58E+01	5.30E+01±1.83E+01	4.68E+01±3.67E+00
F07	6.81E+01±1.94E+01	7.21E+01±2.40E+01	7.04E+01±2.27E+01	6.95E+01±1.89E+01	6.89E+01±1.71E+01
F08	2.11E+01±1.01E-01	2.11E+01±1.02E-01	2.11E+01±9.97E-02	2.11E+01±9.73E-02	2.11E+01±1.03E-01
F09	3.50E+01±6.24E+00	3.55E+01±5.41E+00	3.73E+01±6.79E+00	3.71E+01±5.61E+00	3.72E+01±6.65E+00
F10	5.53E+00±2.14E+00	5.07E+00±1.68E+00	5.10E+00±1.29E+00	4.53E+00±1.50E+00	4.43E+00±1.69E+00
F11	4.65E+01±1.24E+01	3.27E+01±9.00E+00	2.73E+01±7.82E+00	2.73E+01±8.14E+00	2.77E+01±7.69E+00
F12	1.36E+02±2.90E+01	1.31E+02±2.70E+01	1.35E+02±3.38E+01	1.32E+02±2.30E+01	1.32E+02±2.52E+01
F13	2.54E+02±4.30E+01	2.42E+02±5.41E+01	2.49E+02±3.83E+01	2.40E+02±4.65E+01	2.39E+02±4.51E+01
F14	1.45E+03±4.95E+02	1.21E+03±3.40E+02	9.99E+02±2.52E+02	8.15E+02±2.56E+02	1.01E+03±2.54E+02
F15	8.16E+03±9.60E+02	8.53E+03±1.07E+03	8.45E+03±9.64E+02	7.61E+03±1.33E+03	8.24E+03±9.58E+02
F16	1.26E+00±7.15E-01	1.42E+00±6.34E-01	1.13E+00±4.26E-01	1.37E+00±5.85E-01	1.14E+00±5.03E-01
F17	9.38E+01±1.21E+01	8.84E+01±6.82E+00	8.22E+01±4.49E+00	7.96E+01±3.74E+00	8.27E+01±5.84E+00
F18	2.41E+02±3.14E+01	2.46E+02±2.78E+01	2.52E+02±3.05E+01	2.51E+02±2.89E+01	2.63E+02±2.62E+01
F19	8.99E+00±1.78E+00	7.54E+00±1.30E+00	6.61E+00±1.37E+00	6.44E+00±1.01E+00	6.73E+00±9.16E-01
F20	2.11E+01±1.16E+00	2.12E+01±1.04E+00	2.12E+01±1.17E+00	2.11E+01±7.59E-01	2.09E+01±9.04E-01
F21	8.37E+02±3.61E+02	7.06E+02±4.12E+02	8.19E+02±3.27E+02	7.96E+02±3.94E+02	8.06E+02±3.75E+02
F22	1.73E+03±4.70E+02	1.29E+03±3.21E+02	1.02E+03±3.71E+02	7.65E+02±2.87E+02	9.78E+02±3.61E+02
F23	8.42E+03±1.46E+03	8.25E+03±1.13E+03	8.37E+03±1.19E+03	8.70E+03±1.17E+03	8.32E+03±1.33E+03
F24	2.83E+02±1.32E+01	2.82E+02±1.71E+01	2.85E+02±1.43E+01	2.81E+02±1.74E+01	2.80E+02±1.22E+01
F25	3.38E+02±2.38E+01	3.37E+02±2.25E+01	3.39E+02±2.53E+01	3.38E+02±2.49E+01	3.37E+02±2.41E+01
F26	3.89E+02±1.28E+01	3.63E+02±5.63E+01	3.72E+02±5.94E+01	3.83E+02±3.72E+01	3.64E+02±6.63E+01
F27	1.22E+03±1.44E+02	1.19E+03±1.65E+02	1.19E+03±1.18E+02	1.24E+03±1.40E+02	1.20E+03±1.53E+02
F28	1.22E+03±1.37E+03	4.08E+02±6.51E-01	1.12E+03±1.31E+03	7.13E+02±9.28E+02	4.08E+02±4.98E-01

图 7 不同 $\{a, b, c\}$ 值的平均排名情况

在 28 个函数中的 16 个函数上取得了最好结果. 具体而言, 与 ABSDE_{mv}_E 和 ABSDE_{mv}_I 相比 ABSDE_{mv}, 分别在 15 个、16 个测试函数上要更优, 且仅在 2 个、5 个测试函数上要弱于 ABSDE_{mv}_E 和 ABSDE_{mv}_I. 这一比较结果验证了, 同时使用原坐标系和特征坐标系的效果要显著地优于仅使用单一坐标系. 与 ABSDE_{mv}_R 相比, ABSDE_{mv} 在 13 个

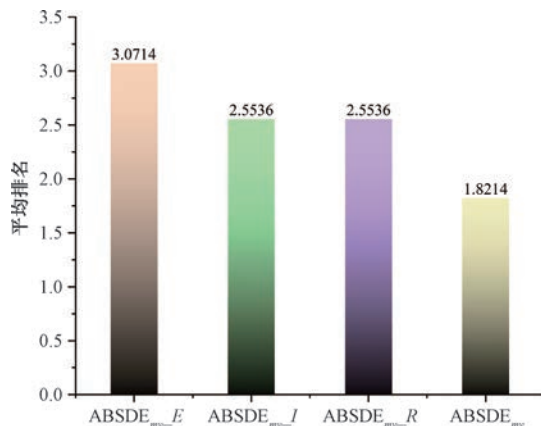
测试函数上更优, 仅在 5 个测试函数上结果略差, 在剩下的 10 个测试函数上两者性能相近; 由此可得出, 在使用双坐标系方面, 本文提出的自适应方式比随机方式要更好. ABSDE_{mv} 与 ABSDE_{mv}_I 在 24 个不可分问题上进行了进一步比较, 结果表明前者在 24 个问题上取得了 12 个最好结果, 仅在 5 个问题上差于前者. 因此, 可以得出结论, 引入特征坐标系松弛变量之间的相关性, 可以有效提高算法求解变量相关的 MVOPs 的性能. 为简化算法对比情况, 图 8 给出了使用 Friedman 检验得到的算法平均排名, 可看到 ABSDE_{mv} 排名最靠前.

此外, 为进一步讨论不同采样方式对算法性能的影响, 此处对比了三种采样方式: (1) 选择种群中适应度值最差的一半个体; (2) 随机选择种群中的一半个体; (3) 选择种群中适应度值中间的一半个体. 基于以上采样方式构造的对比算法分别记作 ABSDE_{mv}-1、ABSDE_{mv}-2 和 ABSDE_{mv}-3. 对这三种对比算法而言, 除采样方式不同外, 其余的算法部分均相同.

表 5 给出了所涉及的四种种算法的实验结果, 最

表 4 自适应双坐标系策略的有效性验证

Func.	ABSDE _{mv_E}	ABSDE _{mv_I}	ABSDE _{mv_R}	ABSDE _{mv}
F01	2.01E+00±1.96E-01-	1.90E+00±8.29E-02-	1.89E+00±8.61E-02-	1.86E+00±4.52E-16
F02	1.78E+06±6.78E+05=	3.07E+06±1.04E+06-	2.35E+06±1.09E+06=	2.04E+06±5.60E+05
F03	3.92E+08±2.35E+08-	1.52E+08±1.80E+08=	1.81E+08±1.28E+08=	1.36E+08±6.34E+07
F04	2.17E+04±5.28E+03-	1.52E+04±3.51E+03-	1.62E+04±3.51E+03-	1.19E+04±3.48E+03
F05	1.43E+01±1.03E+01-	5.68E-01±2.67E-01-	5.07E-01±1.89E-01-	4.33E-01±1.13E-16
F06	5.92E+01±2.21E+01=	4.75E+01±1.20E+00=	5.14E+01±1.30E+01=	4.68E+01±3.67E+00
F07	9.87E+01±2.36E+01-	9.78E+01±3.88E+01-	8.82E+01±2.83E+01-	6.89E+01±1.71E+01
F08	2.11E+01±9.61E-02=	2.12E+01±5.35E-02-	2.12E+01±6.51E-02-	2.11E+01±1.03E-01
F09	3.78E+01±4.42E+00=	3.36E+01±4.16E+00=	3.64E+01±4.10E+00=	3.72E+01±6.65E+00
F10	3.82E+00±1.04E+00=	3.80E+00±1.24E+00=	3.08E+00±6.57E-01+	4.43E+00±1.69E+00
F11	1.74E+02±2.80E+01-	8.33E+01±1.18E+01-	1.35E+02±2.14E+01-	2.77E+01±7.69E+00
F12	1.52E+02±2.52E+01-	9.91E+01±1.90E+01+	1.09E+02±2.27E+01+	1.32E+02±2.52E+01
F13	2.72E+02±3.64E+01-	2.06E+02±5.49E+01+	2.26E+02±4.03E+01=	2.39E+02±4.51E+01
F14	6.85E+03±1.22E+03-	3.77E+03±5.36E+02-	5.83E+03±4.79E+02-	1.01E+03±2.54E+02
F15	6.98E+03±7.97E+02+	9.03E+03±2.19E+03=	7.37E+03±1.91E+03+	8.24E+03±9.58E+02
F16	1.44E+00±1.23E+00=	3.66E+00±3.16E-01-	2.67E+00±1.38E+00-	1.14E+00±5.03E-01
F17	2.30E+02±2.93E+01-	1.50E+02±1.20E+01-	2.04E+02±1.58E+01-	8.27E+01±5.84E+00
F18	2.79E+02±2.34E+01-	3.42E+02±7.94E+01-	2.56E+02±5.04E+01=	2.63E+02±2.62E+01
F19	1.74E+01±3.67E+00-	1.85E+01±2.35E+00-	1.95E+01±3.58E+00-	6.73E+00±9.16E-01
F20	2.11E+01±8.35E-01=	2.22E+01±5.00E-01-	2.19E+01±6.50E-01-	2.09E+01±9.04E-01
F21	7.76E+02±4.07E+02=	8.24E+02±4.06E+02=	8.25E+02±3.83E+02=	8.06E+02±3.75E+02
F22	7.27E+03±8.37E+02-	4.98E+03±7.45E+02-	6.68E+03±6.30E+02-	9.78E+02±3.61E+02
F23	7.59E+03±7.78E+02+	9.73E+03±2.27E+03-	8.37E+03±1.32E+03=	8.32E+03±1.33E+03
F24	2.79E+02±1.33E+01=	2.62E+02±8.46E+00+	2.64E+02±9.87E+00+	2.80E+02±1.22E+01
F25	3.67E+02±2.24E+01-	3.89E+02±3.70E+00-	3.83E+02±1.35E+01-	3.37E+02±2.41E+01
F26	3.16E+02±9.64E+01=	2.67E+02±8.90E+01+	2.78E+02±9.16E+01+	3.64E+02±6.63E+01
F27	1.23E+03±9.53E+01=	1.11E+03±1.18E+02+	1.15E+03±1.15E+02=	1.20E+03±1.53E+02
F28	5.14E+02±5.73E+02-	4.09E+02±8.14E-01=	4.08E+02±7.30E-01=	4.08E+02±4.98E-01
+ / = / -	2/11/15	5/7/16	5/10/13	- / - / -

图 8 ABSDE_{mv_E}、ABSDE_{mv_I}、ABSDE_{mv_R} 和 ABSDE_{mv} 的平均排名情况

好结果以粗体突显. 同样, 也使用 Wilcoxon's 秩和检验进行统计检验. 如表 5 所示, ABSDE_{mv} 要明显优于其他三种对比算法, 尤其是单峰函数上, 均取得最好结果. 图 9 展示了所涉及的四中算法基于 Friedman 检验的平均排名, 可看出 ABSDE_{mv} 排名第一.

5.3.2 基于离散变量相关性的局部搜索策略的有效性验证

为更好地优化 MVOPs 中的离散变量, 我们提出了基于离散变量相关性的局部搜索策略 (SBA_C). 值得注意的是, Peng 等人^[13] 提出的 MCDE_{mv} 也使用了类似的离散变量优化策略, 即基于统计的局部搜索机制 (SBA). 在 SBA 中, 从当前种群中最好的一半个体中随机选择 a 个个体, 并随机选择 b 维离散变量进行局部搜索. 为验证 SBA_C 策略的有效性, 我们构造了如下两种比较算法:

(1) ABSDE_{mv_non}: 移除 SBA_C 策略 ABSDE_{mv} 的算法.

(2) ABSDE_{mv_peng}: 移除 SBA_C 策略, 使用 SBA 机制的 ABSDE_{mv} 算法.

此外, 我们将 MCDE_{mv} 加入比较, 以进一步验证所提出的 SBA_C 策略的有效性和本文算法性能. ABSDE_{mv} 与 ABSDE_{mv_non}、ABSDE_{mv_peng} 之间的区别仅在于算法对于离散变量优化的方法不同. ABSDE_{mv_non} 未单独对离散变量进行优化,

表 5 不同采样方式的结果对比

Func.	ABSDE _{mv} -1	ABSDE _{mv} -2	ABSDE _{mv} -3	ABSDE _{mv}
F01	1.86E+00±2.19E-02=	1.86E+00±4.52E-16=	1.86E+00±6.78E-16=	1.86E+00±4.52E-16
F02	4.17E+06±1.65E+06-	4.55E+06±1.68E+06-	3.69E+06±1.03E+06-	2.04E+06±5.60E+05
F03	3.02E+08±3.34E+08-	3.99E+08±3.98E+08-	3.42E+08±2.24E+08-	1.36E+08±6.34E+07
F04	5.05E+04±7.08E+03-	4.25E+04±8.61E+03-	4.49E+04±7.60E+03-	1.19E+04±3.48E+03
F05	4.37E-01±1.08E-02=	4.38E-01±1.71E-02=	4.33E-01±5.65E-17=	4.33E-01±1.13E-16
F06	5.28E+01±1.58E+01=	5.39E+01±1.83E+01=	5.11E+01±1.30E+01=	4.68E+01±3.67E+00
F07	7.47E+01±1.68E+01=	8.00E+01±1.47E+01-	7.65E+01±1.93E+01=	6.89E+01±1.71E+01
F08	2.12E+01±3.79E-02-	2.12E+01±4.61E-02-	2.12E+01±4.46E-02-	2.11E+01±1.03E-01
F09	4.13E+01±1.11E+01=	4.19E+01±9.23E+00-	3.89E+01±9.76E+00=	3.72E+01±6.65E+00
F10	5.41E+00±1.41E+00-	5.80E+00±1.67E+00-	6.54E+00±1.97E+00-	4.43E+00±1.69E+00
F11	1.68E+01±4.57E+00+	1.95E+01±4.83E+00+	1.74E+01±5.33E+00+	2.77E+01±7.69E+00
F12	1.34E+02±5.62E+01=	1.35E+02±3.51E+01=	1.25E+02±3.05E+01=	1.32E+02±2.52E+01
F13	3.21E+02±7.80E+01-	2.95E+02±5.54E+01-	3.00E+02±6.72E+01-	2.39E+02±4.51E+01
F14	6.88E+02±2.19E+02+	9.20E+02±2.50E+02=	7.34E+02±2.24E+02+	1.01E+03±2.54E+02
F15	1.28E+04±2.03E+03-	1.02E+04±3.00E+03=	1.24E+04±2.39E+03-	8.24E+03±9.58E+02
F16	3.73E+00±3.59E-01-	3.54E+00±9.56E-01-	3.75E+00±3.46E-01-	1.14E+00±5.03E-01
F17	7.76E+01±2.59E+00+	8.13E+01±3.14E+00=	7.78E+01±2.93E+00+	8.27E+01±5.84E+00
F18	4.07E+02±4.41E+01-	3.44E+02±8.24E+01-	3.88E+02±6.53E+01-	2.63E+02±2.62E+01
F19	9.73E+00±3.00E+00-	1.04E+01±2.75E+00-	8.55E+00±2.68E+00-	6.73E+00±9.16E-01
F20	2.25E+01±7.05E-01-	2.23E+01±7.00E-01-	2.23E+01±7.75E-01-	2.09E+01±9.04E-01
F21	9.18E+02±2.71E+02=	7.67E+02±3.79E+02=	9.19E+02±2.72E+02=	8.06E+02±3.75E+02
F22	7.65E+02±2.37E+02+	9.39E+02±2.46E+02=	6.99E+02±2.20E+02+	9.78E+02±3.61E+02
F23	1.35E+04±1.87E+03-	1.01E+04±2.78E+03-	1.28E+04±2.33E+03-	8.32E+03±1.33E+03
F24	2.68E+02±1.44E+01+	2.74E+02±1.88E+01=	2.69E+02±1.67E+01+	2.80E+02±1.22E+01
F25	3.72E+02±1.60E+01-	3.54E+02±3.06E+01-	3.64E+02±2.34E+01-	3.37E+02±2.41E+01
F26	3.13E+02±9.42E+01=	3.30E+02±8.68E+01=	3.14E+02±9.43E+01=	3.64E+02±6.63E+01
F27	1.19E+03±1.22E+02=	1.18E+03±1.81E+02=	1.15E+03±1.39E+02=	1.20E+03±1.53E+02
F28	5.12E+02±5.72E+02-	6.12E+02±7.76E+02=	6.16E+02±7.93E+02=	4.08E+02±4.98E-01
+ / = / -	5/9/14	1/13/14	5/10/13	- / - / -

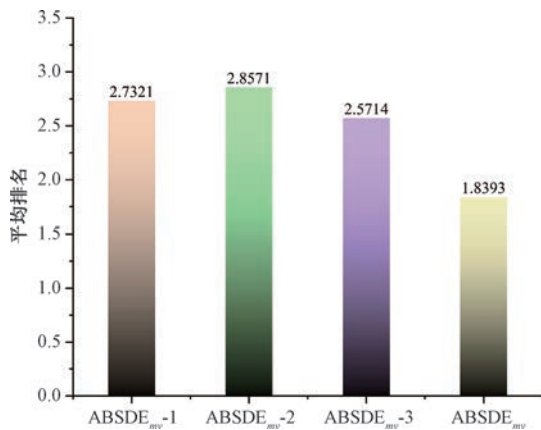


图 9 ABSDE_{mv}-1、ABSDE_{mv}-2、ABSDE_{mv}-3 和 ABSDE_{mv} 的平均排名情况

ABSDE_{mv}_peng 采用了 MCDE_{mv} 的 SBA 机制。

表 6 给出了这四种算法的实验结果,最好结果加粗显示. 同样,也使用 Wilcoxon's 秩和检验对结果进行统计检测. 从表中可看出, ABSDE_{mv} 在大部分测试函数上要优于 ABSDE_{mv}_non、ABSDE_{mv}_peng 和 MCDE_{mv}. 因此,可得出结论, ABSDE_{mv} 采用的

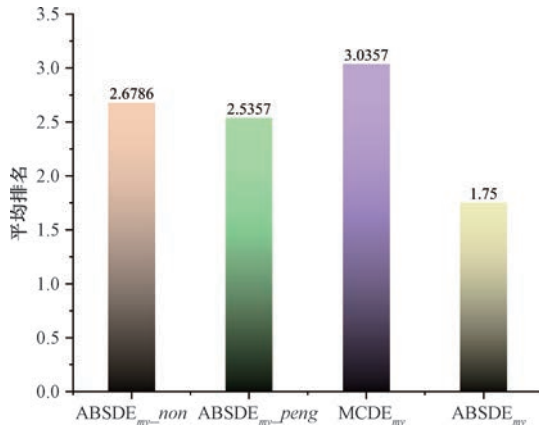
SBA_C 策略是有效的,且优于 MCDE_{mv} 的 SBA 方法. 与 ABSDE_{mv}_non 相比, ABSDE_{mv} 在 9 个测试函数上更优,仅在 1 个测试函数上弱一些,这很好地验证了 SBA_C 策略是有效的. ABSDE_{mv} 在 9 个测试函数上优于 ABSDE_{mv}_peng,仅在 2 个测试函数上弱于 ABSDE_{mv}_peng,同样验证了 SBA_C 策略是有效的. 最后,与 MCDE_{mv} 比较, ABSDE_{mv} 显著地更优,在 15 个测试函数上要优于 MCDE_{mv},仅在 5 个测试函数上差于 MCDE_{mv}. 综上,不论是 SBA_C 策略还是算法的总体性能, ABSDE_{mv} 都要优于 MCDE_{mv}. 图 10 展示了所涉及的四中算法基于 Friedman 检验的平均排名. 很明显, ABSDE_{mv} 排名第一,而 MCDE_{mv} 排名最后,这有力佐证了上述结论.

5.3.3 个体选择方式的有效性验证

在 ABSDE_{mv} 的集成变异策略框架中,为减少目标函数的评估次数,把试验个体与种群最优个体 X^g_{best} 的欧式距离作为评价指标,将距离最小的试验个体作为候选个体. 为验证这一方式的有效性,我们

表 6 SBA_C 策略的有效性验证

Func.	ABSDE _{mv_non}	ABSDE _{mv_peng}	MCDE _{mv}	ABSDE _{mv}
F01	2.26E+00±4.21E-01-	1.87E+00±4.63E-02-	3.33E-02±1.80E-01+	1.86E+00±4.52E-16
F02	1.98E+06±4.39E+05=	2.07E+06±6.13E+05=	3.80E+06±1.03E+06-	2.04E+06±5.60E+05
F03	1.78E+08±1.01E+08=	1.61E+08±8.70E+07=	4.55E+08±2.96E+08-	1.36E+08±6.34E+07
F04	1.05E+04±3.07E+03=	1.06E+04±3.30E+03=	2.38E+04±4.92E+03-	1.19E+04±3.48E+03
F05	6.61E-01±2.10E-01-	4.37E-01±2.06E-02-	1.33E-01±3.40E-01+	4.33E-01±1.13E-16
F06	4.95E+01±9.94E+00=	5.47E+01±1.92E+01-	4.78E+01±9.27E-01=	4.68E+01±3.67E+00
F07	6.81E+01±1.94E+01=	8.14E+01±3.15E+01=	7.20E+01±1.61E+01=	6.89E+01±1.71E+01
F08	2.11E+01±1.01E-01=	2.11E+01±9.99E-02=	2.12E+01±2.76E-02-	2.11E+01±1.03E-01
F09	3.50E+01±6.24E+00=	3.49E+01±6.01E+00=	4.46E+01±7.75E+00-	3.72E+01±6.65E+00
F10	5.53E+00±2.14E+00-	3.85E+00±1.23E+00=	2.98E+00±1.67E-01+	4.43E+00±1.69E+00
F11	4.65E+01±1.24E+01-	7.49E+01±2.11E+01-	3.06E+01±9.52E+00=	2.77E+01±7.69E+00
F12	1.36E+02±2.90E+01=	1.45E+02±3.63E+01=	1.90E+02±6.16E+01-	1.32E+02±2.52E+01
F13	2.54E+02±4.30E+01=	2.40E+02±4.38E+01=	2.70E+02±4.95E+01-	2.39E+02±4.51E+01
F14	1.45E+03±4.95E+02-	1.91E+03±3.84E+02-	8.67E+02±4.58E+02+	1.01E+03±2.54E+02
F15	8.16E+03±9.60E+02=	7.62E+03±1.01E+03+	9.42E+03±1.77E+03-	8.24E+03±9.58E+02
F16	1.26E+00±7.15E-01=	1.26E+00±5.36E-01=	3.42E+00±8.53E-01-	1.14E+00±5.03E-01
F17	9.38E+01±1.21E+01-	1.18E+02±1.28E+01-	1.08E+02±1.14E+01-	8.27E+01±5.84E+00
F18	2.41E+02±3.14E+01+	2.54E+02±3.37E+01=	3.04E+02±6.86E+01-	2.63E+02±2.62E+01
F19	8.99E+00±1.78E+00-	8.03E+00±1.81E+00-	6.46E+00±1.20E+00=	6.73E+00±9.16E-01
F20	2.11E+01±1.16E+00=	2.10E+01±7.75E-01=	2.16E+01±8.24E-01-	2.09E+01±9.04E-01
F21	8.37E+02±3.61E+02=	8.58E+02±3.16E+02=	4.00E+02±5.55E-01+	8.06E+02±3.75E+02
F22	1.73E+03±4.70E+02-	2.44E+03±8.03E+02-	8.92E+02±3.14E+02=	9.78E+02±3.61E+02
F23	8.42E+03±1.46E+03=	7.86E+03±1.28E+03=	1.04E+04±1.98E+03-	8.32E+03±1.33E+03
F24	2.83E+02±1.32E+01=	2.86E+02±1.45E+01=	2.72E+02±1.62E+01=	2.80E+02±1.22E+01
F25	3.38E+02±2.38E+01=	3.23E+02±1.79E+01+	3.38E+02±1.24E+01=	3.37E+02±2.41E+01
F26	3.89E+02±1.28E+01=	3.83E+02±3.79E+01=	3.92E+02±4.05E+01-	3.64E+02±6.63E+01
F27	1.22E+03±1.44E+02=	1.21E+03±1.29E+02=	1.25E+03±2.07E+02=	1.20E+03±1.53E+02
F28	1.22E+03±1.37E+03-	1.02E+03±1.25E+03-	2.14E+03±2.96E+02-	4.08E+02±4.98E-01
+ / = / -	1 / 18 / 9	2 / 17 / 9	5 / 8 / 15	- / - / -

图 10 ABSDE_{mv_non}、ABSDE_{mv_peng}、MCDE_{mv} 和 ABSDE_{mv} 的平均排名情况

采用了其他三种方式与该方式进行比较,包括:随机选择方式、按距离最远的方式、Parzen Window 方法^[36].基于这三种方式,我们构造了如下三种对应的比较算法:

(1) ABSDE_{mv_rnd}:从三个试验个体中随机选择一个作为候选个体.

(2) ABSDE_{mv_max}:根据三个试验个体与种群最优个体 \mathbf{X}_{best}^k 的欧式距离作为评价指标,选择距离最远的试验个体作为候选个体.

(3) ABSDE_{mv_pwd}:使用 Parzen Window 方法对三个试验个体进行评估,选择概率密度最小的作为后代个体.

上述第三个对比算法 ABSDE_{mv_pwd} 中的 Parzen Window 方法是 EAs 中一种非常有代表性的代理模型方法,通过构建概率密度函数来代替真实的目标函数进行个体评估,其计算方式如下^[36]:

$$\hat{F}(y) = \frac{1}{\mu} \sum_{i=1}^{\mu} \left(\frac{R_i}{\mu} \frac{1}{w} \varphi \left(\frac{\|y - x_i\|_2}{w} \right) \right) \quad (17)$$

其中, $\|x\|_2 = \sqrt{\sum_{j=1}^n x_j^2}$ 表示为 L_2 范式, $R_i = \mu - i + 1$ 为个体 X_i 在有序种群中的降序排名; $\varphi(u) = \frac{3}{4}(1 - u^2)1_{(|u| \leq 1)}$ 为 Epanechnikov 核; $w =$

$\sqrt{\frac{1}{n} \sum_{j=1}^n (\bar{a}_j - \bar{b}_j)^2}$ 为窗口宽度, \bar{a}_j 和 \bar{b}_j 为第 j 维

的上下边界.

表 7 给出了所涉及的四种算法的实验结果,最好的结果加粗显示. 同样,也使用 Wilcoxon's 秩和检验进行统计检验. 如表 7 所示, $ABSDE_{mv}$ 要明显优于 $ABSDE_{mv_rnd}$ 和 $ABSDE_{mv_max}$, 仅在 3 个和 1 个函数上结果略差. 而对于 $ABSDE_{mv_pwd}$, $ABSDE_{mv}$ 与其性能相当, 仅在 3 个函数上差于 $ABSDE_{mv_pwd}$, 在 5 个函数上要优于该算法. 同

时, $ABSDE_{mv}$ 和 $ABSDE_{mv_pwd}$ 在四种算法中取得了最好的整体性能, 分别在 13 个和 14 个函数上取得最好结果. 为方便看出算法的整体性能对比, 图 11 中给出了采用 Friedman 检验得到的算法平均排名情况, 可看出 $ABSDE_{mv}$ 和 $ABSDE_{mv_pwd}$ 的排名最靠前, 且两者无明显差距. 鉴于 Parzen Window 方法的计算复杂度要明显高于本文采用方法, 我们继续采用欧氏距离作为试验个体的评估指标.

表 7 个体选择方式的有效性验证

Func.	$ABSDE_{mv_rnd}$	$ABSDE_{mv_max}$	$ABSDE_{mv_pwd}$	$ABSDE_{mv}$
F01	1.86E+00±4.52E-16=	5.92E+02±1.11E+02-	1.86E+00±4.52E-16=	1.86E+00±4.52E-16
F02	3.76E+06±1.08E+06-	7.60E+07±1.70E+07-	3.26E+06±8.58E+05-	2.04E+06±5.60E+05
F03	3.51E+08±1.84E+08-	7.09E+10±1.03E+10-	2.26E+08±1.55E+08-	1.36E+08±6.34E+07
F04	3.42E+04±7.45E+03-	9.58E+04±1.22E+04-	1.45E+04±3.76E+03-	1.19E+04±3.48E+03
F05	4.33E-01±1.13E-16=	2.79E+01±5.73E+00-	4.33E-01±1.13E-16=	4.33E-01±1.13E-16
F06	4.96E+01±1.22E+01=	3.81E+02±7.97E+01-	5.42E+01±1.79E+01=	4.68E+01±3.67E+00
F07	1.21E+02±2.54E+01-	1.30E+03±9.87E+02-	6.91E+01±1.35E+01=	6.89E+01±1.71E+01
F08	2.12E+01±6.06E-02-	2.12E+01±5.35E-02-	2.11E+01±9.23E-02=	2.11E+01±1.03E-01
F09	4.67E+01±7.70E+00-	6.25E+01±3.72E+00-	3.74E+01±4.20E+00=	3.72E+01±6.65E+00
F10	2.29E+00±4.67E-01+	8.12E+02±1.80E+02-	7.24E+00±2.27E+00-	4.43E+00±1.69E+00
F11	2.17E+01±6.83E+00-	1.07E+02±1.19E+01-	1.81E+01±5.30E+00+	2.77E+01±7.69E+00
F12	1.74E+02±8.86E+01=	5.43E+02±3.45E+01-	1.28E+02±2.25E+01=	1.32E+02±2.52E+01
F13	3.30E+02±6.17E+01-	5.62E+02±2.69E+01-	2.38E+02±5.70E+01=	2.39E+02±4.51E+01
F14	7.03E+02±2.32E+02+	1.10E+03±2.09E+02=	6.34E+02±2.22E+02+	1.01E+03±2.54E+02
F15	1.12E+04±2.47E+03-	1.38E+04±7.98E+02-	8.25E+03±7.72E+02=	8.24E+03±9.58E+02
F16	3.14E+00±8.99E-01-	3.61E+00±3.69E-01-	9.11E-01±3.00E-01+	1.14E+00±5.03E-01
F17	8.78E+01±4.58E+00-	3.15E+02±2.70E+01-	8.24E+01±5.23E+00=	8.27E+01±5.84E+00
F18	3.73E+02±6.39E+01-	6.81E+02±3.95E+01-	2.56E+02±2.87E+01=	2.63E+02±2.62E+01
F19	1.36E+01±2.33E+00-	9.01E+01±1.92E+01-	6.82E+00±1.17E+00=	6.73E+00±9.16E-01
F20	2.23E+01±4.77E-01-	2.36E+01±2.66E-01-	2.06E+01±9.94E-01=	2.09E+01±9.04E-01
F21	6.47E+02±4.12E+02=	8.86E+02±1.38E+02=	8.75E+02±3.50E+02-	8.06E+02±3.75E+02
F22	8.28E+02±2.58E+02=	4.03E+03±7.57E+02-	5.76E+02±2.18E+02=	9.78E+02±3.61E+02
F23	1.12E+04±2.65E+03-	1.39E+04±8.41E+02-	8.40E+03±9.11E+02=	8.32E+03±1.33E+03
F24	2.82E+02±2.16E+01=	3.75E+02±4.78E+00-	2.75E+02±1.46E+01=	2.80E+02±1.22E+01
F25	3.79E+02±1.23E+01-	3.89E+02±4.42E+00-	3.26E+02±1.91E+01=	3.37E+02±2.41E+01
F26	2.70E+02±1.01E+02+	2.24E+02±2.04E+01+	3.26E+02±9.13E+01=	3.64E+02±6.63E+01
F27	1.33E+03±2.06E+02-	2.00E+03±5.40E+01-	1.14E+03±1.16E+02=	1.20E+03±1.53E+02
F28	5.16E+02±5.92E+02-	1.25E+03±1.51E+02-	7.16E+02±9.40E+02=	4.08E+02±4.98E-01
+ / = / -	3/7/18	1/2/25	3/20/5	- / - / -

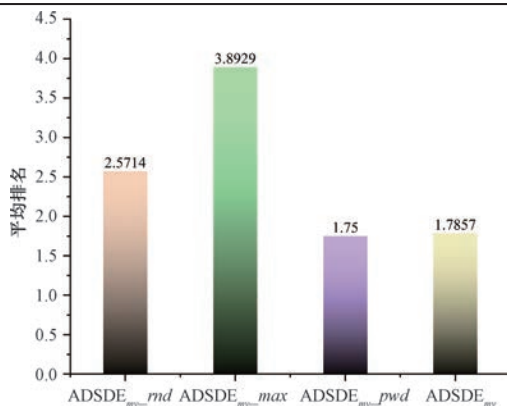


图 11 $ABSDE_{mv_rnd}$ 、 $ABSDE_{mv_max}$ 、 $MCDE_{mv_pwd}$ 和 $ABSDE_{mv}$ 的平均排名情况

5.4 与其他 EAs 性能对比

在本小节中, 将 $ABSDE_{mv}$ 与五种求解 MVOPs 的知名 EAs 进行性能对比, 以验证算法的竞争力. 这五种对比算法如下:

- (1) AEDA^[17]: 基于离散化方法的分布估计算法.
- (2) ACO_{mv} ^[21]: 基于松弛方法的蚁群优化算法.
- (3) DE_{mv} ^[18]: 基于二分方法的混合 DE 算法.
- (4) PSO_{mv} ^[14]: 基于二分方法的粒子群优化算法.

(5) MCDE_{mv}^[15]: 基于松弛方法的多策略协同 DE 算法.

上述算法中, AEDA 是基于离散化方法的算法, ACO_{mv} 和 MCDE_{mv} 是基于松弛方法的算法, DE_{mv} 和 PSO_{mv} 是基于二分方法的算法, 关于算法的其他特点已在 3.1 小节的“相关工作”中做了简介, 此处不再赘述. 为确保对比算法的结果准确,

AEDA、ACO_{mv}、DE_{mv} 以及 PSO_{mv} 的实验结果均直接源自 PSO_{mv} 的原始文献[14], MCDE_{mv} 的算法结果源自其原文献[15]. 本文算法在每个测试函数上独立运行 30 次, 将 30 次的平均值作为最终结果. 表 8 给出了这六种算法的实验结果, 表的最后一行统计了每种算法取得最好结果的函数个数.

表 8 与五种知名 EAs 的对比结果

Func.	AEDA	ACO _{mv}	DE _{mv}	PSO _{mv}	MCDE _{mv}	ABSDE _{mv}
F01	3.89E-02±9.83E-01	1.26E+02±2.13E+03	4.31E+03±3.62E+03	3.71E-11±4.63E-10	3.33E-02±1.80E-01	1.86E+00±4.52E-16
F02	8.60E+07±2.99E+08	2.79E+08±6.75E+08	1.64E+08±1.34E+08	3.39E+07±8.88E+07	3.80E+06±1.03E+06	2.04E+06±5.60E+05
F03	6.33E+08±2.36E+09	3.75E+10±1.07E+11	7.34E+10±7.92E+10	1.53E+09±6.39E+09	4.55E+08±2.96E+08	1.36E+08±6.34E+07
F04	4.80E+04±3.08E+04	6.80E+04±3.05E+04	9.18E+04±4.71E+04	1.00E+04±9.47E+03	2.38E+04±4.92E+03	1.19E+04±3.48E+03
F05	3.69E-01±2.33E+00	9.78E+00±7.66E+01	7.43E+02±4.51E+02	2.44E-05±1.73E-04	1.33E-01±3.40E-01	4.33E-01±1.13E-16
F06	2.10E+02±2.12E+02	4.43E+02±8.00E+02	6.73E+02±4.59E+02	2.87E+02±5.53E+02	4.78E+01±9.27E-01	4.68E+01±3.67E+00
F07	2.55E+01±3.21E+01	1.36E+02±1.63E+02	1.88E+02±1.18E+02	2.88E+01±2.88E+01	7.20E+01±1.61E+01	6.89E+01±1.71E+01
F08	2.12E+01±1.71E-01	2.12E+01±2.05E-01	2.12E+01±2.47E-01	2.12E+01±2.11E-01	2.12E+01±2.76E-02	2.11E+01±1.03E-01
F09	6.40E+01±1.11E+02	6.93E+01±3.21E+01	6.58E+01±9.55E+00	1.12E+01±2.20E+01	4.46E+01±7.75E+00	3.72E+01±6.65E+00
F10	2.77E+01±7.31E+01	4.04E+02±2.01E+03	1.46E+03±8.86E+02	2.08E-03±3.51E-02	2.98E+00±1.67E-01	4.43E+00±1.69E+00
F11	1.57E+02±1.11E+02	4.09E+02±4.40E+03	3.82E+02±1.15E+02	9.98E+01±2.63E+02	3.06E+01±9.52E+00	2.77E+01±7.69E+00
F12	3.70E+02±1.01E+02	5.31E+02±2.54E+02	6.23E+02±1.49E+02	3.55E+02±1.12E+02	1.90E+02±6.16E+01	1.32E+02±2.52E+01
F13	3.71E+02±7.72E+01	5.26E+02±2.56E+01	6.34E+02±1.61E+02	3.62E+02±7.56E+01	2.70E+02±4.95E+01	2.39E+02±4.51E+01
F14	6.47E+03±1.68E+03	1.35E+04±1.58E+03	6.24E+03±1.44E+03	1.81E+03±9.00E+03	8.67E+02±4.58E+02	1.01E+03±2.54E+02
F15	1.36E+04±2.04E+03	1.37E+04±2.28E+03	1.31E+04±1.94E+03	1.28E+04±2.08E+03	9.42E+03±1.77E+03	8.24E+03±9.58E+02
F16	3.73E+00±1.27E+00	3.92E+00±1.65E+00	3.75E+00±2.08E+00	3.68E+00±2.12E+00	3.42E+00±8.53E-01	1.14E+00±5.03E-01
F17	1.69E+02±7.76E+01	4.63E+02±7.83E+02	6.28E+02±2.23E+02	9.38E+01±3.11E+02	1.08E+02±1.14E+01	8.27E+01±5.84E+00
F18	1.76E+02±3.62E+02	4.51E+02±7.99E+02	6.40E+02±1.91E+02	7.54E+01±3.10E+02	3.04E+02±6.86E+01	2.63E+02±2.62E+01
F19	1.62E+01±1.15E+01	4.75E+03±2.63E+04	3.71E+02±6.69E+02	1.24E+01±3.15E+01	6.46E+00±1.20E+00	6.73E+00±9.16E-01
F20	2.16E+01±1.32E+00	2.16E+01±1.65E+00	2.18E+01±1.22E+00	1.92E+01±2.70E+00	2.16E+01±8.24E-01	2.09E+01±9.04E-01
F21	2.00E+02±1.71E-01	2.28E+02±4.03E+02	4.78E+02±1.84E+02	2.00E+02±4.41E-10	4.00E+02±5.55E-01	8.06E+02±3.75E+02
F22	6.61E+03±1.68E+03	1.36E+04±2.27E+03	6.39E+03±1.43E+03	1.85E+03±9.50E+03	8.92E+02±3.14E+02	9.78E+02±3.61E+02
F23	1.64E+04±9.06E+02	1.64E+04±1.16E+03	1.62E+04±1.50E+03	1.39E+04±3.29E+03	1.04E+04±1.98E+03	8.32E+03±1.33E+03
F24	1.43E+03±2.96E+02	1.68E+03±3.01E+02	1.74E+03±2.75E+02	1.25E+03±2.84E+02	2.72E+02±1.62E+01	2.80E+02±1.22E+01
F25	5.36E+02±6.83E+01	5.81E+02±6.25E+01	6.14E+02±5.44E+01	5.33E+02±5.34E+01	3.38E+02±1.24E+01	3.37E+02±2.41E+01
F26	7.70E+02±8.39E+02	9.54E+02±6.87E+03	3.05E+03±2.33E+03	2.90E+02±9.90E+02	3.92E+02±4.05E+01	3.64E+02±6.63E+01
F27	5.08E+03±5.83E+02	5.20E+03±1.36E+03	5.54E+03±2.19E+02	5.35E+03±5.97E+02	1.25E+03±2.07E+02	1.20E+03±1.53E+02
F28	1.15E+04±1.18E+03	1.13E+04±1.38E+03	1.12E+04±1.45E+03	1.06E+04±4.51E+03	2.14E+03±2.96E+02	4.08E+02±4.98E-01
Best	1	0	0	9	4	15

从表 8 中可看出, ABSDE_{mv} 在 28 个测试函数上的表现显著优于五种对比算法, 能在 15 个函数上取得最好结果. 具体而言, 与 AEDA 相比, ABSDE_{mv} 在 23 个函数上的结果要明显更优, 且仅在 5 个函数上略差. 值得注意的是, 与 ACO_{mv} 和 DE_{mv} 相比, ABSDE_{mv} 仅在测试函数 F21 上的性能更差, 但在剩余的其他函数上均明显好于它们. 与 PSO_{mv} 和 MCDE_{mv} 相比, ABSDE_{mv} 的竞争力有微弱下降, 但仍显著优于这两个对比算法; 其中, ABSDE_{mv} 分别在 18 和 15 个函数上明显优于 PSO_{mv} 和 MCDE_{mv}. 除此之外, 从函数类型方面来看, 不管是单峰函数, 还是多峰函数或者组合函数, ABSDE_{mv} 均表现出了强有力的

性能. 为简化整体对比情况, 根据 Friedman 检验结果, 在图 12 中给出了所涉及算法的平均排名. 可看出, 前三名分别是 ABSDE_{mv}、PSO_{mv} 以及 MCDE_{mv}, 并且 ABSDE_{mv} 的排名要比第二名好很多.

5.5 与其他 DE 性能对比

在本小节中, 为了进一步检验 ABSDE_{mv} 性能, 与三种知名的改进 DE 算法, 即 jSO^[37]、L-SHADE^[38] 和 CPI-DE^[31] 进行比较. 前两种算法 jSO 和 L-SHADE 分别为 CEC2017 和 CEC2014 单目标优化竞赛中的冠军算法; 而第三种 CPI-DE 是基于累计种群信息的 DE, 主要特点是在原坐标系和特征坐标系中同时实现了交叉操作, 并生成两个试验向量,

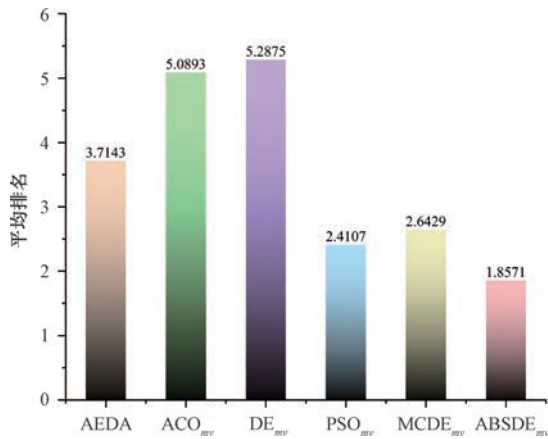


图 12 AEDA、ACO_{mv}、DE_{mv}、PSO_{mv}、MCDE_{mv} 和 ABSDE_{mv} 的平均排名情况

有非常好的性能.需说明的是,这三种 DE 是为求解 COPs 而设计的,为方便它们能求解 MVOPs,我们对其进行了适当改造,与本文算法 ABSDE_{mv} 一样分别对连续变量和离散变量进行优化,且同样采用了向下取整的方式.

为了保证实验的公平性,上述对比算法的公共参数设置与第 5.1 小节保持一致.表 9 给出了这四种算法的实验结果,且最好结果加粗显示. Wilcoxon's 秩和检验的结果汇总在表 9 的最后一行.从表 9 中可看出,本文算法 ABSDE_{mv} 在大多数问题上都能取得最好表现,在 21 个测试函数上取得了最好结果,且显著优于三种对比算法.与两个冠军算法 jSO

和 L-SHADE 相比,ABSDE_{mv} 分别在 23 个和 18 个函数上的结果明显更优.值得注意的是,jSO 是基于 L-SHADE 的改进版本,具有更强的单目标优化能力,但在求解 MVOPs 时性能弱于后者.与 ABSDE_{mv} 相比,CPI-DE 虽然同样采用特征坐标系松弛变量间相关性,但其求解的结果更差,仅在测数函数 F21 上优于 ABSDE_{mv}.为简化整体对比情况,根据 Friedman 检验结果,在图 13 中给出了所涉及算法的平均排名.可看出,前三名分别是 ABSDE_{mv}、L-SHADE 以及 jSO.从本小节实验可得出,在设计算法过程中,若不考虑 MVOPs 的问题特性,不做针对性的策略设计,算法仍无法取得较好结果.

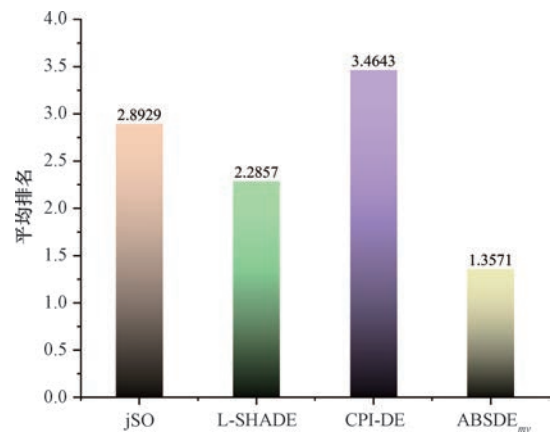


图 13 jSO、L-SHADE、CPI-DE 和 ABSDE_{mv} 的平均排名情况

表 9 与三种知名的改进 DE 算法的对比结果

Func.	jSO	L-SHADE	CPI-DE	ABSDE _{mv}
F01	6.82E+02±3.54E+02-	4.10E+01±1.39E+02-	6.10E+02±9.66E+01-	1.86E+00±4.52E-16
F02	1.63E+07±4.54E+06-	1.07E+07±4.43E+06-	1.90E+07±3.69E+06-	2.04E+06±5.60E+05
F03	1.75E+10±9.95E+09-	5.37E+09±3.69E+09-	3.59E+10±6.13E+09-	1.36E+08±6.34E+07
F04	2.73E+04±6.43E+03-	9.31E+04±4.78E+04-	2.93E+04±5.31E+03-	1.19E+04±3.48E+03
F05	1.47E+03±4.49E+02-	1.26E+02±1.23E+02-	7.76E+01±7.71E+00-	4.33E-01±1.13E-16
F06	2.15E+02±5.16E+01-	6.40E+01±2.53E+01=	1.36E+02±1.70E+01-	4.68E+01±3.67E+00
F07	1.06E+02±1.62E+01-	1.11E+02±2.11E+01-	1.36E+02±9.12E+00-	6.89E+01±1.71E+01
F08	2.12E+01±5.35E-02-	2.11E+01±8.98E-02=	2.12E+01±4.07E-02-	2.11E+01±1.03E-01
F09	5.71E+01±4.30E+00-	6.12E+01±3.29E+00-	7.45E+01±1.64E+00-	3.72E+01±6.65E+00
F10	4.09E+02±8.99E+01-	7.09E+01±6.86E+01-	1.35E+02±2.46E+01-	4.43E+00±1.69E+00
F11	1.27E+02±5.93E+01-	8.71E+00±6.93E+00+	4.25E+02±1.51E+01-	2.77E+01±7.69E+00
F12	2.78E+02±5.48E+01-	1.63E+02±3.66E+01-	4.52E+02±1.83E+01-	1.32E+02±2.52E+01
F13	3.97E+02±6.91E+01-	3.23E+02±5.77E+01-	4.50E+02±1.37E+01-	2.39E+02±4.51E+01
F14	2.61E+02±3.12E+02+	6.35E+01±3.65E+01+	1.15E+04±4.09E+02-	1.01E+03±2.54E+02
F15	8.20E+03±1.43E+03=	7.46E+03±8.45E+02+	1.46E+04±3.52E+02-	8.24E+03±9.58E+02
F16	3.77E+00±3.29E-01-	1.59E+00±3.97E-01-	3.78E+00±3.40E-01-	1.14E+00±5.03E-01
F17	1.34E+02±3.64E+01-	7.81E+01±1.03E+01+	5.57E+02±2.14E+01-	8.27E+01±5.84E+00
F18	6.47E+02±7.42E+01-	4.93E+02±1.14E+02-	5.56E+02±2.89E+01-	2.63E+02±2.62E+01
F19	5.88E+02±6.72E+02-	9.92E+00±1.03E+01=	8.41E+01±1.41E+01-	6.73E+00±9.16E-01
F20	2.21E+01±7.99E-01-	2.31E+01±1.40E+00-	2.29E+01±2.17E-01-	2.09E+01±9.04E-01
F21	1.08E+03±2.69E+02-	8.93E+02±2.96E+02=	7.20E+02±2.34E+02+	8.06E+02±3.75E+02
F22	3.54E+02±1.79E+02+	1.30E+02±7.27E+01+	1.20E+04±5.17E+02-	9.78E+02±3.61E+02

续表

Func.	jSO	L-SHADE	CPI-DE	ABSDE _{mv}
F23	8.41E+03±1.43E+03=	8.68E+03±1.33E+03=	1.48E+04±4.42E+02-	8.32E+03±1.33E+03
F24	3.42E+02±1.19E+01-	3.40E+02±1.39E+01-	3.69E+02±7.48E+00-	2.80E+02±1.22E+01
F25	3.73E+02±1.62E+01-	3.65E+02±1.30E+01-	4.25E+02±5.38E+00-	3.37E+02±2.41E+01
F26	3.14E+02±1.15E+02=	4.27E+02±6.34E+01-	4.12E+02±1.03E+02-	3.64E+02±6.63E+01
F27	1.62E+03±1.10E+02-	1.72E+03±1.59E+02-	2.20E+03±4.49E+01-	1.20E+03±1.53E+02
F28	1.59E+03±1.32E+03-	1.06E+03±1.32E+03-	5.87E+02±2.03E+01-	4.08E+02±4.98E-01
+/-/-	2/3/23	5/5/18	1/0/27	-/-/-

5.6 求解焊接梁设计问题

为进一步评估 ABSDE_{mv} 性能,在本小节中,我们使用 ABSDE_{mv} 来求解一个实际优化问题,即焊接梁设计问题(WBD)^[39],图 14 给出了该问题的一个示意图. WBD 问题包含一个梁(图中绿色部分)和焊接体(图中蓝色部分),梁由低碳钢(C-1010)制成,焊接在刚性支架上,并受到作用于自由端部的剪切载荷 P . 该问题主要包括四个决策变量:宽度 t 、梁的厚度 b 、焊缝厚度 h 以及焊接接头的长度 l ,以最小化梁的成本为设计目标. 需注意的是, h 和 l 为离散变量, t 和 b 为连续变量,且离散变量的值必须为 0.0065 的倍数. 表 10 给出了这一问题的详细定义. 为求解 WBD 问题,研究人员提出了多种方法^[21,40-42],

其中文献[21,40,42]是将离散变量 h 和 l 作为连续变量进行优化,而文献[39,41-44]是作为离散变量进行优化. 应指出的是,在文献[39,41-44]中,求解 WBD 问题取得了已知最好结果 1.7312.

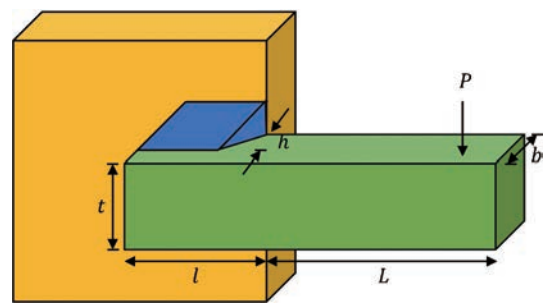


图 14 焊接梁设计问题示意图

表 10 焊接梁设计问题的具体定义

目标函数	$\text{Min } f(x) = 1.10471h^2l + 0.04811tb(14.0 + l)$
	$g_1 \quad \tau(x) - \tau_{max} \leq 0$
	$g_2 \quad \sigma(x) - \sigma_{max} \leq 0$
	$g_3 \quad h - b \leq 0$
约束条件	$g_4 \quad 0.10471h^2 + 0.04811tb(14.0 + l) - 5.0 \leq 0$
	$g_5 \quad 0.125 - h \leq 0$
	$g_6 \quad \delta(x) - \delta_{max} \leq 0$
	$g_7 \quad P - P_c(x) \leq 0$
	$\tau(x) = \sqrt{(\tau')^2 + (\tau'')^2} + 2\tau'\tau''\frac{1}{2R}, \tau'' = \frac{MR}{J},$
	$M = P\left(L + \frac{l}{2}\right), R = \sqrt{\frac{l^2}{4} + \left(\frac{h+t}{2}\right)^2}, J = 2\left\{\sqrt{2}hl\left[\frac{l^2}{12} + \left(\frac{h+t}{2}\right)^2\right]\right\},$
其他说明	$\sigma(x) = \frac{6PL}{t^2}, \delta(x) = \frac{4PL^3}{Et^3b}, P_c(x) = \frac{4.03E\sqrt{t^2b^6}}{L^2}\left(1 - \frac{t}{2L}\sqrt{\frac{E}{4G}}\right),$
	$P = 6000\text{lb}, L = 14\text{in}, E = 30 \times 10^6\text{psi}, G = 12 \times 10^6\text{psi},$
	$\tau_{max} = 13600\text{psi}, \sigma_{max} = 30000\text{psi}, \delta_{max} = 0.25\text{in},$
	$0.1 \leq h \leq 2, 0.1 \leq l \leq 10, 0.1 \leq t \leq 10, 0.1 \leq b \leq 2$

为充分验证 ABSDE_{mv} 在 WBD 问题上的性能,我们将上述文献中对应的算法 StrGA-PSOA^[39]、CLPSO^[43]、FA^[44]以及 MCDE_{mv}^[15]作为对比算法.

表 11 给出了所涉及五种算法的实验结果,其中对比算法的实验结果直接来自于原文献. 对比结果表明, ABSDE_{mv} 同样可得到已知的最好结果. 然

而,需说明的是,尽管其他对比算法也能获得最好结果,但 $ABSDE_{mv}$ 更加稳定且使用的评估次数最

少,这也表明 $ABSDE_{mv}$ 在该问题上本质上性能更好.

表 11 五种算法在 WBD 问题上的实验结果

算法	最佳值	平均值	最差值	标准差	评估次数(FEs)
StrGA-PSOA	1.731186	1.7622	—	0.0659	100000
CLPSO	1.731186	1.737459	—	0.017577	100000
FA	1.7312065	1.8786560	2.3455793	0.2677989	50000
MCDE _{mv}	1.7312	1.7312	1.7314	4.08E-05	20000
ABSDE _{mv}	1.7312	1.7312	1.7313	6.72E-08	20000

6 总 结

混合变量优化问题 MVOPs 是一类重要的工程问题,其难点在于需同时优化连续型和离散型决策变量.然而,连续变量与离散变量之间的相关性,是导致 MVOPs 较难求解的原因之一.在已知求解 MVOPs 的现有相关工作中,这种相关性往往被忽略了,从而导致算法搜索方向极易受到变量间相关性的影响.为此,本文提出了一种自适应双坐标系的差分进化算法($ABSDE_{mv}$)来求解 MVOPs,通过使用特征坐标系来松弛连续变量与连续变量、离散变量与离散变量以及连续变量与离散变量之间的相关性.同时,算法还保留原坐标系,设计了一种原坐标系和特征坐标系的自适应切换策略,以发挥出两种坐标系的各自优势,避免种群多样性丢失.此外,为进一步提高离散变量的优化效果,还专门设计了一种基于离散变量间相关性的局部搜索策略,有助于增强算法的整体性能.

在一套通用测试集和一个实际混合变量优化问题上进行了一系列的实验验证和分析,实验结果表明本文算法能有效提高 EAs 求解 MVOPs 的精度.与其他五种相关的知名算法相比,本文算法有很强的竞争力.在 WBD 问题上的实验结果表明,本文算法不仅在通用测试集上表现良好,在实际混合变量优化问题上同样也有很好表现.但是,必须指出的是,我们的工作仍存在两个缺点:其一,用于建立特征坐标系的协方差矩阵可反映任意两个决策变量之间的相关性,但无法反映多个决策变量之间的相关性;其二,在集成变异框架中,考虑到计算复杂度而未使用 Parzen Window 方法.因此,如何获取多个决策变量之间的相关性,以获得更可靠的特征坐标系,以及使用 Parzen Window 方法或更准确的个体选择方法,是未来值得进一步研究的问题.

此外,在其他类型的优化问题中,例如:多目标优化问题、多任务优化问题等,它们的决策变量间同样存在相关性的特点.因此,若能在问题求解过程中松弛变量间的相关性,这同样也能提升算法求解上述优化问题的性能.在未来的工作中,我们将本文方法拓展至更多类型的优化问题求解中,并设计相应的改进算法以求解更多的实际 MVOPs,例如:盘式制动器设计问题、CNN 超参数优化等.

参 考 文 献

- [1] Garg V, Deep K, Bansal S. Improved teaching learning algorithm with Laplacian operator for solving nonlinear engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 2023, 124: 106549
 - [2] Wang F, Li Y, Zhou A, et al. An estimation of distribution algorithm for mixed-variable newsvendor problems. *IEEE Transactions on Evolutionary Computation*, 2019, 24(3): 479-493
 - [3] Han B, Li B, Qin C. A novel hybrid particle swarm optimization with marine predators. *Swarm and Evolutionary Computation*, 2023, 83: 101375
 - [4] Roldán-Blay C, Escrivá-Escrivá G, Roldán-Porta C, et al. Optimal sizing and design of renewable power plants in rural microgrids using multi-objective particle swarm optimization and branch and bound methods. *Energy*, 2023, 284: 129318
 - [5] Vieira D A G, Lisboa A C. A cutting-plane method to nonsmooth multiobjective optimization problems. *European Journal of Operational Research*, 2019, 275(3): 822-829
 - [6] Subramanian A S R, Kannan R, Holtorf F, et al. Optimization under uncertainty of a hybrid waste tire and natural gas feedstock flexible polygeneration system using a decomposition algorithm. *Energy*, 2023, 284: 129222
 - [7] Chen Z G, Zhang Z H. Two-layer collaborative differential evolution algorithm for multimodal optimization problems. *Chinese Journal of Computers*, 2021, 44(9): 1806-1823 (in Chinese)
- (陈宗淦, 詹志辉. 面向多峰优化问题的双层协同差分进化算法. *计算机学报*, 2021, 44(9): 1806-1823)

- [8] Wang F, Zhang H, Han M, Xing L N. Co-evolution based mixed-variable multi-objective particle swarm optimization for UAV cooperative multi-task allocation problem. *Chinese Journal of Computers*, 2021, 44 (10): 1967-1983. (in Chinese)
(王峰, 张衡, 韩孟臣, 邢立宁. 基于协同进化的混合变量多目标粒子群优化算法求解无人机协同多任务分配问题. *计算机学报*, 2021, 44(10): 1967-1983)
- [9] Zhou X Y, Wu Z J, Wang M W. Artificial bee colony algorithm based on orthogonal experimental design. *Journal of Software*, 2015, 26(9): 2167-2190. (in Chinese)
(周新宇, 吴志健, 王明文. 基于正交实验设计的人工蜂群算法. *软件学报*, 2015, 26(9): 2167-2190)
- [10] Qu D P, Wang X W, Huang M. An aware ant routing algorithm in mobile peer-to-peer networks. *Chinese Journal of Computers*, 2013, 36(7): 1456-1464. (in Chinese)
(曲大鹏, 王兴伟, 黄敏. 移动对等网络中的感知蚁群路由算法. *计算机学报*, 2013, 36(7): 1456-1464)
- [11] Deb K, Goyal M. A flexible optimization procedure for mechanical component design based on genetic adaptive search. *Journal of Mechanical Design*, 1998, 120(2): 162-164
- [12] Ocenášek J, Schwarz J. Estimation of distribution algorithm for mixed continuous-discrete optimization problems//Proceedings of the 2nd Euro-International Symposium on Computational Intelligence. Kosice: IOS Press, 2002: 227-232
- [13] Guo C, Hu J, Ye B, et al. Swarm intelligence for mixed-variable design optimization. *Journal of Zhejiang University-SCIENCE A*, 2004, 5(7): 851-860
- [14] Wang F, Zhang H, Zhou A. A particle swarm optimization algorithm for mixed-variable optimization problems. *Swarm and Evolutionary Computation*, 2021, 60: 100808
- [15] Peng H, Han Y, Deng C, et al. Multi-strategy co-evolutionary differential evolution for mixed-variable optimization. *Knowledge-based Systems*, 2021, 229: 107366
- [16] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(4): 341-359
- [17] Shi W, Chen W N, Lin Y, et al. An adaptive estimation of distribution algorithm for multipolicy insurance investment planning. *IEEE Transactions on Evolutionary Computation*, 2017, 23(1): 1-14
- [18] Lin Y, Liu Y, Chen W N, et al. A hybrid differential evolution algorithm for mixed-variable optimization problems. *Information Sciences*, 2018, 466: 170-188
- [19] Liu J, Wang Y, Xin B, et al. A biobjective perspective for mixed-integer programming. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, 52(4): 2374-2385
- [20] Mashinchi M H, Orgun M A, Pedrycz W. Hybrid optimization with improved tabu search. *Applied Soft Computing*, 2011, 11(2): 1993-2006
- [21] Liao T, Socha K, de Oca M A M, et al. Ant colony optimization for mixed-variable optimization problems. *IEEE Transactions on Evolutionary Computation*, 2013, 18(4): 503-518
- [22] Zhang J, Sanderson A C. JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945-958
- [23] Das S, Abraham A, Chakraborty U K, et al. Differential evolution using a neighborhood based mutation operator. *IEEE Transactions on Evolutionary Computation*, 2009, 13(3): 526-553
- [24] Wang H, Rahnamayan S, Sun H, et al. Gaussian bare-bones differential evolution. *IEEE Transactions on Cybernetics*, 2013, 43(2): 634-647
- [25] Ld A, Cla B, Yi C, et al. Differential evolution with dynamic combination based mutation operator and two-level parameter adaptation strategy. *Expert Systems with Applications*, 2022, 192: 116298
- [26] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE transactions on Evolutionary Computation*, 2008, 13(2): 398-417
- [27] Mallipeddi R, Suganthan P N, Pan Q K, et al. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, 2011, 11(2): 1679-1696
- [28] Gong W, Cai Z, Ling C X, et al. Enhanced differential evolution with adaptive strategies for numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2010, 41(2): 397-413
- [29] Yi W, Chen Y, Pei Z, et al. Adaptive differential evolution with ensembling operators for continuous optimization problems. *Swarm and Evolutionary Computation*, 2022, 69: 100994
- [30] Guo S M, Yang C C. Enhancing differential evolution utilizing eigenvector-based crossover operator. *IEEE Transactions on Evolutionary Computation*, 2014, 19(1): 31-49
- [31] Wang Y, Liu Z Z, Li J, et al. Utilizing cumulative population distribution information in differential evolution. *Applied Soft Computing*, 2016, 48: 329-346
- [32] Das S, Suganthan P N. Differential evolution: A survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation*, 2010, 15(1): 4-31
- [33] Gong W, Fialho Á, Cai Z, et al. Adaptive strategy selection in differential evolution for numerical optimization: An empirical study. *Information Sciences*, 2011, 181(24): 5364-5386
- [34] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1): 55-66
- [35] Liang J J, Qu B Y, Suganthan P N, et al. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. *Computational Intelligence*

Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, 201212.34, 2013

- [36] Gao W, Wei Z, Luo Y, et al. Artificial bee colony algorithm based on Parzen window method. *Applied Soft Computing*, 2019, 74: 679-692
- [37] Brest J, Maucec M S, Bošković B. Single objective real-parameter optimization: Algorithm jSO//Proceedings of the 2017 IEEE congress on evolutionary computation (CEC). Donostia, Spain, 2017: 1311-1318
- [38] Tanabe R, Fukunaga A S. Improving the search performance of SHADE using linear population size reduction//Proceedings of the 2014 IEEE congress on evolutionary computation (CEC). Beijing, China, 2014: 1658-1665
- [39] Dimopoulos G G. Mixed-variable engineering optimization based on evolutionary and social metaphors. *Computer Methods in Applied Mechanics and Engineering*, 2007, 196(4-6): 803-817
- [40] Biswas R, Sharma D. A single-loop reliability-based design optimization using adaptive differential evolution. *Applied Soft Computing*, 2023, 132: 109907
- [41] Coello C A C, Montes E M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 2002, 16(3): 193-203
- [42] Wang L, Li L. An effective differential evolution with level comparison for constrained engineering design. *Structural and Multidisciplinary Optimization*, 2010, 41(6): 947-963
- [43] Gao L, Hailu A. Comprehensive learning particle swarm optimizer for constrained mixed-variable optimization problems. *International Journal of Computational Intelligence Systems*, 2010, 3(6): 832-842
- [44] Gandomi A H, Yang X S, Alavi A H. Mixed variable structural optimization using firefly algorithm. *Computers & Structures*, 2011, 89(23-24): 2325-2336



ZHOU Xin-Yu, Ph. D., associate professor. His main research interests include intelligent computing and machine learning.

HUANG Jun-Hong, Ph. D. candidate, His main research interests include intelligent optimization and edge

computing.

PENG Hu, Ph. D., associate professor. His main research interests include big data analysis, intelligent computing and application.

WANG Hui, Ph. D., professor. His main research interests include evolutionary computing, swarm intelligence, and optimal scheduling.

WANG Feng, Ph. D., professor. Her main research interests include data mining management and analysis, machine learning, and intelligent computing.

Background

Mixed-variable optimization problems (MVOPs) are not only those complex optimization problems with nonconvex, multimodal or even nondifferentiable, but also have the unique feature of containing both continuous and discrete variables, which exist in various fields, such as pressure vessel design problem, mixed-variable newsvendor problem and welded beam design problem. However, due to that the decision variables of MVOPs present different spatial distribution features, how to design an effective algorithm to solve MVOPs is still a challenge. The key issue in developing an efficient optimization algorithm lies in how to design efficient methods to deal with continuous and discrete variables separately and how to search the search space of mixed variables efficiently and cooperatively, so that the algorithm can quickly locate the global optimal solution.

To address those challenges, some researchers have made some initial attempts. But these related works still have some shortcomings. They ignore the correlation be-

tween continuous variables and discrete variables, which is one of the reasons why MVOPs are more difficult to solve. It will affect the search direction of the algorithm and reduce its effectiveness.

Therefore, this paper proposes a differential evolution algorithm based on adaptive bi-coordinate systems to solve MVOPs from the perspective of variable correlation. There are three contributions and innovations of this paper when compared with existing work. First, this paper establishes the eigen coordinate system using the covariance matrix information of the population to relax the correlation between continuous and discrete variables. Second, this paper designs an adaptive strategy to switch the eigen coordinate system and the original coordinate system for taking full advantages of the two coordinate systems. Third, to better optimize the discrete variables, this paper specially proposes a local search strategy based on the correlation between discrete variables, which is helpful for the overall performance of the algorithm.

Extensive experimental results have shown the great superiority of the proposed algorithm and methods over compared state-of-the-art algorithms.

This work is supported by the National Natural Science

Foundation of China under Grant Nos. 62366022, 61966019, 62166027 and 62173258, and the National Natural Science Foundation of Jiangxi Province under Grant Nos. 20232BAB 202048.