



决策变量分组优化的多目标萤火虫算法

邢文来, 吴润秀, 肖人彬, 钟劲文, 赵嘉

引用本文:

邢文来, 吴润秀, 肖人彬, 等. 决策变量分组优化的多目标萤火虫算法[J]. *智能系统学报*, 2025, 20(4): 838-857.
XING Wenlai, WU Runxiu, XIAO Renbin, et al. A multi-objective firefly algorithm with group optimization of decision variables[J]. *CAAI Transactions on Intelligent Systems*, 2025, 20(4): 838-857.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202406005>

您可能感兴趣的其他文章

融合分区和局部搜索的多模态多目标优化

Multimodal multi-objective optimization combining zoning and local search
智能系统学报. 2021, 16(4): 774-784 <https://dx.doi.org/10.11992/tis.202010026>

响应动态约束条件的多目标货位优化算法研究

Multi-objective location optimization algorithm in response to dynamic constraints
智能系统学报. 2020, 15(5): 925-933 <https://dx.doi.org/10.11992/tis.201906041>

布谷鸟搜索算法研究及其应用进展

Overview of the cuckoo search algorithm and its applications
智能系统学报. 2020, 15(3): 435-444 <https://dx.doi.org/10.11992/tis.201811005>

改进萤火虫优化算法的Renyi熵油污图像分割

Renyi entropy based on improved firefly optimization algorithm for image segmentation of waste oil
智能系统学报. 2020, 15(2): 367-373 <https://dx.doi.org/10.11992/tis.201809002>

缺失数据的混合式重建方法

Hybrid reconstruction method for missing data
智能系统学报. 2019, 14(5): 947-952 <https://dx.doi.org/10.11992/tis.201807037>

群智能算法优化支持向量机参数综述

Optimization of support vector machine parameters based on group intelligence algorithm
智能系统学报. 2018, 13(1): 70-84 <https://dx.doi.org/10.11992/tis.201707011>

DOI: 10.11992/tis.202406005

网络出版地址: <https://link.cnki.net/urlid/23.1538.tp.20250226.1541.007>

决策变量分组优化的多目标萤火虫算法

邢文来^{1,2,3}, 吴润秀^{1,2,3}, 肖人彬⁴, 钟劲文^{1,2,3}, 赵嘉^{1,2,3}

(1. 江西水利电力大学信息工程学院, 江西南昌 330099; 2. 江西省水利大数据智能处理与预警技术工程研究中心, 江西南昌, 330099; 3. 南昌市智慧城市物联感知与协同计算重点实验室, 江西南昌 330099; 4. 华中科技大学人工智能与自动化学院, 湖北武汉 430074)

摘要: 多目标萤火虫算法采用整体维度更新策略, 常因某几维变量上优化效果不佳, 导致算法收敛速度慢和寻优精度低。针对上述问题, 本文提出基于决策变量分组优化的多目标萤火虫算法 (multi-objective firefly algorithm with group optimization of decision variables, MOFA-GD)。引入决策变量分组机制, 根据各变量对算法性能的不同影响, 将整体决策变量划分成收敛性变量组和多样性变量组; 设计决策变量分组优化模型, 利用学习行为优化收敛性变量组, 加快种群收敛速度, 非均匀变异算子优化多样性变量组, 避免种群过早收敛, 逐渐减小的变异幅度引导种群局部开发, 提升算法寻优精度; 采用档案截断策略维护外部档案, 精准删除拥挤个体, 从而保持外部档案的多样性。实验结果表明: MOFA-GD 表现出优秀的收敛速度和寻优精度, 获得了均匀分布的 Pareto 解集。本文所提算法为求解多目标优化问题提供了一种高效且可靠的解决方案。

关键词: 多目标优化问题; 多目标萤火虫算法; 变量分组; 学习行为; 变异算子; 档案截断; 收敛速度; 寻优精度

中图分类号: TP18 **文献标志码:** A **文章编号:** 1673-4785(2025)04-0838-20

中文引用格式: 邢文来, 吴润秀, 肖人彬, 等. 决策变量分组优化的多目标萤火虫算法 [J]. 智能系统学报, 2025, 20(4): 838-857.

英文引用格式: XING Wenlai, WU Runxiu, XIAO Renbin, et al. A multi-objective firefly algorithm with group optimization of decision variables[J]. CAAI transactions on intelligent systems, 2025, 20(4): 838-857.

A multi-objective firefly algorithm with group optimization of decision variables

XING Wenlai^{1,2,3}, WU Runxiu^{1,2,3}, XIAO Renbin⁴, ZHONG Jinwen^{1,2,3}, ZHAO Jia^{1,2,3}

(1. School of Information Engineering, Jiangxi University of Water Resources and Electric Power, Nanchang 330099, China; 2. Jiangxi Province Engineering Research Center for Intelligent Processing and Early Warning Technology of Water Conservancy Big Data, Nanchang 330099, China; 3. Nanchang Key Laboratory of IoT Perception and Collaborative Computing for Smart City, Nanchang 330099, China; 4. School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: Multi-objective firefly algorithm adopts an overall dimension update strategy, which often results in slow convergence and poor optimization accuracy due to inadequate optimization effects on certain dimensions. To address these problems, this paper proposes a multi-objective firefly algorithm with group optimization of decision variables (MOFA-GD). Firstly, it introduces a decision variable grouping mechanism, dividing the entire set of decision variables into a convergence variable group and a diversity variable group based on different impacts of each variable on the algorithm's performance. Secondly, it designs a decision variable grouping optimization model, utilizing learning behavior to optimize the convergence variable group to accelerate the population's convergence speed, while using a non-uniform mutation operator to optimize the diversity variable group to prevent premature convergence. A gradually decreasing mutation amplitude guides local exploitation by the population, thereby enhancing the algorithm's optimization accuracy. Finally, it adopts an archive truncation strategy to maintain the external archive, accurately removing crowded individuals to preserve diversity of the external archive. Experimental results show that MOFA-GD demonstrates excellent convergence speed and optimization accuracy, achieving a uniformly distributed Pareto optimal solution set. The proposed algorithm provides a high-efficiency and reliable solution for solving multi-objective optimization problems.

Keywords: multi-objective optimization problems; multi-objective firefly algorithm; variable grouping; learning behaviour; mutation operator; archive truncation; convergence speed; optimization accuracy

收稿日期: 2024-06-03. 网络出版日期: 2025-02-26.

基金项目: 国家自然科学基金项目 (62466037); 南昌市重大科技攻关项目 (2024zdxm002, 2024zdxm010).

通信作者: 吴润秀. E-mail: wrx@nit.edu.cn.

在科学研究和工程实践的诸多领域中, 常常遇到需要同时优化多个目标的问题, 这类问题被称为多目标优化问题 (multi-objective optimization

problems, MOP)^[1]。在 MOP 中,各目标之间往往存在冲突,这意味着侧重任意一个目标的决策可能对其他目标产生负面影响。因此,在求解 MOP 时并非同时优化所有目标,而是在多个冲突的目标之间找到一种平衡,得到一个折衷的解集,即 Pareto 解集^[2]。随着科学技术的快速发展,机器学习^[3]、深度学习^[4]和演化计算等人工智能技术在诸多领域得到了广泛的应用,展示了其巨大的潜力和价值。多目标进化算法(multi-objective evolutionary algorithms, MOEA)^[5]是一类模拟生物进化的人工智能算法,具有种群搜索和信息交互等特点,能自动计算得到一个 Pareto 解集,为解决 MOP 提供了新思路。其中,群智能优化算法^[6]是一类模拟动物群体行为的元启发式算法,例如模拟灰狼群体狩猎行为的多目标灰狼算法(multi-objective grey wolf optimizer, MOGWO)^[7]、模拟蜂群采蜜行为的多目标人工蜂群算法(multi-objective artificial bee colony algorithm, MOABC)^[8]等。这类新型进化算法为解决复杂 MOP 拓宽了视野,丰富了 MOEA 在不同领域的应用。

2008年, Yang^[9]通过模拟萤火虫的群体行为,提出了萤火虫算法(firefly algorithm, FA)。萤火虫算法具有全局搜索能力强、行为简单易于实现和参数设置少等优点,使得萤火虫算法成为解决复杂优化问题的一种有力工具。2013年, Yang^[10]基于 FA 的优点将其拓展到求解多目标优化问题,提出了多目标萤火虫算法(multi-objective firefly algorithm, MOFA)。MOFA 继承了 FA 的优势,但同时也存在寻优精度不高、易过早收敛和全吸引模型导致不合理移动等缺陷。

为克服 MOFA 的缺陷,学者们从不同的角度进行研究,并提出相关改进算法。针对改进方式的不同,大致可归结为 3 类:1) 基于初始化种群的改进,这类算法通过改进种群初始化机制,提高初始种群的分布性,进而提升算法性能。谢承旺等^[11]提出一种混合型多目标萤火虫算法(hybrid multi-objective firefly algorithm, HMOFA),该算法提出混合水平正交表生成均匀分布初始化种群,为种群进化提供良好基础,同时融入三点最短路径技术提升算法性能。Zhao 等^[12]提出平衡收敛性与多样性的多策略集成萤火虫算法(multi-strategy ensemble firefly algorithm with equilibrium of convergence and diversity, MEFA-CD),该算法提出改进的线性同余法应用于种群初始化阶段,保证全局搜索能力,同时利用混合学习策略引导萤火虫学习,提高算法的收敛性能。赵嘉等^[13]提出

一种基于动态评分和邻域搜索的高维多目标萤火虫算法(many-objective firefly algorithm for solving large-scale sparse optimization problems, SMOFA),该算法采用双编码混合集成的方式生成稀疏的初始种群,并提出一种动态评分策略,为种群进化提供先验知识,有效保证解集的稀疏性。2) 基于更新机制的改进,这类算法改进萤火虫的位置更新公式,提高萤火虫的寻优能力。Lyu 等^[14]提出了一种基于补偿因子与精英学习的多目标萤火虫算法(multi-objective firefly algorithm based on compensation factor and elite learning, CFMOFA),该算法引入补偿因子改进位置更新公式,使个体在迭代中避免陷入局部最优并且快速靠近 Pareto 最优解。赵嘉等^[15]提出一种基于最大最小策略和非均匀变异的萤火虫算法(heterogeneous variation firefly algorithm with maximin strategy, HVFA-M),该算法引入最大最小策略区别萤火虫的优劣和实现外部档案的维护,同时采用精英解引导萤火虫移动改进学习公式,提升算法收敛速度和解集分布性。陈娟等^[16]提出基于动态反向学习和莱维飞行的双搜索模式萤火虫算法(double search mode firefly algorithm based on dynamic reverse learning and Levy flight, MOFA-LR),该算法利用反向解和莱维飞行分别对被支配和非支配个体的学习公式改进,提高了算法的搜索效率,避免算法早熟收敛。Zhao 等^[17]提出自适应区域划分的多目标萤火虫算法(multi-objective firefly algorithm with adaptive region division, MOFA-ARD),该算法对不同区域的萤火虫个体使用不同的位置更新机制,保证萤火虫个体之间进行优势互补,有效提高了算法的开发能力。3) 基于混合算法的改进,这类算法通常融入其他算法的思想,弥补萤火虫算法的不足,使算法具备更优的搜索性能。Wang 等^[18]提出一种面向大数据优化的混合多目标萤火虫算法(a hybrid multi-objective firefly algorithm for big data optimization, HMOFA),该算法将交叉策略嵌入到萤火虫算法,以保持种群多样性。Zhao 等^[19]提出分层学习多目标萤火虫算法(hierarchical learning multi-objective firefly algorithm for high-dimensional feature selection, HMOFA),该算法引入聚类初始化方法,减少了冗余特征,有效提升初始种群质量。

算法在求解 MOP 的过程中,决策变量扮演着至关重要的角色。决策变量将各目标函数相互联系,通过不断调整决策变量的值以寻找到满足约束条件的 Pareto 最优解集。尽管上述各改进算法

虽然对 MOFA 做出很好的改进,在一定程度上提升了算法的寻优能力,但是未充分考虑决策变量对算法性能的影响,整体维度更新的学习公式可能会导致某些维度的决策变量优化效果不佳,从而影响算法的收敛速度和寻优精度。为解决此类问题,本文提出决策变量分组优化的多目标萤火虫算法 (multi-objective firefly algorithm with group optimization of decision variables, MOFA-GD)。MOFA-GD 具有以下特点: 1) 加入决策变量分组机制^[20]。该机制基于变量对算法性能的不同影响,将整体决策变量划分成收敛性变量组和多样性变量组,为算法对决策变量分组优化奠定良好基础。2) 设计决策变量分组优化模型。此模型添加学习行为优化收敛性变量组,使萤火虫个体快速收敛到真实 Pareto 前沿,加快种群收敛速度;同时,添加非均匀变异算子^[21]优化多样性变量组,使算法前期克服种群约束,充分探索决策空间,后期注重局部探索,提升算法寻优精度。3) 采用档案截断策略^[22]。该策略实现对外部档案的动态调整,通过删除拥挤区域的萤火虫个体来维护种群的多样性,最终得到一组分布均匀的 Pareto 解集。

上述策略考虑决策变量对算法的影响,对变量组分配与之相适应的优化策略,有效改善算法的收敛性和多样性,提升了算法的收敛速度和收敛精度。

1 多目标优化相关理论

1.1 多目标优化问题

实际生活很多工程设计通常出现多个目标的优化问题。一般情况下,各目标之间互相制约,不能同时找到最优解。以最小化问题为例, MOP 可以表达为

$$\begin{cases} \min \mathbf{y} = F(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \cdots \ f_m(\mathbf{x})] \\ \mathbf{x} = [x^1 \ x^2 \ \cdots \ x^n] \in \mathbf{X} \subset \mathbf{R}^n \\ \mathbf{y} = [y^1 \ y^2 \ \cdots \ y^m] \in \mathbf{Y} \subset \mathbf{R}^m \end{cases}$$

式中: \mathbf{x} 为决策向量, n 为决策向量的维度, \mathbf{X} 为 n 维决策空间, \mathbf{y} 为目标向量, m 为目标函数的个数, \mathbf{Y} 为 m 维目标空间。通常 MOP 不存在唯一的最优解,取而代之的是一组折衷最优解。这组折衷最优解在决策空间被称为帕累托集合 (Pareto set, PS), 在目标空间称为帕累托前沿 (Pareto front, PF)。若 \mathbf{x}_i 和 \mathbf{x}_j 为决策空间 \mathbf{X} 的 2 个决策向量, 当且仅当对于所有的 $l = 1, 2, \dots, m$ 都有 $f_l(\mathbf{x}_i) \leq f_l(\mathbf{x}_j)$, 且存在一个 $h = 1, 2, \dots, m$ 使得 $f_h(\mathbf{x}_i) < f_h(\mathbf{x}_j)$, 称 \mathbf{x}_i 帕累托支配 \mathbf{x}_j , 记 $\mathbf{x}_i < \mathbf{x}_j$ 。如果不存在任何一个决策向量 $\mathbf{x} \in \mathbf{X}$ 能够使得 $\mathbf{x} < \mathbf{x}^*$, 则 \mathbf{x}^* 是一个帕累

托最优解 (Pareto optimal solution)。

1.2 多目标萤火虫算法

萤火虫算法中的发光机制和行为方式比较独特,萤火虫在决策空间中被描述为一系列的点,每一个点代表一个解。解映射在目标空间的适应值就代表萤火虫的亮度,也代表萤火虫所处位置的优劣程度。萤火虫依靠彼此之间的亮度相互吸引,通过不断的相互吸引更新位置和亮度,从而完成寻优任务。吸引力大小由萤火虫的亮度和彼此之间的距离 i 决定。萤火虫 j 对萤火虫 i 的吸引力 β_{ij} 可定义为^[23]

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2}$$

式中: β_0 为最大吸引力,通常 β_0 取 1; r_{ij} 为萤火虫 i 到萤火虫 j 的欧氏距离; γ 为光吸收系数,控制着 2 个萤火虫个体之间的距离 r_{ij} 对彼此之间吸引力 β_{ij} 的影响,一般取 $\gamma \in [0.1, 100]$ 。

求解多目标优化问题时,通常采用萤火虫个体之间的 Pareto 支配关系确定它们之间的吸引关系。对于决策空间中的 2 只萤火虫,若萤火虫 i 被萤火虫 j 支配,此时萤火虫 j 比萤火虫 i 更亮,则萤火虫 i 被萤火虫 j 吸引进行移动,萤火虫 i 的位置更新公式为

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \beta_{ij}(\mathbf{x}_j(t) - \mathbf{x}_i(t)) + \alpha \cdot \boldsymbol{\varepsilon}_i(t) \quad (1)$$

式中: t 为当前迭代次数; \mathbf{x}_i 和 \mathbf{x}_j 分别为萤火虫 i 和萤火虫 j 所处的决策空间位置; β_{ij} 为萤火虫 i 和萤火虫 j 之间的吸引力; $\boldsymbol{\varepsilon}_i(t)$ 通常为均匀分布、正态分布或其他分布得到的随机数向量; α 为步长因子,控制随机数向量 $\boldsymbol{\varepsilon}_i(t)$ 的幅度,一般取 $\alpha \in [0, 1]$ 。若萤火虫 i 未找到被吸引的萤火虫 j , 此时萤火虫 i 的位置更新公式为^[24]

$$\mathbf{x}_i(t+1) = \mathbf{g}^*(t) + \alpha \cdot \boldsymbol{\varepsilon}_i(t) \quad (2)$$

式中 $\mathbf{g}^*(t)$ 为多个目标函数以随机加权求和的方式取得的当前最优解。

2 决策变量分组优化的多目标萤火虫算法

2.1 决策变量分组机制

2.1.1 决策变量分组

在多目标优化问题中,决策变量与目标函数值之间存在着映射关系,决策变量值的变化直接影响目标函数值大小。在多目标进化算法中,个体按照预定的规则,在迭代过程中不断调整其决策变量,从而实现多个目标的同时优化。为了探究各个维度的变量对目标函数值的具体影响,本文采用控制变量法,通过持续改变个体的一维变量值,且保持其余变量值不变,产生新的个体,观

察相应个体的 Pareto 支配关系, 进而归纳出不同变量的作用机制与特性。

若改变个体 \mathbf{x} 的其中一维变量一次, 生成了一个新个体 \mathbf{x}^* , 则它们之间的关系可能有以下 2 种情况: 1) \mathbf{x} 与 \mathbf{x}^* 互不支配, 即个体 \mathbf{x} 和 \mathbf{x}^* 之间不存在支配关系。2) $\mathbf{x} < \mathbf{x}^*$ 或 $\mathbf{x}^* < \mathbf{x}$, 即个体 \mathbf{x} 和 \mathbf{x}^* 之间存在支配关系。由此可以推断, 持续改变某一维度的变量产生的个体集合会呈现以下 3 种情况。为了区分这些情况, 特此定义如下: 1) 位置变量, 当持续改变位置变量时, 所产生的个体集合中, 任意 2 个个体之间不存在支配关系。2) 距离变量, 当持续改变距离变量时, 所产生的个体集合中, 任意 2 个个体之间存在支配的关系。3) 混合变量, 当持续改变混合变量时, 所产生的个体集合中, 既包含存在支配关系的个体对, 也包含不存在支配关系的个体对。综上所述: 距离变量在优化过程中, 其变化驱动种群向 Pareto 前沿靠近, 影响着种群的收敛性, 因此将其划分为收敛性变量组; 位置变量的变化有助于保持种群的多样性, 防止算法过早收敛到 Pareto 前沿局部区域, 因此将其划分为多样性变量组; 混合变量虽然对种群的收敛性和分布性均有影响, 但是在处理具有多个局部最优解的 MOP 中, 找到更广泛分布的 Pareto 最优解集更为关键, 因此也将其划分为多样性变量组。

2.1.2 决策变量检测

为了检测决策变量对算法性能的作用机制,

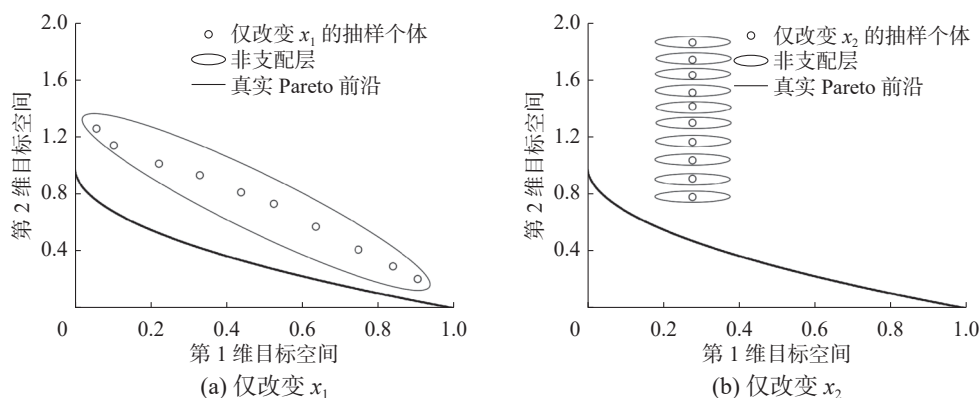


图 1 在 ZDT1 测试函数中仅改变 x_1 和仅改变 x_2 的采样解在目标空间划分非支配层

Fig. 1 Sampling solutions that change only x_1 and only x_2 in the ZDT1 test function map to the non-dominated layer of the target space

2.2 决策变量分组优化模型

在 2.1 节将萤火虫个体决策向量划分成收敛性变量组和多样性变量组; 在本节中提出决策变量分组优化模型, 对收敛性变量组和多样性变量组分配适应其特性的策略以一种动态递减的概率

对每个维度的变量进行随机扰动, 并进行固定次数的采样, 利用快速非支配排序, 确定采样解在目标空间的最大非支配层, 以确定各采样解之间的支配关系。具体地, 令采样频次 N_{sample} 为采样扰动解的数量, x_i 为 MOP 的决策向量 $\mathbf{x}(\mathbf{x} = [x_1 \ x_2 \ \dots \ x_d])$ 的任意一维变量。在 x_i 的取值范围内随机扰动且保持其他变量不变, 进行 N_{sample} 次随机采样, 观察采样解映射到目标空间中的非支配分层情况, 若最大非支配层数 $R_{\text{max}} = 1$, 则变量 x_i 是位置变量; 若最大非支配层数 $R_{\text{max}} = N_{\text{sample}}$, 则变量 x_i 是距离变量; 若最大非支配层数 $R_{\text{max}} > 1$ 且 $R_{\text{max}} < N_{\text{sample}}$, 则变量 x_i 是混合变量。其中, 采样频次 N_{sample} 越大, 检测结果越准确, 但同时消耗的计算资源越大。为平衡算法的计算复杂度和实验结果稳定性, 通过反复实验对比, 采样频次 N_{sample} 设置为 10 已能满足检测决策变量的需求, 为进一步确保结果可靠性, 将采样频次 N_{sample} 设置为 20 更为合适。

以 ZDT1 测试函数^[14] 为例, 图 1 给出了仅对变量 x_1 和 x_2 在其可行域内随机采样 $N_{\text{sample}} = 10$ 次后, 采样解在目标空间中的非支配分层情况。其中空心圆点代表采样解映射到目标空间中的位置, 黑色的线代表 ZDT1 的真实 Pareto 前沿, 椭圆代表非支配层。图 1(a) 中仅改变 x_1 采样解最大非支配层 $R_{\text{max}} = 1$, 根据前文的规则, x_1 是位置变量, 应归类到多样性变量; 图 1(b) 中仅改变 x_2 采样解最大非支配层 $R_{\text{max}} = N_{\text{sample}}$, 根据前文的规则, x_2 变量是距离变量, 应划分到收敛性变量组。

p_m 进行优化, 以提高算法的收敛性和多样性。在萤火虫相互吸引位置更新后, 添加向优秀个体学习的行为, 优化收敛性变量组, 使萤火虫个体快速收敛到真实 Pareto 前沿, 加快种群收敛速度; 添加非均匀变异算子, 优化多样性变量组, 使算法

避免早熟收敛,增加种群多样性。分组优化概率 p_m 是一个随迭代次数增加动态递减的因子,其表达式为

$$p_m = 1 - \exp(t/t_{\max} - 1)$$

式中: t 为当前迭代次数, t_{\max} 为最大迭代次数, e 为自然常数。算法前期较大的 p_m 有利于发挥决策变量分组优化模型的寻优能力;算法后期 p_m 逐渐减小,防止偏离种群最优进化方向。

2.2.1 学习行为优化收敛性变量组

收敛性变量组中的距离变量能驱动个体向真实 Pareto 前沿收敛。外部档案中的精英解具有非劣特性,而多个目标加权求和得出的当前最优解更具综合性,因此这些优秀个体更接近全局最优解。鉴于此,将精英解和当前最优解共同引导萤火虫个体学习,优化萤火虫个体的收敛性变量组,从而提升算法的收敛速度,同时保留个体多样性变量组的信息,避免大量个体向优秀个体学习出现的聚集现象。这里的精英解是从外部档案中随机选取一个萤火虫个体。对每一代萤火虫 i 向优秀个体学习,优化收敛性变量组的操作可定义为

$$x_{i,Q_c}(t+1) = \omega_1 \times d_{Q_c}^*(t) + \omega_2 \times g_{Q_c}^* \quad (3)$$

式中: x_{i,Q_c} 、 $d_{Q_c}^*$ 和 $g_{Q_c}^*$ 分别为萤火虫 i 、精英解 d^* 和当前最优解 g^* 的收敛性变量组, $\omega_1, \omega_2 \in [0, 1]$ 且 $\omega_1 + \omega_2 = 1$ 。

2.2.2 变异算子优化多样性变量组

非均匀变异算子在算法前期变异幅度较大,能够使算法克服种群约束,充分开发解空间,提升算法的勘探能力;随着算法的迭代变异幅度逐渐减小,能够进行更细致的搜索,提高算法的寻优精度。利用非均匀变异算子优化多样性变量组,有效防止种群聚集,提升种群多样性。因此,每一代的萤火虫 i 进行变异,优化多样性变量组的操作定义为^[14]

$$x_i^{k'} = \begin{cases} x_i^k + \Delta(t, U^k - x_i^k), & \theta < 0.5 \\ x_i^k + \Delta(t, x_i^k - L^k), & \theta \geq 0.5 \end{cases} \quad (4)$$

式中: x_i^k 为萤火虫个体 i 多样性变量组 Q_d 中随机选取的第 k 维变量, θ 为区间 $[0, 1]$ 一个随机数, t 为算法的当前迭代次数, U^k 和 L^k 分别为第 k 维变量的上界和下界。 $\Delta(t, y)$ 可表示为

$$\Delta(t, y) = y \times (1 - r^{(1-t/t_{\max})^b})$$

式中: r 为 $[0, 1]$ 的随机数; t_{\max} 为最大迭代次数; b 为决定变异幅度的参数,一般取值为 2~5, 本文取 $b = 3$ 。变异幅度随着迭代次数 t 的增加而逐渐减小,使该算子注重局部搜索,有利于产生更靠近真实 Pareto 前沿的新解。

2.3 档案截断策略

在 MOEA 中,为防止精英解在种群进化过程中丢失,通常会设立一个容量固定的外部档案来保存算法运行中的精英解。此外,外部档案的个体提供了全局多样化的信息,帮助算法做更全面的决策。外部档案中的个体数量对算法的性能和结果有一定影响,如果外部档案个体数量过少,则会导致解的多样性不足;如果外部档案个体数量过多,会导致算法的运行速度变慢,则增加计算复杂度。随着算法迭代的进行,不断有新的非支配解进入外部档案,为保证算法性能,当外部档案个体数量超过其档案最大容量时,采用 Zitzler 等^[22] 提出的档案截断策略实现对外部档案维护。

算法进行一轮迭代后,外部档案加入新的非支配解,若此时外部档案 P_t 的当前个体数量 $|P_t|$ 大于外部档案最大容量 N_{elite} ,调用档案截断策略迭代地从外部档案 P_t 中移除个体,直到外部档案 P_t 的当前个体数量 $|P_t|$ 等于最大容量 N_{elite} 。在每次的迭代中,选择萤火虫个体 i 满足对于所有的 $j \in P_t$ 且 $i \leq_d j$ 就将萤火虫个体 i 移除,其中 $i \leq_d j$ 的定义为^[22]

$$\forall 0 < k < |P_t| : \sigma_i^k = \sigma_j^k \vee \exists 0 < k < |P_t| : \left[(\forall 0 < l < k : \sigma_i^l = \sigma_j^l) \wedge \sigma_i^k < \sigma_j^k \right] \quad (5)$$

式中 σ_i^k 为个体 i 到外部档案 P_t 第 k 近邻距离。

2.4 算法流程

输入 MOP 表达式 $F(x)$, 决策向量维度 D , 决策空间上界 U 和下界 L , 光吸收系数 γ , 最大吸引力 β_0 , 初始步长因子 α , 采样频次 N_{sample} , 初始种群大小 N , 最大迭代次数 t_{\max} 和外部档案最大容量 N_{elite} 。

输出 Pareto 最优解集。

- 1) 初始化,设置输入参数。
- 2) 在可行域内随机生成一个决策向量 x , 进行决策变量分组。
- 3) 若 $l \leq D$, 转至步骤 4); 否则,获得多样性变量组 Q_d 和收敛性变量组 Q_c , 转至步骤 7)。
- 4) 仅改变的第 l 维变量采样 N_{sample} 次,获得采样解集合 P_x , 计算采样解适应值。
- 5) 对采样解进行快速非支配排序,判断最大非支配层 R_{\max} 的大小。若 $R_{\max} = 1$, 则第 l 维变量为位置变量;若 $R_{\max} = N_{\text{sample}}$, 则第 l 维变量为距离变量;若 $1 < R_{\max} < N_{\text{sample}}$, 则第 l 维变量为混合变量。将位置变量和混合变量划分到多样性变量组 Q_d , 将距离变量划分到收敛性变量组 Q_c 。
- 6) $l = l + 1$, 转至步骤 3)。
- 7) 种群初始化,计算种群在每一个目标函数

上的适应值

8) 若 $t \leq t_{\max}$, 转至步骤 9); 否则, 转至步骤 19)。

9) 若 $i \leq N$, 转至步骤 10); 否则, 转至步骤 17)。

10) 若 $j \leq N$, 转至步骤 11); 否则, 转至步骤 13)。

11) 若萤火虫 j 支配萤火虫 i , 按照式 (1) 更新其位置; 若萤火虫 i 不受任何个体支配, 按照式 (2) 更新其位置。检查萤火虫位置是否超出可行域范围, 若超出可行域范围则赋予其可行域边界值, 计算其适应值。

12) $f = f + 1$ 。转至步骤 10)。

13) 若 $\text{rand}(1) < p_m$, 萤火虫 i 进行分组优化, 转至 14); 否则, 转至步骤 16)。

14) 萤火虫 i 按照式 (3) 进行向优秀个体学习优化收敛性变量组 Q_c ; 按照式 (4) 进行非均匀变异优化多样性变量组 Q_d 。

15) 计算萤火虫 i 分组优化后的适应值。若分组优化后的萤火虫 i^* 支配优化前的萤火虫 i , 则分组优化后萤火虫 i^* 的位置和适应值; 否则保留原来萤火虫 i 的位置和适应值。

16) $i = i + 1$, 转至步骤 9)。

17) 外部档案的更新, 若外部档案中个体数量大于最大容量 N_{elite} , 则按照满足式 (5) 的个体删除。

18) $t = t + 1$, 转至步骤 8)。

19) 迭代结束, 输出 Pareto 最优解集。

2.5 算法时间复杂度分析

设种群规模为 N , 外部档案最大容量 N_{elite} , 决策向量维度为 D , 目标空间的维度为 m , 采样频次

N_{sample} 。MOFA 中萤火虫彼此相互吸引, 需要 2 个循环嵌套遍历种群, 所以 MOFA 的时间复杂度为 $O(D \times N^2)$ 。MOFA-GD 基于 MOFA 添加决策变量分组机制, 分析了 D 个维度变量, 每个维度采样频次为 N_{sample} , 因此时间复杂度是 $O(N_{\text{sample}} \times D)$ 。决策变量分组优化模型的时间复杂度不大于 $O(D)$ 。获取非支配解需要对比 N 个萤火虫在 m 个目标空间的适应值, 因此获取非支配解的时间复杂度为 $O(m \times N \log N)$ 。采用档案截断策略需要计算一个距离关系矩阵, 外部档案最大容量为 N_{elite} , 所以截断档案策略的时间复杂度为 $O(N_{\text{elite}} \log N_{\text{elite}})$ 。综上, MOFA-GD 时间复杂度为 $O(N_{\text{sample}} \times D) + O(D \times N^2) + O(m \times N \log N) + O(N_{\text{elite}} \log N_{\text{elite}})$ 。由于求解的目标个数 m 和采样频次 N_{sample} 远小于种群规模 N , 外部档案最大容量 N_{elite} 通常与种群规模 N 设置相同, 因此 MOFA-GD 的时间复杂度为 $O(D \times N^2)$, 与 MOFA 时间复杂度在同一数量级上。

3 实验仿真与结果分析

3.1 测试函数

为了测试 MOFA-GD 的有效性, 本文将 MOFA-GD 在 ZDT、Viennet 和 DTLZ 系列共 15 个测试函数上进行验证^[14]。15 个测试函数中由 5 个 2 目标函数和 10 个 3 目标函数组成, 表达式和真实 Pareto 前沿数据集采取 PLATEMO 平台^[25] 的数据。15 个测试函数的特性如表 1。各个测试函数通过决策变量分组机制, 划分决策变量和分组情况见表 2。

表 1 测试函数集
Table 1 Set of test functions

测试函数	目标个数	约束	Pareto 前沿特征
ZDT1	2	$n = 30, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	凸
ZDT2	2	$n = 30, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	非凸
ZDT3	2	$n = 30, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	非连续
ZDT4	2	$n = 10, 0 \leq x_1 \leq 1, -5 \leq x_i \leq 5, i = 2, 3, \dots, n$	凸+多模态
ZDT6	2	$n = 10, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	非凸+多模态+有偏好
Viennet1	3	$n = 2, -2 \leq x_i \leq 2, i = 1, 2$	凸
Viennet2	3	$n = 2, -4 \leq x_i \leq 4, i = 1, 2$	混合
Viennet3	3	$n = 2, -3 \leq x_i \leq 3, i = 1, 2$	混合+退化
DTLZ1	3	$n = 7, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	多模态
DTLZ2	3	$n = 12, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	非凸
DTLZ3	3	$n = 12, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	非凸+多模态
DTLZ4	3	$n = 12, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	非凸+有偏好

续表 1

测试函数	目标个数	约束	Pareto 前沿特征
DTLZ5	3	$n = 12, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	非凸+退化
DTLZ6	3	$n = 12, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	非凸+退化+有偏好
DTLZ7	3	$n = 22, 0 \leq x_i \leq 1, i = 1, 2, \dots, n$	混合+不连续+多模态

表 2 各个测试函数的决策变量属性和分组

Table 2 Each test function of the decision variables properties and grouping

测试函数	多样性变量组		收敛性变量组
	位置变量	混合变量	距离变量
ZDT1、ZDT2、 ZDT3	x_1	—	x_2, x_3, \dots, x_{30}
ZDT4、ZDT6	x_1	—	x_2, x_3, \dots, x_{10}
Viennet1、Viennet2、Viennet3	—	x_1, x_2	—
DTLZ1	x_1, x_2	—	x_3, x_4, \dots, x_7
DTLZ2、DTLZ3、DTLZ5、DTLZ6	x_1, x_2	—	x_3, x_4, \dots, x_{12}
DTLZ4	—	x_1, x_2	x_3, x_4, \dots, x_{12}
DTLZ7	—	x_1, x_2	x_3, x_4, \dots, x_{12}

3.2 评价指标

在求解 MOP 时, 算法应注重 Pareto 解集逼近真实 Pareto 最优解集的能力, 同时也应注重其在真实 Pareto 最优解集上的分布情况。为了准确地评价 MOFA-GD 的性能, 本文采用世代距离 (generational distance, GD)^[10] 和最大覆盖范围 (maximum spread, MS)^[15] 2 个指标来分别评估其收敛性和多样性。

GD 的表达式为

$$\mathcal{L}_{GD} = \frac{1}{N} \sqrt{\sum_{j=1}^N (\mathbf{V}_j - \mathbf{O}_j^*)^2} \quad (6)$$

式中: N 为算法所得 Pareto 解的个数, \mathbf{V}_j 为第 j 个 Pareto 解的目标值, \mathbf{O}_j^* 为距离 \mathbf{V}_j 最近的真实 Pareto 解的目标值。从式 (6) 可知, \mathcal{L}_{GD} 评估了算法所得 Pareto 前沿与真实 Pareto 前沿之间的最小距离的平均值, \mathcal{L}_{GD} 值越小, 则算法所得 Pareto 前沿在真实 Pareto 前沿的拟合程度越好。

MS 的表达式为

$$\mathcal{L}_{MS} = \sqrt{\frac{1}{M} \sum_{l=1}^M \left(\frac{\min(f_l^{\max}, F_l^{\max}) - \max(f_l^{\min}, F_l^{\min})}{F_l^{\max} - F_l^{\min}} \right)^2} \quad (7)$$

式中: M 为目标数, f_l^{\max} 和 f_l^{\min} 分别为算法所得 Pareto 前沿第 l 个目标最大值和最小值, F_l^{\max} 和 F_l^{\min} 分别为真实 Pareto 前沿第 l 个目标最大值和最小值。从式 (7) 可以看出 MS 评估了算法所得

Pareto 前沿在真实 Pareto 前沿上的覆盖率, MS 值越大, 则算法所得 Pareto 前沿上覆盖范围越广。

但是 GD 和 MS 不能反映其在真实 Pareto 前沿上的分布情况, 针对此, 本文采用一种综合性能指标: 反世代距离 (inverted generational distance, IGD)^[16] 评估算法所得 Pareto 前沿在真实 Pareto 前沿上得的分布情况。

IGD 的表达式为

$$\mathcal{L}_{IGD} = \frac{1}{N^*} \sqrt{\sum_{j=1}^{N^*} (\mathbf{O}_j - \mathbf{V}_j^*)^2} \quad (8)$$

式中: N^* 为真实 Pareto 解的个数, \mathbf{O}_j 为第 j 个真实 Pareto 解的目标值, \mathbf{V}_j^* 为距离 \mathbf{O}_j 最近的算法所得 Pareto 解的目标值。从式 (8) 可知, IGD 计算每个在真实 Pareto 前沿到算法所得 Pareto 前沿之间的最小距离和, 当 IGD 值越小, 说明种群分布越均匀且越广泛, 同时也能较好地反映算法所得的 Pareto 前沿的收敛程度。

3.3 MOFA-GD 与经典多目标优化算法对比实验

为了验证 MOFA-GD 的性能, 将 MOFA-GD 选用 6 种在多目标优化领域出现较早、广泛应用且在理论和实践中表现出良好性能的经典优化算法, 进行对比实验, 6 种算法分别为 MOPSO-CD^[26]、SPEA2^[22]、MOEA/D^[27]、NSGA-II^[28]、PESA-II^[29] 和 MOFA^[10], 所有算法的实验参数设置与相应参考文献保持一致, 各算法参数设置如表 3 所示。

表 3 各算法的实验参数设置

Table 3 Experimental parameter settings for each algorithms

算法	参数设置	年份
MOPSO-CD	$\omega = 0.4, R_1, R_2 = \text{rand}[0, 1]$	2005
SPEA2	$p_c = 0.9, p_m = 1/n, \eta_c = 20, \eta_m = 20$	2002
MOEA/D	$\gamma = 0.5$	2007
NSGA-II	$p_c = 0.9, p_m = 1/n, \eta_c = 20, \eta_m = 20$	2002
PESA-II	$p_c = 0.7, p_m = 1/n, \eta_c = 20, \eta_m = 20$	2001
MOFA	$\alpha = 0.2, \beta_0 = 1, \gamma = 1$	2013
MOFA-GD	$\alpha = 0.2, \beta_0 = 1, \gamma = 1, N_{\text{sample}} = 20$	—

对于所有的测试函数, 各个算法的种群规模 N 均设置为 100, 外部档案最大容量 N_{elite} 设置为 100。为确保实验的公平性和随机性, 所有算法均在相同实验环境下进行仿真, 每个测试函数评估 10 000 次, 各算法在同一测试函数上独立运行 50 次。

表 4~6 给出了 MOFA-GD 与 6 种经典算法在 15 个测试函数上的 GD、MS、IGD 和运行时间的均值 (mean) 和标准差 (std), 给出了各算法的优势解统计 (total) 和基于 Friedman 检验的秩平均值 (ranking)。

表 4 MOFA-GD 与 6 种经典算法在 GD 指标上的实验结果

Table 4 Experimental results of MOFA-GD and six classical algorithms on GD

测试函数	指标	MOFA-GD	MOPSO-CD	SPEA2	MOEA/D	NSGA-II	PESA-II	MOFA
ZDT1	mean	1.5797 $\times 10^{-4}$	4.1580 $\times 10^{-4}$	1.5486 $\times 10^{-3}$	1.2063 $\times 10^{-2}$	1.1935 $\times 10^{-3}$	1.5487 $\times 10^{-3}$	9.4602 $\times 10^{-4}$
	std	2.1113 $\times 10^{-5}$	2.2613 $\times 10^{-3}$	2.9557 $\times 10^{-4}$	5.0316 $\times 10^{-3}$	2.5680 $\times 10^{-4}$	4.7015 $\times 10^{-4}$	5.3092 $\times 10^{-4}$
ZDT2	mean	4.6483 $\times 10^{-5}$	3.6238 $\times 10^{-3}$	1.8163 $\times 10^{-3}$	5.4349 $\times 10^{-4}$	1.4472 $\times 10^{-3}$	1.5546 $\times 10^{-3}$	9.6352 $\times 10^{-4}$
	std	2.4484 $\times 10^{-6}$	9.8606 $\times 10^{-3}$	3.8308 $\times 10^{-4}$	5.6675 $\times 10^{-4}$	3.9701 $\times 10^{-4}$	5.8745 $\times 10^{-4}$	2.8644 $\times 10^{-6}$
ZDT3	mean	1.5579 $\times 10^{-4}$	1.0033 $\times 10^{-3}$	1.1870 $\times 10^{-3}$	1.4266 $\times 10^{-2}$	8.3916 $\times 10^{-4}$	1.3459 $\times 10^{-3}$	1.2628 $\times 10^{-3}$
	std	1.4596 $\times 10^{-5}$	2.9992 $\times 10^{-3}$	1.2586 $\times 10^{-3}$	5.4727 $\times 10^{-3}$	7.9735 $\times 10^{-4}$	7.5793 $\times 10^{-4}$	9.4632 $\times 10^{-4}$
ZDT4	mean	5.2585 $\times 10^{-4}$	6.8083 $\times 10^0$	2.7215 $\times 10^{-2}$	6.9146 $\times 10^{-2}$	3.0052 $\times 10^{-2}$	1.3807 $\times 10^{-2}$	2.7017 $\times 10^0$
	std	2.8694 $\times 10^{-5}$	5.4800 $\times 10^0$	2.4889 $\times 10^{-2}$	3.5250 $\times 10^{-2}$	2.8266 $\times 10^{-2}$	3.8637 $\times 10^{-2}$	1.1051 $\times 10^0$
ZDT6	mean	3.3150 $\times 10^{-2}$	4.9561 $\times 10^{-2}$	8.7334 $\times 10^{-3}$	1.0812 $\times 10^{-2}$	7.3054 $\times 10^{-3}$	3.6392 $\times 10^{-3}$	2.2541 $\times 10^{-1}$
	std	2.8344 $\times 10^{-2}$	2.8408 $\times 10^{-2}$	4.6539 $\times 10^{-3}$	3.8292 $\times 10^{-3}$	3.6134 $\times 10^{-3}$	5.9133 $\times 10^{-3}$	2.1215 $\times 10^{-1}$
Viennet1	mean	1.4446 $\times 10^{-2}$	1.2424 $\times 10^{-2}$	1.3966 $\times 10^{-2}$	1.5988 $\times 10^{-2}$	1.1262 $\times 10^{-2}$	1.3562 $\times 10^{-2}$	1.4087 $\times 10^{-2}$
	std	1.2262 $\times 10^{-3}$	1.8224 $\times 10^{-3}$	1.1148 $\times 10^{-3}$	1.5173 $\times 10^{-3}$	1.6053 $\times 10^{-3}$	1.6710 $\times 10^{-3}$	1.2347 $\times 10^{-3}$
Viennet2	mean	1.8617 $\times 10^{-3}$	1.9582 $\times 10^{-3}$	1.6730 $\times 10^{-3}$	9.9840 $\times 10^{-4}$	1.9994 $\times 10^{-3}$	1.7633 $\times 10^{-3}$	2.4516 $\times 10^{-3}$
	std	1.0599 $\times 10^{-3}$	1.9222 $\times 10^{-4}$	6.0380 $\times 10^{-5}$	2.8373 $\times 10^{-5}$	1.2968 $\times 10^{-4}$	2.0598 $\times 10^{-4}$	7.0558 $\times 10^{-4}$
Viennet3	mean	3.7705 $\times 10^{-3}$	3.4290 $\times 10^{-3}$	4.1426 $\times 10^{-3}$	1.4746 $\times 10^{-3}$	3.7435 $\times 10^{-3}$	3.2771 $\times 10^{-3}$	4.3021 $\times 10^{-3}$
	std	1.4029 $\times 10^{-3}$	3.7429 $\times 10^{-4}$	2.2419 $\times 10^{-4}$	3.7110 $\times 10^{-5}$	2.6085 $\times 10^{-4}$	4.0190 $\times 10^{-4}$	3.9276 $\times 10^{-4}$
DTLZ1	mean	1.7041 $\times 10^{-3}$	6.2810 $\times 10^0$	3.5185 $\times 10^{-2}$	3.2462 $\times 10^{-2}$	4.1118 $\times 10^{-2}$	1.6928 $\times 10^{-1}$	1.7593 $\times 10^1$
	std	1.4869 $\times 10^{-3}$	6.6422 $\times 10^{-1}$	6.4340 $\times 10^{-2}$	5.3184 $\times 10^{-2}$	4.0005 $\times 10^{-2}$	4.1372 $\times 10^{-1}$	6.9750 $\times 10^0$
DTLZ2	mean	9.4096 $\times 10^{-3}$	8.5339 $\times 10^{-3}$	1.1439 $\times 10^{-3}$	6.4824 $\times 10^{-4}$	1.3374 $\times 10^{-3}$	1.6265 $\times 10^{-3}$	4.3248 $\times 10^{-2}$
	std	8.5338 $\times 10^{-3}$	1.8044 $\times 10^{-3}$	1.3759 $\times 10^{-4}$	4.6147 $\times 10^{-5}$	1.5506 $\times 10^{-4}$	2.2228 $\times 10^{-4}$	5.2943 $\times 10^{-3}$
DTLZ3	mean	1.7592 $\times 10^{-3}$	4.0485 $\times 10^1$	1.3795 $\times 10^0$	2.8785 $\times 10^0$	1.4312 $\times 10^0$	1.1830 $\times 10^0$	1.1619 $\times 10^2$
	std	4.1840 $\times 10^{-4}$	4.8136 $\times 10^0$	7.3996 $\times 10^{-1}$	2.0913 $\times 10^0$	8.4159 $\times 10^{-1}$	1.0822 $\times 10^0$	7.2704 $\times 10^1$
DTLZ4	mean	7.3557 $\times 10^{-3}$	2.3393 $\times 10^{-2}$	9.9281 $\times 10^{-4}$	4.0295 $\times 10^{-4}$	1.2039 $\times 10^{-3}$	1.5777 $\times 10^{-3}$	3.4964 $\times 10^{-2}$
	std	8.7921 $\times 10^{-3}$	1.1199 $\times 10^{-2}$	4.7151 $\times 10^{-4}$	2.3155 $\times 10^{-4}$	3.4383 $\times 10^{-4}$	1.9360 $\times 10^{-4}$	8.3391 $\times 10^{-3}$
DTLZ5	mean	1.3975 $\times 10^{-4}$	3.7954 $\times 10^{-3}$	4.1055 $\times 10^{-4}$	2.1938 $\times 10^{-4}$	2.8279 $\times 10^{-4}$	2.9674 $\times 10^{-4}$	2.6448 $\times 10^{-2}$
	std	3.8346 $\times 10^{-5}$	1.4698 $\times 10^{-3}$	7.5034 $\times 10^{-5}$	8.8732 $\times 10^{-4}$	5.8888 $\times 10^{-5}$	9.6254 $\times 10^{-5}$	8.0642 $\times 10^{-3}$
DTLZ6	mean	4.7943 $\times 10^{-6}$	1.0614 $\times 10^{-2}$	4.7789 $\times 10^{-6}$	2.0849 $\times 10^{-2}$	4.7571 $\times 10^{-6}$	2.8885 $\times 10^{-3}$	9.1980 $\times 10^{-1}$
	std	2.2657 $\times 10^{-7}$	5.2044 $\times 10^{-2}$	2.2863 $\times 10^{-7}$	4.6275 $\times 10^{-2}$	2.6931 $\times 10^{-7}$	6.9810 $\times 10^{-3}$	1.9528 $\times 10^{-1}$

续表 4

测试函数	指标	MOFA-GD	MOPSO-CD	SPEA2	MOEA/D	NSGA-II	PESA-II	MOFA
DTLZ7	mean	4.0464 $\times 10^{-3}$	4.3405 $\times 10^{-3}$	5.1354 $\times 10^{-3}$	4.6933 $\times 10^{-3}$	6.5109 $\times 10^{-3}$	6.1822 $\times 10^{-3}$	6.3770 $\times 10^{-3}$
	std	6.3265 $\times 10^{-4}$	6.2178 $\times 10^{-4}$	1.0026 $\times 10^{-3}$	1.3401 $\times 10^{-3}$	1.1154 $\times 10^{-3}$	1.6348 $\times 10^{-3}$	1.2759 $\times 10^{-2}$
total		9	0	0	4	1	1	0
ranking		2.80	4.73	3.60	3.60	3.47	3.73	6.07

注: 加粗的数据为各算法在同一测试函数上取得的最优结果。

表 5 MOFA-GD 与 6 种经典算法在 MS 指标上的实验结果
Table 5 Experimental results of MOFA-GD and six classical algorithms on MS

测试函数	指标	MOFA-GD	MOPSO-CD	SPEA2	MOEA/D	NSGA-II	PESA-II	MOFA
ZDT1	mean	1.0000 $\times 10^0$	1.0000 $\times 10^0$	9.9184 $\times 10^{-1}$	6.5493 $\times 10^{-1}$	9.9552 $\times 10^{-1}$	9.3366 $\times 10^{-1}$	6.9247 $\times 10^{-1}$
	std	0.0000 $\times 10^0$	0.0000 $\times 10^0$	4.3402 $\times 10^{-3}$	1.3790 $\times 10^{-1}$	8.8171 $\times 10^{-4}$	8.9534 $\times 10^{-2}$	2.5355 $\times 10^{-2}$
ZDT2	mean	1.0000 $\times 10^0$	1.0000 $\times 10^0$	9.8581 $\times 10^{-1}$	1.3608 $\times 10^{-1}$	8.3096 $\times 10^{-1}$	7.9724 $\times 10^{-1}$	7.0711 $\times 10^{-1}$
	std	0.0000 $\times 10^0$	0.0000 $\times 10^0$	2.0411 $\times 10^{-3}$	5.2328 $\times 10^{-2}$	2.7449 $\times 10^{-1}$	2.9011 $\times 10^{-1}$	0.0000 $\times 10^0$
ZDT3	mean	9.9996 $\times 10^{-1}$	9.8022 $\times 10^{-1}$	9.9366 $\times 10^{-1}$	6.6855 $\times 10^{-1}$	9.9437 $\times 10^{-1}$	8.5050 $\times 10^{-1}$	6.5297 $\times 10^{-1}$
	std	4.8976 $\times 10^{-5}$	2.4984 $\times 10^{-2}$	9.0955 $\times 10^{-4}$	1.4231 $\times 10^{-2}$	3.7976 $\times 10^{-3}$	1.1396 $\times 10^{-1}$	4.9455 $\times 10^{-2}$
ZDT4	mean	1.0000 $\times 10^0$	3.8371 $\times 10^{-1}$	6.2497 $\times 10^{-1}$	6.2654 $\times 10^{-1}$	6.5809 $\times 10^{-1}$	5.7167 $\times 10^{-1}$	4.3224 $\times 10^{-1}$
	std	0.0000 $\times 10^0$	2.8228 $\times 10^{-1}$	2.5438 $\times 10^{-1}$	9.9928 $\times 10^{-2}$	1.5300 $\times 10^{-1}$	1.6921 $\times 10^{-1}$	4.5264 $\times 10^{-2}$
ZDT6	mean	1.0000 $\times 10^0$	1.0000 $\times 10^0$	9.5415 $\times 10^{-1}$	8.7609 $\times 10^{-1}$	9.4777 $\times 10^{-1}$	9.6437 $\times 10^{-1}$	5.8983 $\times 10^{-1}$
	std	1.4259 $\times 10^{-6}$	3.5081 $\times 10^{-7}$	2.3633 $\times 10^{-2}$	1.0001 $\times 10^{-1}$	2.1674 $\times 10^{-2}$	1.9990 $\times 10^{-3}$	1.9757 $\times 10^{-1}$
Viennet1	mean	9.9999 $\times 10^{-1}$	9.9994 $\times 10^{-1}$	9.9786 $\times 10^{-1}$	7.7898 $\times 10^{-1}$	9.9901 $\times 10^{-1}$	9.7234 $\times 10^{-1}$	6.9402 $\times 10^{-1}$
	std	8.2326 $\times 10^{-3}$	4.5693 $\times 10^{-5}$	2.2057 $\times 10^{-3}$	2.3568 $\times 10^{-3}$	1.7065 $\times 10^{-3}$	4.0741 $\times 10^{-2}$	4.8855 $\times 10^{-2}$
Viennet2	mean	9.9834 $\times 10^{-1}$	9.9041 $\times 10^{-1}$	9.9966 $\times 10^{-1}$	2.5711 $\times 10^{-1}$	9.9927 $\times 10^{-1}$	9.8993 $\times 10^{-1}$	6.3116 $\times 10^{-1}$
	std	1.6710 $\times 10^{-3}$	3.9147 $\times 10^{-3}$	3.3057 $\times 10^{-4}$	3.6598 $\times 10^{-3}$	6.9880 $\times 10^{-4}$	8.1029 $\times 10^{-3}$	1.2115 $\times 10^{-1}$
Viennet3	mean	9.9889 $\times 10^{-1}$	9.9804 $\times 10^{-1}$	9.9598 $\times 10^{-1}$	5.2283 $\times 10^{-1}$	9.9895 $\times 10^{-1}$	9.9502 $\times 10^{-1}$	3.9336 $\times 10^{-1}$
	std	9.4656 $\times 10^{-4}$	2.3929 $\times 10^{-3}$	8.1494 $\times 10^{-4}$	7.7723 $\times 10^{-3}$	1.8107 $\times 10^{-3}$	6.0191 $\times 10^{-3}$	7.4213 $\times 10^{-2}$
DTLZ1	mean	9.9538 $\times 10^{-1}$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	9.4027 $\times 10^{-1}$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	8.1650 $\times 10^{-1}$
	std	5.4572 $\times 10^{-3}$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	1.0129 $\times 10^{-1}$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	1.3597E ⁻¹⁶
DTLZ2	mean	9.9959 $\times 10^{-1}$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	9.9623 $\times 10^{-1}$	6.1030 $\times 10^{-1}$
	std	8.1412 $\times 10^{-4}$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	3.3221 $\times 10^{-3}$	5.7050 $\times 10^{-2}$
DTLZ3	mean	9.9968 $\times 10^{-1}$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	9.8288 $\times 10^{-1}$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	8.1650 $\times 10^{-1}$
	std	3.4151 $\times 10^{-4}$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	2.0122 $\times 10^{-2}$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	1.3597E ⁻¹⁶
DTLZ4	mean	1.0000 $\times 10^0$	1.0000 $\times 10^0$	9.3883 $\times 10^{-1}$	5.5563 $\times 10^{-3}$	6.6667 $\times 10^{-1}$	9.9948 $\times 10^{-1}$	1.0000 $\times 10^0$
	std	0.0000 $\times 10^0$	0.0000 $\times 10^0$	1.0595 $\times 10^{-1}$	9.6238 $\times 10^{-3}$	5.7735 $\times 10^{-1}$	8.9238 $\times 10^{-4}$	0.0000 $\times 10^0$
DTLZ5	mean	1.0000 $\times 10^0$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	9.8726 $\times 10^{-1}$	7.9163 $\times 10^{-1}$
	std	0.0000 $\times 10^0$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	1.6742 $\times 10^{-2}$	1.0510 $\times 10^{-1}$
DTLZ6	mean	1.0000 $\times 10^0$	1.0000 $\times 10^0$	1.0000 $\times 10^0$	9.9409 $\times 10^{-1}$	1.0000 $\times 10^0$	9.9463 $\times 10^{-1}$	8.1650 $\times 10^{-1}$
	std	0.0000 $\times 10^0$	0.0000 $\times 10^0$	0.0000 $\times 10^0$	9.9481 $\times 10^{-3}$	0.0000 $\times 10^0$	8.9274 $\times 10^{-3}$	1.3597E ⁻¹⁶
DTLZ7	mean	9.9913 $\times 10^{-1}$	9.8816 $\times 10^{-1}$	9.9094 $\times 10^{-1}$	9.8876 $\times 10^{-1}$	9.8702 $\times 10^{-1}$	8.5730 $\times 10^{-1}$	7.2191 $\times 10^{-1}$
	std	8.6620 $\times 10^{-4}$	5.1758 $\times 10^{-3}$	1.6569 $\times 10^{-3}$	1.7458 $\times 10^{-2}$	8.7592 $\times 10^{-4}$	1.4787 $\times 10^{-1}$	8.2089 $\times 10^{-2}$
total		11	9	6	2	5	1	1
ranking		5.06	5.10	4.87	2.57	4.93	3.33	1.60

注: 加粗的数据为各算法在同一测试函数上取得的最优结果。

表 6 MOFA-GD 与 6 种经典算法在 IGD 指标上的实验结果
Table 6 Experimental results of MOFA-GD and six classical algorithms on IGD

测试函数	指标	MOFA-GD	MOPSO-CD	SPEA2	MOEA/D	NSGA-II	PESA-II	MOFA
ZDT1	mean	3.8849 $\times 10^{-3}$	7.2150 $\times 10^{-3}$	1.5893 $\times 10^{-2}$	1.5067 $\times 10^{-1}$	1.2773 $\times 10^{-2}$	3.4504 $\times 10^{-2}$	5.3718 $\times 10^{-1}$
	std	3.5299 $\times 10^{-5}$	2.1606 $\times 10^{-2}$	2.7429 $\times 10^{-3}$	6.2132 $\times 10^{-2}$	2.2630 $\times 10^{-3}$	3.0983 $\times 10^{-2}$	7.2900 $\times 10^{-2}$
ZDT2	mean	3.8654 $\times 10^{-3}$	5.9912 $\times 10^{-2}$	2.0011 $\times 10^{-2}$	5.2955 $\times 10^{-1}$	3.0198 $\times 10^{-2}$	9.7354 $\times 10^{-2}$	6.0793 $\times 10^{-1}$
	std	3.0624 $\times 10^{-5}$	1.4181 $\times 10^{-1}$	7.6091 $\times 10^{-3}$	7.8624 $\times 10^{-2}$	4.3802 $\times 10^{-2}$	7.3808 $\times 10^{-2}$	7.2257 $\times 10^{-3}$
ZDT3	mean	4.7560 $\times 10^{-3}$	1.3238 $\times 10^{-2}$	1.4667 $\times 10^{-2}$	1.4836 $\times 10^{-1}$	1.5712 $\times 10^{-2}$	3.5133 $\times 10^{-2}$	4.6203 $\times 10^{-1}$
	std	1.5045 $\times 10^{-4}$	2.5378 $\times 10^{-2}$	6.8054 $\times 10^{-3}$	4.4974 $\times 10^{-2}$	1.7003 $\times 10^{-2}$	2.3084 $\times 10^{-2}$	9.4936 $\times 10^{-2}$
ZDT4	mean	3.9754 $\times 10^{-3}$	1.4990 $\times 10^1$	2.2729 $\times 10^{-1}$	4.9037 $\times 10^{-1}$	3.0316 $\times 10^{-1}$	1.6025 $\times 10^{-1}$	2.7495 $\times 10^{-1}$
	std	7.9899 $\times 10^{-5}$	8.1023 $\times 10^0$	1.3432 $\times 10^{-1}$	2.0115 $\times 10^{-1}$	1.8224 $\times 10^{-1}$	9.1766 $\times 10^{-2}$	7.8013 $\times 10^{-3}$
ZDT6	mean	3.1228 $\times 10^{-3}$	3.9721 $\times 10^{-3}$	6.2222 $\times 10^{-2}$	7.6398 $\times 10^{-2}$	5.3301 $\times 10^{-2}$	2.3854 $\times 10^{-2}$	1.4086 $\times 10^{-1}$
	std	3.4597 $\times 10^{-5}$	6.5297 $\times 10^{-4}$	2.6778 $\times 10^{-2}$	2.4561 $\times 10^{-2}$	2.5303 $\times 10^{-2}$	1.0888 $\times 10^{-2}$	2.2434 $\times 10^{-1}$
Viennet1	mean	1.2650 $\times 10^{-1}$	1.4970 $\times 10^{-1}$	1.2635 $\times 10^{-1}$	2.0452 $\times 10^{-1}$	1.5732 $\times 10^{-1}$	1.4052 $\times 10^{-1}$	2.9885 $\times 10^{-1}$
	std	2.9588 $\times 10^{-3}$	5.5004 $\times 10^{-3}$	2.3433 $\times 10^{-3}$	4.1264 $\times 10^{-3}$	8.4072 $\times 10^{-3}$	5.0250 $\times 10^{-3}$	2.2626 $\times 10^{-2}$
Viennet2	mean	2.6223 $\times 10^{-2}$	1.9021 $\times 10^{-2}$	1.2697 $\times 10^{-2}$	1.1043 $\times 10^{-1}$	2.1668 $\times 10^{-2}$	1.9136 $\times 10^{-2}$	2.4455 $\times 10^{-2}$
	std	4.1432 $\times 10^{-3}$	1.1159 $\times 10^{-3}$	3.7431 $\times 10^{-4}$	1.0514 $\times 10^{-3}$	1.8111 $\times 10^{-3}$	1.1918 $\times 10^{-3}$	3.0092 $\times 10^{-3}$
Viennet3	mean	4.4177 $\times 10^{-2}$	3.7074 $\times 10^{-2}$	3.3686 $\times 10^{-2}$	2.0249 $\times 10^0$	4.1283 $\times 10^{-2}$	8.9543 $\times 10^{-2}$	4.6168 $\times 10^{-2}$
	std	6.0781 $\times 10^{-4}$	1.7387 $\times 10^{-3}$	9.4065 $\times 10^{-3}$	4.1172 $\times 10^{-2}$	3.1965 $\times 10^{-3}$	2.2292 $\times 10^{-1}$	3.9934 $\times 10^{-3}$
DTLZ1	mean	3.7092 $\times 10^{-2}$	1.7421 $\times 10^1$	1.6921 $\times 10^{-1}$	2.0533 $\times 10^{-1}$	2.8926 $\times 10^{-1}$	2.0599 $\times 10^{-1}$	2.0671 $\times 10^{-1}$
	std	3.7023 $\times 10^{-3}$	5.0585 $\times 10^0$	1.5299 $\times 10^{-1}$	2.6218 $\times 10^{-1}$	2.6636 $\times 10^{-1}$	3.0184 $\times 10^{-1}$	5.2550 $\times 10^{-3}$
DTLZ2	mean	8.0807 $\times 10^{-2}$	9.9274 $\times 10^{-2}$	5.4686 $\times 10^{-2}$	5.4904 $\times 10^{-2}$	6.9291 $\times 10^{-2}$	6.7582 $\times 10^{-2}$	2.3675 $\times 10^{-1}$
	std	6.3738 $\times 10^{-3}$	1.0452 $\times 10^{-2}$	6.2684 $\times 10^{-4}$	1.9886 $\times 10^{-4}$	3.6187 $\times 10^{-3}$	3.9193 $\times 10^{-3}$	2.0530 $\times 10^{-2}$
DTLZ3	mean	5.9972 $\times 10^{-2}$	1.1464 $\times 10^2$	7.8474 $\times 10^0$	1.7444 $\times 10^1$	8.3355 $\times 10^0$	7.4589 $\times 10^0$	5.6091 $\times 10^{-1}$
	std	4.7918 $\times 10^{-3}$	3.6816 $\times 10^1$	4.1377 $\times 10^0$	1.3031 $\times 10^1$	4.9823 $\times 10^0$	6.4979 $\times 10^0$	4.7536 $\times 10^{-3}$
DTLZ4	mean	1.6353 $\times 10^{-1}$	2.9417 $\times 10^{-1}$	2.1555 $\times 10^{-1}$	4.3956 $\times 10^{-1}$	1.1553 $\times 10^{-1}$	6.4794 $\times 10^{-2}$	6.4337 $\times 10^{-1}$
	std	3.8610 $\times 10^{-2}$	4.8680 $\times 10^{-2}$	2.7442 $\times 10^{-1}$	3.2580 $\times 10^{-1}$	1.6505 $\times 10^{-1}$	1.9913 $\times 10^{-3}$	5.1571 $\times 10^{-2}$
DTLZ5	mean	4.4077 $\times 10^{-3}$	2.4077 $\times 10^{-2}$	5.2891 $\times 10^{-3}$	3.2317 $\times 10^{-2}$	6.0572 $\times 10^{-3}$	1.2854 $\times 10^{-2}$	7.2396 $\times 10^{-2}$
	std	1.0577 $\times 10^{-4}$	5.7398 $\times 10^{-3}$	3.2714 $\times 10^{-4}$	8.1148 $\times 10^{-4}$	2.7943 $\times 10^{-4}$	1.7048 $\times 10^{-3}$	1.5063 $\times 10^{-2}$
DTLZ6	mean	4.1229 $\times 10^{-3}$	3.3619 $\times 10^{-2}$	5.7534 $\times 10^{-3}$	1.5875 $\times 10^{-1}$	6.1384 $\times 10^{-3}$	1.5459 $\times 10^{-2}$	6.0019 $\times 10^{-1}$
	std	3.1729 $\times 10^{-5}$	1.2390 $\times 10^{-1}$	6.4367 $\times 10^{-3}$	2.9157 $\times 10^{-1}$	1.1575 $\times 10^{-3}$	6.6019 $\times 10^{-3}$	1.1210 $\times 10^{-1}$
DTLZ7	mean	9.2645 $\times 10^{-2}$	8.5723 $\times 10^{-2}$	8.8454 $\times 10^{-2}$	1.8334 $\times 10^{-1}$	1.0638 $\times 10^{-1}$	1.6993 $\times 10^{-1}$	1.4738 $\times 10^0$
	std	1.0077 $\times 10^{-2}$	5.9124 $\times 10^{-3}$	5.7305 $\times 10^{-2}$	1.2903 $\times 10^{-1}$	5.5775 $\times 10^{-2}$	1.5060 $\times 10^{-1}$	9.6465 $\times 10^{-2}$
total		10	1	3	0	0	1	0
ranking		2.13	4.07	2.47	5.67	3.87	3.76	6.07

注: 加粗的数据为各算法在同一测试函数上取得的最优结果。

根据表 4, MOFA-GD 在收敛性能上取得了 9 次最优, 相比较于其他算法取得最优次数最多, 尤其是在 ZDT4、DTLZ1、DTLZ3 这 3 个具有多模态特

征的测试函数上取得的 GD 均值数量级比其他算法取得的 GD 均值数量级优, 充分证明其收敛性能优、寻优精度高的特点。由表 5 可知, MOFA-GD

计算所得的 Pareto 解集在真实 Pareto 解集的覆盖范围也取得了优秀的结果, 在 15 个测试函数上取得了 11 次最优, 并且在所有的测试函数上的覆盖程度接近 1, 充分验证其覆盖范围广泛的优点。从表 6 给出的 IGD 平均值来看, MOFA-GD 在 15 个测试函数中取得 10 次最优, 在所有的优化算法中占优最多, 其中在 5 个 2 目标测试函数全部占优; 从 IGD 标准差来看, MOFA-GD 在 10 个占优测试函数上有 8 个所得的 IGD 标准差比其他算法至少优一个数量级, 说明其具有多样性高、稳定性良好的特点。从表 4~6 给出的各算法基于 Friedman 检验结果可知, MOFA-GD 在 IGD、GD 和 MS 上均排名第一, 说明其不同测试函数及不同指标上的差异较小, 进一步验证了 MOFA-GD 的稳定性和可靠性。表 7 给出 MOFA-GD 与 6 种经典算法的 CPU 运行时间。根据表 7, MOPSO-CD 算法在 8 个测试函数上的运行时间最快, NSGA-II 在

5 个测试函数上表现最佳, PESA-II 在 2 个测试函数上表现出最优的运行速度, 而 MOFA 和 MOFA-GD 的表现处于劣势。究其原因, 除 MOFA 和 MOFA-GD 外, 其余算法的时间复杂度为 $O(N)$, 而 MOFA 采用全吸引模型, 其时间复杂度为 $O(N^2)$, 致使 MOFA 的在求解过程中需要更多的计算资源; MOFA-GD 是基于 MOFA 改进的算法, 根据 2.5 节对算法时间复杂度的分析可知, MOFA-GD 的时间复杂度与 MOFA 在同一数量级, 但 MOFA-GD 进行了决策变量分组优化等步骤, 消耗了额外计算资源, 使 MOFA-GD 的运行时间稍慢于 MOFA。结合表 4~6 中的 IGD、GD 和 MS 结果可知, MOFA-GD 表现出优秀的收敛性和分布性, 在所有测试函数上具有较高的精度。正如“世上没有免费的午餐”定律, MOFA-GD 获得了更优的综合性能, 必然付出了额外的计算代价。综合来看, MOFA-GD 与 6 种经典算法相比具有较强的竞争力。

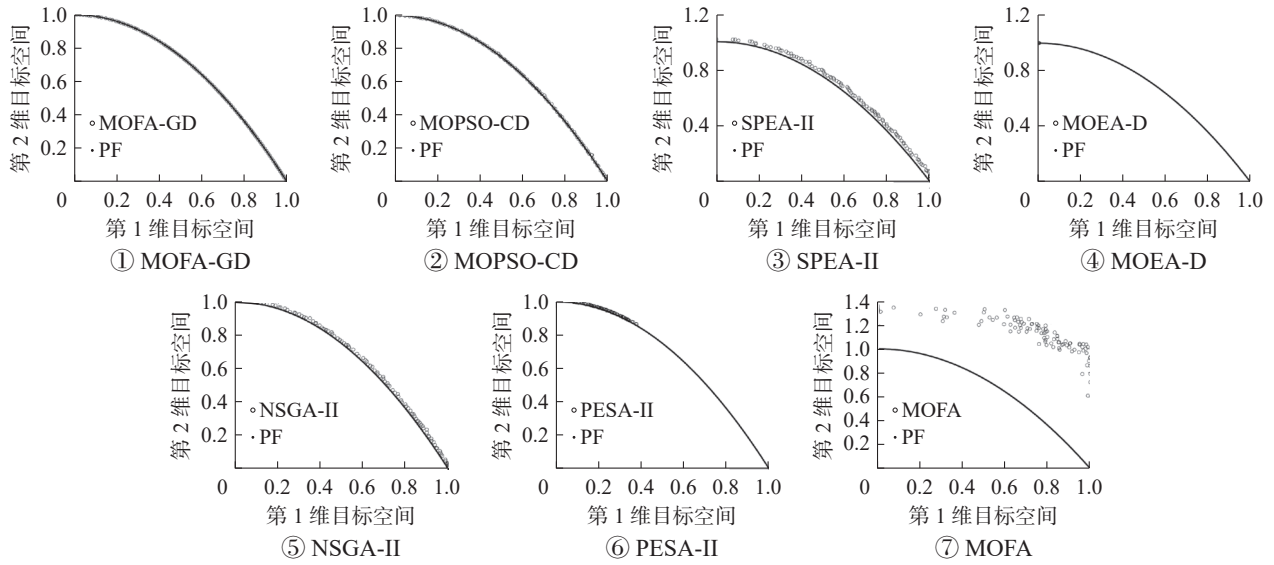
表 7 MOFA-GD 与 6 种经典算法在 CPU 运行时间的实验结果
Table 7 Experimental results of MOFA-GD and six classical algorithms on CPU running time

测试函数	MOFA-GD	MOPSO-CD	SPEA2	MOEA/D	NSGA-II	PESA-II	MOFA
ZDT1	1.6736×10 ¹	4.7239×10⁰	5.5465×10 ⁰	7.4319×10 ⁰	4.8380×10 ⁰	4.8619×10 ⁰	1.7050×10 ¹
ZDT2	1.7184×10 ¹	4.8023×10⁰	6.4233×10 ⁰	7.2925×10 ⁰	4.8272×10 ⁰	4.8329×10 ⁰	1.5665×10 ¹
ZDT3	1.4033×10 ¹	4.7961×10⁰	5.0801×10 ⁰	7.2905×10 ⁰	4.8417×10 ⁰	4.8304×10 ⁰	1.2033×10 ¹
ZDT4	1.6588×10 ¹	4.8048×10⁰	6.3097×10 ⁰	7.1388×10 ⁰	4.8215×10 ⁰	4.8283×10 ⁰	1.4565×10 ¹
ZDT6	1.9193×10 ¹	4.8095×10⁰	6.3342×10 ⁰	7.1877×10 ⁰	4.8343×10 ⁰	4.8263×10 ⁰	1.8775×10 ¹
Viennet1	2.2136×10 ¹	4.8432×10 ⁰	6.0527×10 ⁰	7.2786×10 ⁰	4.8215×10⁰	4.8269×10 ⁰	2.3516×10 ¹
Viennet2	1.1267×10 ¹	4.7977×10⁰	6.3067×10 ⁰	8.0447×10 ⁰	4.8078×10 ⁰	4.8030×10 ⁰	1.1359×10 ¹
Viennet3	1.2774×10 ¹	4.8075×10⁰	7.0385×10 ⁰	7.0981×10 ⁰	4.8308×10 ⁰	4.8235×10 ⁰	1.1848×10 ¹
DTLZ1	1.5298×10 ¹	4.8214×10⁰	6.5560×10 ⁰	7.5724×10 ⁰	4.8261×10 ⁰	4.8252×10 ⁰	1.0821×10 ¹
DTLZ2	1.8327×10 ¹	4.8296×10 ⁰	6.6649×10 ⁰	7.0017×10 ⁰	4.8793×10 ⁰	4.8220×10⁰	1.1174×10 ¹
DTLZ3	1.4658×10 ¹	4.8719×10 ⁰	5.8858×10 ⁰	7.8922×10 ⁰	4.6352×10⁰	4.6412×10 ⁰	1.6373×10 ¹
DTLZ4	1.1558×10 ¹	4.7883×10 ⁰	5.2525×10 ⁰	7.6795×10 ⁰	4.6482×10⁰	4.6734×10 ⁰	1.0091×10 ¹
DTLZ5	1.9149×10 ¹	4.6411×10 ⁰	5.6652×10 ⁰	7.4119×10 ⁰	4.6367×10⁰	4.6401×10 ⁰	1.6204×10 ¹
DTLZ6	2.4564×10 ¹	4.6695×10 ⁰	5.2249×10 ⁰	7.5378×10 ⁰	4.6398×10⁰	4.6422×10 ⁰	1.9895×10 ¹
DTLZ7	1.9876×10 ¹	4.6508×10 ⁰	5.1222×10 ⁰	7.2282×10 ⁰	4.6432×10 ⁰	4.6378×10⁰	1.6925×10 ¹

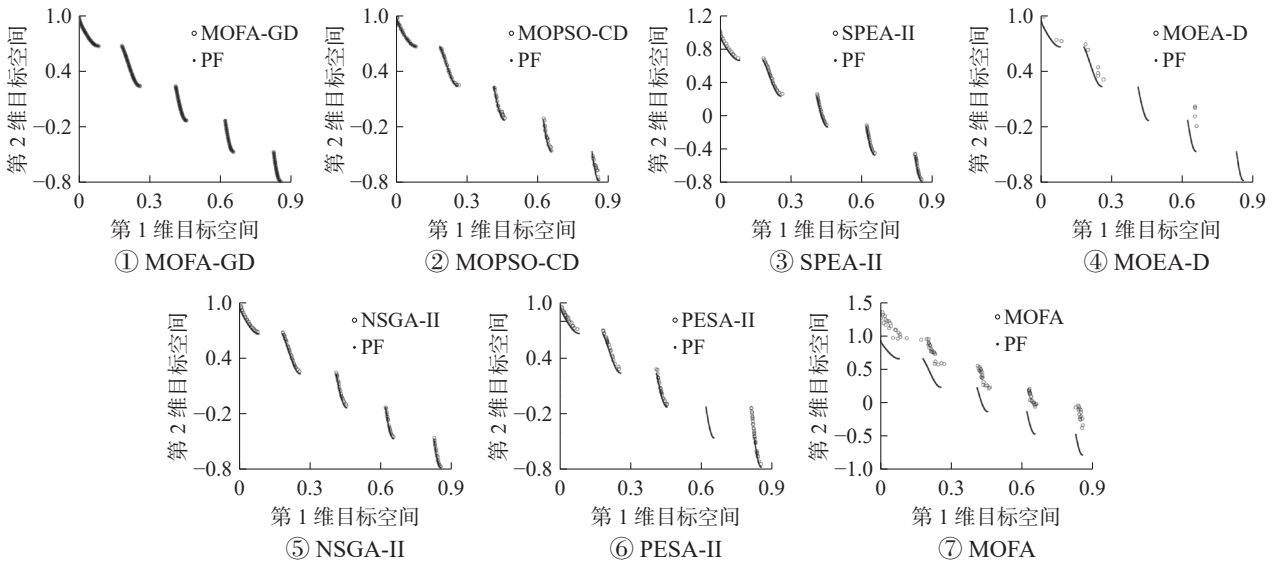
注: 加粗的数据为运行时间最优的结果。

为了更直观地体现 MOFA-GD 的性能优势, 图 2 给出了 MOFA-GD 和 6 种经典多目标优化算法在 6 个测试函数上的 Pareto 前沿拟合曲线, 2 目标测试函数选取 PF 特征为非凸 (ZDT2)、非连

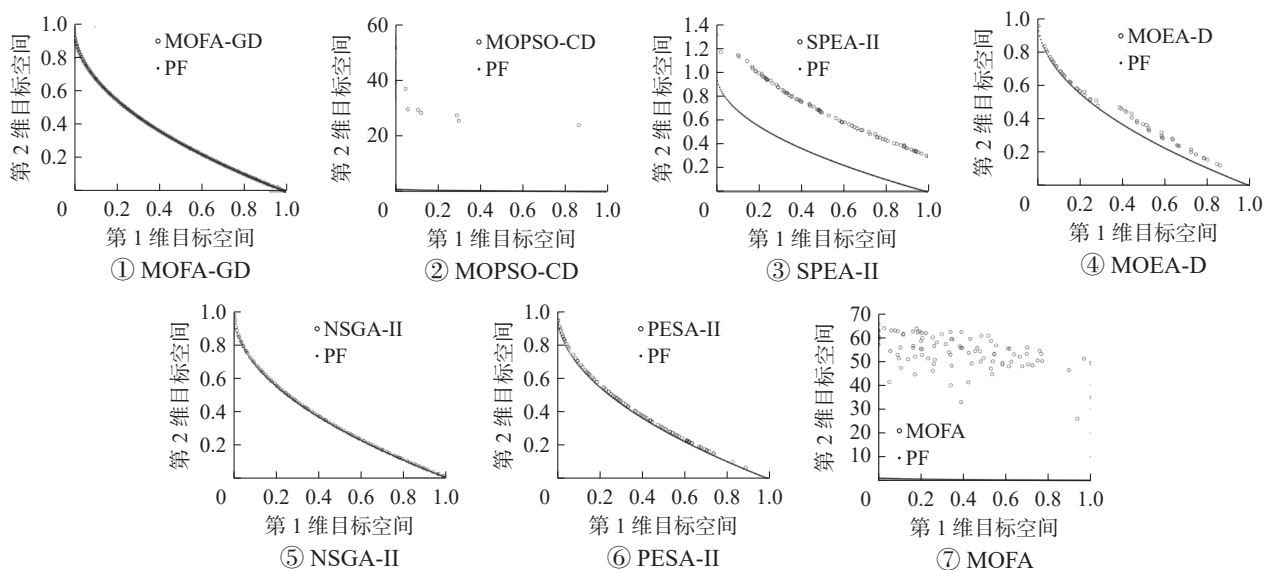
续 (ZDT3) 和多模态 (ZDT4), 3 目标测试函数选取 PF 特征为非凸 (DTLZ2)、多模态 (DTLZ3) 和非连续 (DTLZ7)。图 3 给出了各算法在上述 6 个测试函数的 IGD 收敛曲线。



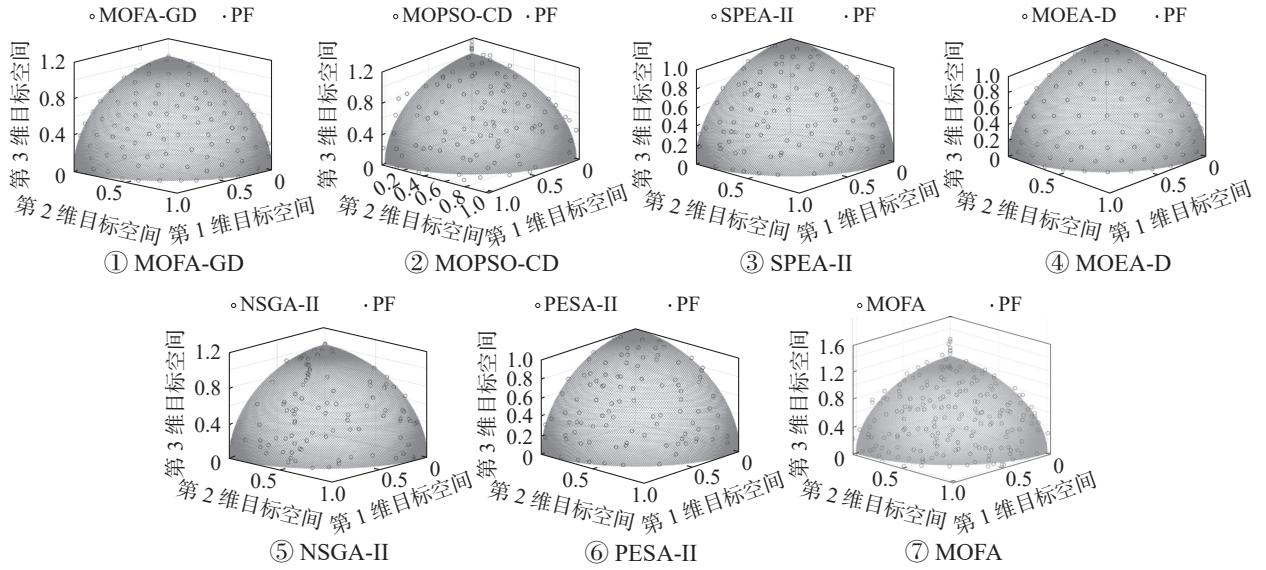
(a) ZDT1 上的 Pareto 前沿拟合曲线



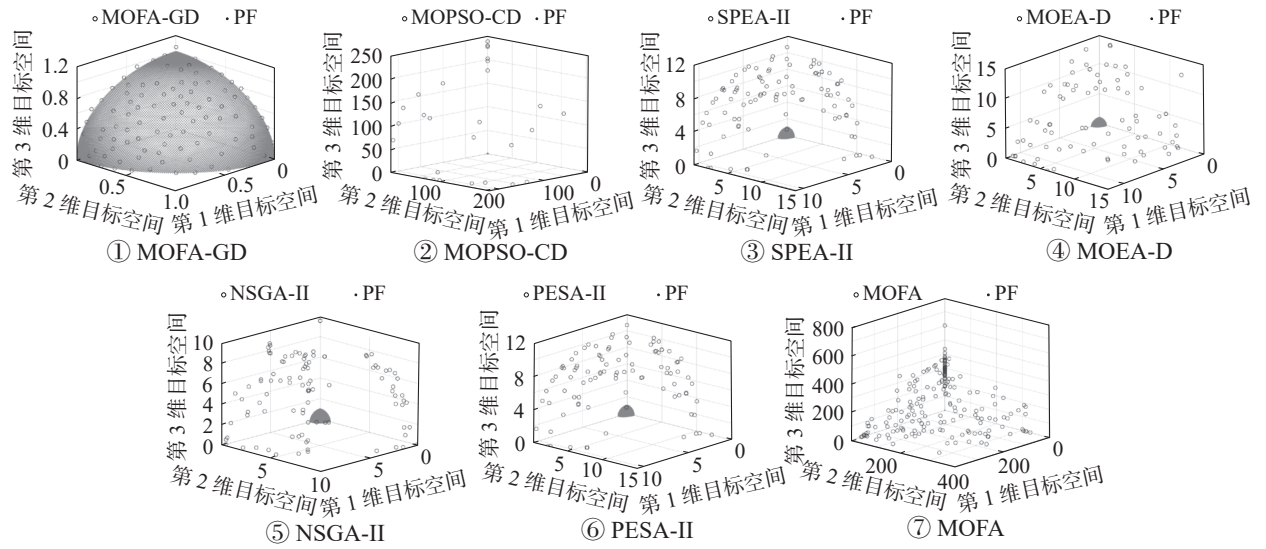
(b) ZDT3 上的 Pareto 前沿拟合曲线



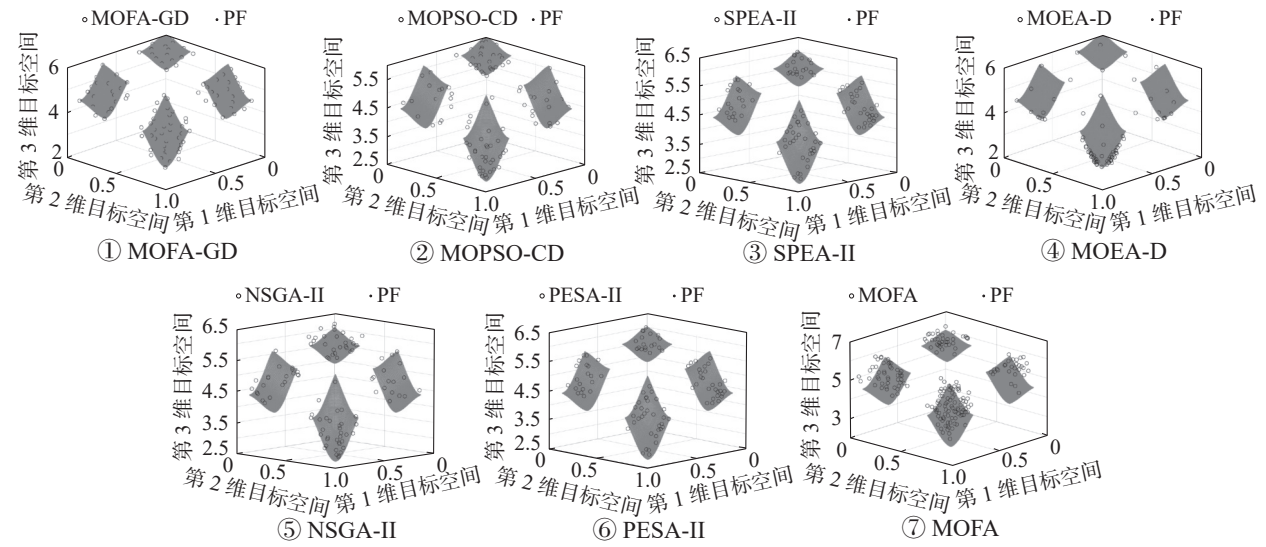
(c) ZDT4 上的 Pareto 前沿拟合曲线



(d) DTLZ2 上的 Pareto 前沿拟合曲线



(e) DTLZ3 上的 Pareto 前沿拟合曲线



(f) DTLZ7 上的 Pareto 前沿拟合曲线

图 2 MOFA-GD 与 6 种经典算法的 Pareto 前沿拟合曲线

Fig. 2 Fitting of Pareto fronts of MOFA-GD and six classical algorithms

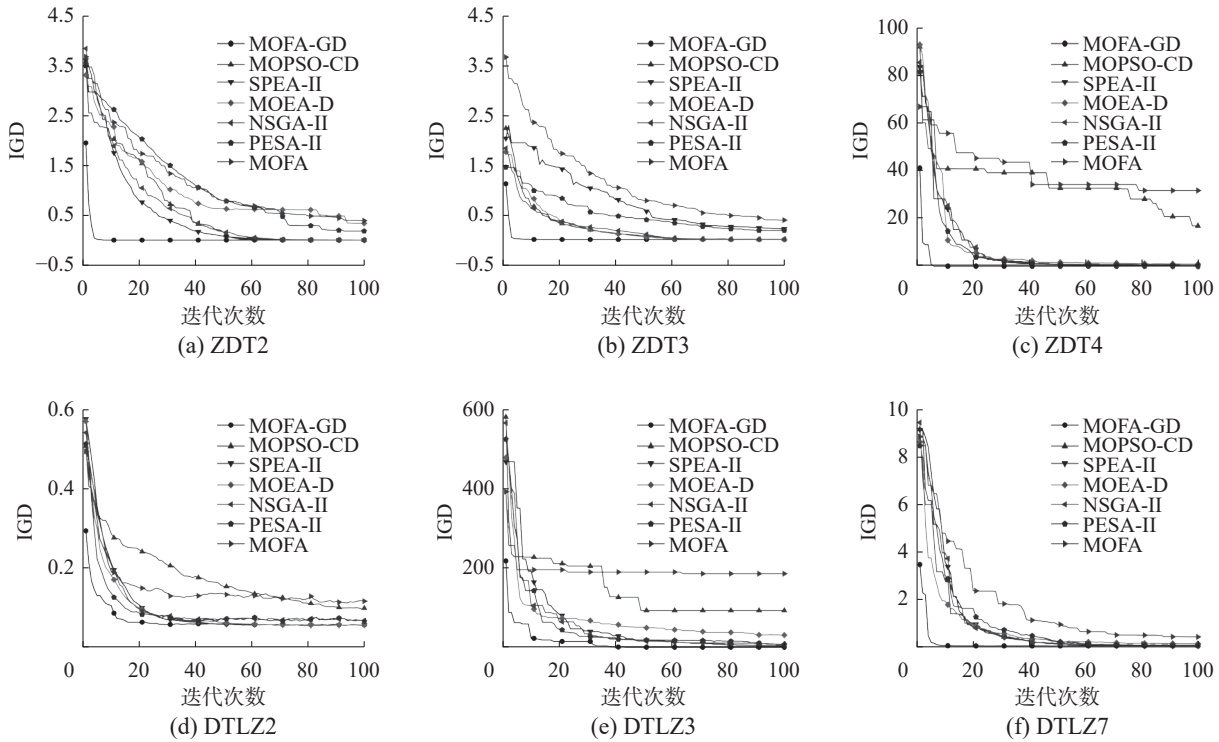


图 3 各算法在 6 个测试函数的 IGD 收敛曲线
 Fig. 3 IGD convergence curve for each algorithm for 6 test functions

从图 2 可知, MOFA-GD 算法得出的 Pareto 前沿在 6 个测试问题上都能够拟合到真实 Pareto 前沿, 说明 MOFA-GD 在求解非连续、多模态等复杂优化函数时具有优秀的收敛性, 同时解集在 Pareto 前沿上均匀分布, 原因是档案截断策略能够精准识别较拥挤的个体, 动态调整萤火虫的分布, 使其分布均匀。根据图 3, MOFA-GD 在各测试函数上迭代到 10 代左右 IGD 值趋于最优且呈平稳趋势, 而其他算法大约迭代到 50 代才能逐渐平稳, 由此可见 MOFA-GD 比其他算法收敛速度更快且精度更高, 尤其在 ZDT4 和 DTLZ3 测试函数上, MOFA-GD 不仅能够快速收敛到 IGD 最优值附近, 还能保持平稳不再有波动。究其原因, MOFA-GD 采用了学习行为, 使萤火虫能够快速靠近到优秀个体附近, 随着迭代的进行能够学习多个优秀个体的信息, 使得种群稳定分布在真实 Pareto 前沿附近。

3.4 MOFA-GD 与新近多目标优化算法对比实验

为了进一步验证 MOFA-GD 的性能, 本节采用 3.1 节的测试问题, 将 MOFA-GD 与 10 种在近几年内提出、具有创新性的多目标优化算法进行仿真实验对比。10 种算法分别为基于竞争机制的

多目标粒子群算法 (CMOPSO)^[30]、基于参考点非支配排序的进化多目标优化算法 (NSGA-III)^[31]、基于参考点支配的快速非支配排序算法 (RPD-NSGA-II)^[32]、多阶段进化算法 (MSEA)^[33]、基于参考点引导后代的大规模多目标优化算法 (FLEA)^[34]、基于动态 Thompson 自适应算子选择的多目标差分进化算法 (MOEA/D-DYTS)^[35]、求解大规模稀疏优化问题的基于学习 Pareto 最优子空间的多目标进化算法 (MOEA-PSL)^[36]、基于聚类的自适应多目标进化算法 (CA-MOEA)^[37]、基于竞争因子和精英解引导的多目标萤火虫算法 (CFMOFA)^[14] 和基于决策变量分析的多目标进化算法 (MOEA/DVA)^[20], 上述算法针对经典算法的不足进行了改进, 并且在实验中验证其具备良好性能。各个算法采用的实验参数均采用 PLATEMO 的默认参数。实验设置同 3.3 节, 为保证实验随机性, 所有算法独立运行 50 次。

表 8~10 给出了 MOFA-GD 与 10 种新近多目标优化算法在 15 个测试函数计算的 IGD、GD 和 MS 平均值。倒数第 2 行给出了各算法的优势统计结果 (total), 最后 1 行是各算法在 15 个测试函数优化结果基于 Friedman 检验的平秩 (ranking)。

表 8 MOFA-GD 与 10 种新近算法在 IGD 上的实验结果
Table 8 Experimental results of MOFA-GD and ten recent algorithms on IGD

测试函数	MOFA-GD	CMOPSO	NSGA-III	RPD-NSGA-II	MSEA	FLEA	MOEA/D-DYTS	MOEA-PSL	CA-MOEA	CFMOFA	MOEA/DVA
ZDT1	3.88×10⁻³	5.07×10 ⁻³	1.99×10 ⁻²	1.41×10 ⁻²	6.42×10 ⁻³	1.17×10 ⁻¹	3.15×10 ⁻¹	4.65×10 ⁻³	1.29×10 ⁻²	3.57×10 ⁻¹	8.18×10 ⁻¹
ZDT2	3.87×10⁻³	4.58×10 ⁻³	6.45×10 ⁻²	2.16×10 ⁻²	9.73×10 ⁻¹	5.93×10 ⁻¹	5.93×10 ⁻¹	5.53×10 ⁻³	2.81×10 ⁻²	5.28×10 ⁻¹	1.47×10 ⁰
ZDT3	4.76×10⁻³	5.42×10 ⁻³	1.80×10 ⁻²	1.80×10 ⁻²	1.36×10 ⁻²	1.82×10 ⁻¹	3.35×10 ⁻¹	5.27×10 ⁻³	1.32×10 ⁻²	4.17×10 ⁻¹	6.66×10 ⁻¹
ZDT4	3.98×10⁻³	7.81×10 ⁰	5.11×10 ⁻¹	4.07×10 ⁻¹	8.76×10 ⁰	2.67×10 ¹	1.11×10 ¹	4.57×10 ⁻³	3.89×10 ⁻¹	3.43×10 ¹	7.93×10 ⁰
ZDT6	3.12×10⁻³	3.21×10 ⁻³	1.74×10 ⁻¹	4.66×10 ⁻²	1.45×10 ⁻¹	4.43×10 ⁻¹	1.73×10 ⁻²	3.65×10 ⁻³	1.62×10 ⁻¹	3.19×10 ⁰	8.80×10 ⁻¹
Viennet1	1.27×10 ⁻¹	1.31×10 ⁻¹	1.69×10 ⁻¹	1.88×10 ⁻¹	1.15×10⁻¹	1.70×10 ⁻¹	1.86×10 ⁻¹	1.80×10 ⁻¹	1.34×10 ⁻¹	1.31×10 ⁻¹	6.97×10 ⁻¹
Viennet2	2.62×10 ⁻²	1.29×10 ⁻²	5.63×10 ⁻²	4.17×10 ⁻²	1.28×10⁻²	3.32×10 ⁻²	4.61×10 ⁻²	2.04×10 ⁻²	1.33×10 ⁻²	1.32×10 ⁻²	1.19×10 ⁻¹
Viennet3	4.42×10 ⁻²	3.21×10⁻²	1.76×10 ⁰	1.88×10 ⁻¹	7.03×10 ⁻²	4.28×10 ⁻²	7.54×10 ⁻¹	2.32×10 ⁻¹	3.55×10 ⁻²	6.00×10 ⁻²	5.69×10 ⁻¹
DTLZ1	3.71×10⁻²	1.37×10 ¹	1.64×10 ⁻¹	1.10×10 ⁻¹	9.61×10 ⁻²	6.78×10 ⁰	2.31×10 ⁰	1.44×10 ⁰	1.89×10 ⁻¹	1.97×10 ¹	1.61×10 ⁰
DTLZ2	8.08×10 ⁻²	5.86×10 ⁻²	5.49×10 ⁻²	5.70×10 ⁻²	5.32×10⁻²	1.96×10 ⁻¹	7.71×10 ⁻²	7.71×10 ⁻²	5.80×10 ⁻²	8.02×10 ⁻²	8.84×10 ⁻²
DTLZ3	6.00×10⁻²	1.41×10 ²	9.95×10 ⁰	7.05×10 ⁰	9.42×10 ⁰	6.80×10 ¹	1.28×10 ¹	4.70×10 ¹	9.15×10 ⁰	2.06×10 ²	4.93×10 ¹
DTLZ4	1.64×10 ⁻¹	1.67×10 ⁻¹	1.62×10 ⁻¹	5.72×10⁻²	7.13×10 ⁻²	3.55×10 ⁻¹	1.59×10 ⁻¹	1.49×10 ⁻¹	1.14×10 ⁻¹	2.31×10 ⁻¹	5.04×10 ⁻¹
DTLZ5	4.41×10 ⁻³	7.40×10 ⁻³	1.25×10 ⁻²	3.19×10 ⁻²	4.20×10⁻³	4.50×10 ⁻²	1.41×10 ⁻²	1.01×10 ⁻²	5.86×10 ⁻³	1.72×10 ⁻²	2.63×10 ⁻²
DTLZ6	4.12×10⁻³	1.74×10 ⁻¹	1.93×10 ⁻²	1.49×10 ⁻¹	4.29×10 ⁻³	4.42×10 ⁻¹	1.43×10 ⁻²	5.36×10 ⁻³	8.77×10 ⁻³	2.14×10 ⁰	7.63×10 ⁻¹
DTLZ7	9.26×10 ⁻²	1.58×10 ⁻¹	1.03×10 ⁻¹	1.22×10 ⁻¹	1.85×10 ⁻¹	1.16×10 ⁰	5.78×10 ⁻¹	7.86×10⁻²	8.09×10 ⁻²	8.53×10 ⁻¹	1.88×10 ⁰
total	8	1	0	1	4	0	0	1	0	0	0
ranking	2.73	4.90	6.17	5.43	4.00	8.63	7.53	4.37	3.93	8.50	9.80

注: 加粗的数据为各算法在同一测试函数上取得的最优结果, 下同。

表 9 MOFA-GD 与 10 种新近算法在 GD 上的实验结果
Table 9 Experimental results of MOFA-GD and ten recent algorithms on GD

测试函数	MOFA-GD	CMOPSO	NSGA-III	RPD-NSGA-II	MSEA	FLEA	MOEA/D-DYTS	MOEA-PSL	CA-MOEA	CFMOFA	MOEA/DVA
ZDT1	1.58×10⁻⁴	2.88×10 ⁻⁴	1.77×10 ⁻³	1.37×10 ⁻³	3.53×10 ⁻⁴	1.77×10 ⁻³	3.58×10 ⁻²	5.39×10 ⁻⁴	1.21×10 ⁻³	8.53×10 ⁻²	3.46×10 ⁻¹
ZDT2	4.65×10⁻⁵	2.10×10 ⁻⁴	2.54×10 ⁻³	1.80×10 ⁻³	7.08×10 ⁻¹	9.21×10 ⁻²	9.34×10 ⁻²	5.65×10 ⁻⁵	1.66×10 ⁻³	1.60×10 ⁻¹	6.96×10 ⁻¹
ZDT3	1.56×10⁻⁴	2.11×10 ⁻⁴	1.15×10 ⁻³	1.01×10 ⁻³	3.13×10 ⁻⁴	7.18×10 ⁻³	4.52×10 ⁻²	4.75×10 ⁻⁴	9.22×10 ⁻⁴	1.14×10 ⁻¹	3.27×10 ⁻¹
ZDT4	5.26×10⁻⁴	5.70×10 ⁰	1.02×10 ⁻¹	3.80×10 ⁻²	7.52×10 ⁰	5.24×10 ⁰	7.02×10 ⁰	6.32×10 ⁻⁴	6.76×10 ⁻²	2.25×10 ¹	8.35×10 ⁰
ZDT6	3.32×10 ⁻²	1.91×10 ⁻²	3.51×10 ⁻²	7.22×10⁻²	3.08×10 ⁻²	2.79×10 ⁻¹	2.54×10 ⁻²	3.04×10 ⁻²	3.17×10 ⁻²	1.54×10 ⁰	1.36×10 ⁰
Viennet1	1.44×10 ⁻²	1.47×10 ⁻²	1.34×10 ⁻²	1.37×10 ⁻²	8.84×10 ⁻³	1.53×10 ⁻²	6.47×10⁻³	1.12×10 ⁻²	1.29×10 ⁻²	1.52×10 ⁻²	1.76×10 ⁻²
Viennet2	1.86×10 ⁻³	1.66×10 ⁻³	1.57×10 ⁻³	1.39×10 ⁻³	1.60×10 ⁻³	2.12×10 ⁻³	9.74×10⁻⁴	1.96×10 ⁻³	1.63×10 ⁻³	1.70×10 ⁻³	8.66×10 ⁻³
Viennet3	3.77×10 ⁻³	3.50×10 ⁻³	3.38×10 ⁻³	5.10×10 ⁻³	3.30×10 ⁻³	3.63×10 ⁻³	4.70×10 ⁻³	3.23×10 ⁻³	3.60×10 ⁻³	2.86×10⁻³	9.14×10 ⁻³
DTLZ1	1.70×10⁻³	4.73×10 ⁰	3.04×10 ⁻²	1.89×10 ⁻²	1.26×10 ⁻²	3.91×10 ⁰	2.17×10 ⁰	3.60×10 ⁻¹	2.92×10 ⁻²	8.05×10 ⁰	2.34×10 ⁰
DTLZ2	9.41×10 ⁻³	2.25×10 ⁻³	6.04×10 ⁻⁴	6.90×10 ⁻⁴	5.27×10⁻⁴	2.77×10 ⁻²	1.56×10 ⁻³	2.99×10 ⁻³	1.19×10 ⁻³	1.16×10 ⁻²	8.06×10 ⁻³
DTLZ3	1.76×10⁻³	4.70×10 ¹	1.88×10 ⁰	2.31×10 ⁰	2.01×10 ⁰	2.04×10 ¹	1.31×10 ¹	1.31×10 ¹	2.23×10 ⁰	7.29×10 ¹	2.25×10 ¹
DTLZ4	7.36×10 ⁻³	2.86×10 ⁻³	5.88×10 ⁻⁴	7.01×10 ⁻⁴	5.55×10⁻⁴	3.87×10 ⁻²	1.36×10 ⁻³	2.51×10 ⁻³	1.10×10 ⁻³	1.20×10 ⁻²	1.64×10 ⁻²
DTLZ5	1.40×10⁻⁴	9.04×10 ⁻⁴	3.43×10 ⁻⁴	9.40×10 ⁻⁴	9.43×10 ⁻⁴	1.89×10 ⁻²	3.19×10 ⁻⁴	9.43×10 ⁻⁴	3.76×10 ⁻⁴	2.35×10 ⁻³	6.32×10 ⁻³
DTLZ6	4.79×10⁻⁶	2.32×10 ⁻²	6.65×10 ⁻⁶	2.14×10 ⁻³	7.60×10 ⁻⁶	1.26×10 ⁻¹	4.78×10 ⁻⁶	4.81×10 ⁻⁶	1.80×10 ⁻³	7.48×10 ⁻¹	6.76×10 ⁻¹
DTLZ7	4.05×10 ⁻³	4.21×10 ⁻³	5.61×10 ⁻³	3.68×10 ⁻³	1.63×10⁻³	1.28×10 ⁻²	1.49×10 ⁻¹	2.96×10 ⁻³	5.60×10 ⁻³	1.57×10 ⁻¹	8.36×10 ⁻¹
total	8	0	0	1	3	1	2	0	0	1	0
ranking	3.93	5.67	4.63	5.07	3.97	8.77	5.50	4.47	4.67	9.27	10.07

表 10 MOFA-GD 与 10 种新近算法在 MS 上的实验结果
Table 10 Experimental results of MOFA-GD and ten recent algorithms on MS

测试函数	MOFA-GD	CMOPSO	NSGA-III	RPD-NSGA-II	MSEA	FLEA	MOEA/D-DYTS	MOEA-PSL	CA-MOEA	CFMOFA	MOEA/DVA
ZDT1	1.00×10^0	9.99×10^{-1}	9.87×10^{-1}	9.96×10^{-1}	9.99×10^{-1}	1.00×10^0	8.37×10^{-1}	1.00×10^0	9.96×10^{-1}	8.01×10^{-1}	7.11×10^{-1}
ZDT2	1.00×10^0	9.99×10^{-1}	6.97×10^{-1}	9.87×10^{-1}	6.60×10^{-1}	2.49×10^{-1}	4.96×10^{-1}	1.00×10^0	9.76×10^{-1}	7.14×10^{-1}	7.07×10^{-1}
ZDT3	1.00×10^0	9.97×10^{-1}	9.91×10^{-1}	9.93×10^{-1}	9.99×10^{-1}	5.85×10^{-1}	8.22×10^{-1}	1.00×10^0	9.93×10^{-1}	8.14×10^{-1}	7.41×10^{-1}
ZDT4	1.00×10^0	1.03×10^{-1}	4.81×10^{-1}	5.03×10^{-1}	4.06×10^{-1}	2.03×10^{-1}	4.83×10^{-1}	1.00×10^0	5.06×10^{-1}	6.66×10^{-1}	7.07×10^{-1}
ZDT6	1.00×10^0	1.00×10^0	8.42×10^{-1}	9.52×10^{-1}	8.58×10^{-1}	1.00×10^0	1.00×10^0	1.00×10^0	8.91×10^{-1}	7.07×10^{-1}	8.89×10^{-1}
Viennet1	9.91×10^{-1}	1.00×10^0	8.57×10^{-1}	9.87×10^{-1}	9.93×10^{-1}	9.90×10^{-1}	9.99×10^{-1}	9.97×10^{-1}	9.99×10^{-1}	9.97×10^{-1}	8.77×10^{-1}
Viennet2	9.98×10^{-1}	9.97×10^{-1}	5.76×10^{-1}	9.98×10^{-1}	9.97×10^{-1}	8.77×10^{-1}	9.98×10^{-1}	9.87×10^{-1}	9.88×10^{-1}	9.73×10^{-1}	8.78×10^{-1}
Viennet3	9.99×10^{-1}	9.96×10^{-1}	7.84×10^{-1}	1.00×10^0	9.97×10^{-1}	1.00×10^0	9.59×10^{-1}	9.66×10^{-1}	9.96×10^{-1}	9.97×10^{-1}	9.18×10^{-1}
DTLZ1	9.95×10^{-1}	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	9.27×10^{-1}	1.00×10^0	1.00×10^0	1.00×10^0
DTLZ2	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0
DTLZ3	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0
DTLZ4	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	8.84×10^{-1}
DTLZ5	1.00×10^0	1.00×10^0	9.96×10^{-1}	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0
DTLZ6	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0
DTLZ7	9.99×10^{-1}	9.86×10^{-1}	9.80×10^{-1}	9.87×10^{-1}	8.00×10^{-1}	1.77×10^{-1}	9.10×10^{-1}	8.65×10^{-1}	9.84×10^{-1}	8.42×10^{-1}	8.18×10^{-1}
total	12	8	5	7	6	8	6	10	6	6	4
ranking	7.90	7.03	4.10	6.90	5.77	5.23	5.97	6.93	6.53	5.40	4.23

根据表 8, 从综合性指标 IGD 来看, MOFA-GD 在 15 个测试函数上取得了 8 次最优, MSEA 取得了 4 次最优, CMOPSO、RPD-NSGA-II、MOEA-PSL 均取得 1 次最优, 其余算法没有取得最优, 说明 MOFA-GD 相比新近多目标优化算法仍具有优良的表现。值得注意的是在 ZDT4、DTLZ1 和 DTLZ3 测试函数上, MOFA-GD 的优化结果比较其余算法的优化结果优一个数量级, 原因是 ZDT4、DTLZ1 和 DTLZ3 测试函数具有多模态特征, 种群在迭代过程中极易过早收敛到某个局部最优解, 而 MOFA-GD 采用了分组优化模型, 其中变异算子优化多样性变量组使个体充分探索决策空间, 避免个体陷入局部最优。从表 9~10 可知, MOFA-GD 在收敛性和多样性方面均有优良表现, 在 15 个测试函数上, GD 指标结果取得 8 次最优, 在所有对比算法中取得最优次数最多; MS 指标上取得 12 次最优, 其中 11 个测试函数上覆盖程度为 1。从表 8~10 给出的 Friedman 秩检验结果可知, MOFA-GD 在所有对比算法中排名第一, 由此可见 MOFA-GD 具有较好的算法稳定性和可靠性。值得注意的是, 算法 MOEA/DVA 同样采

用对变量进行分组优化, 在大部分测试函数上收敛性和分布性均优于 MOEA/DVA, 原因是 MOFA-GD 对分组变量采用了适应其特征的策略进行优化: 学习行为优化距离变量, 使个体快速收敛到真实 Pareto 解集附近, 有效加快种群收敛速度; 变异算子优化位置变量, 使个体充分探索决策空间, 极大地提升种群的多样性, 进而使算法具有较好的性能。

4 盘式制动器的优化设计问题

为验证算法求解实际工程 MOP 的性能, 本文将 MOFA-GD 应用于优化盘式制动器设计问题。盘式制动器的优化设计是实际工程中经典的结构设计模式, 最早由 Ray 等^[38]对盘式制动器进行了设计与分析, 2020 年 Tanabe 等^[39]将其归纳为实际工程的多目标优化问题并公开了数据集。该案例以制动盘内半径为决策变量 x_1 、制动盘外半径为决策变量 x_2 、制动力为决策变量 x_3 和摩擦曲面个数为决策变量 x_4 组成决策向量 $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]$, 以最小制动器质量 $f_1(x)$ 和最短制动时间 $f_2(x)$ 为优化目标, 建立数学模型, 其表达式为

$$\min F(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}) = 4.9 \times 10^{-5}(x_2^2 - x_1^2)(x_4 - 1) \\ f_2(\mathbf{x}) = \frac{9.82 \times 10^6(x_2^2 - x_1^2)}{x_3 x_4 (x_2^3 - x_1^3)} \end{cases}$$

其约束函数为

$$\begin{aligned} g_1(\mathbf{x}) &= (x_2 - x_1) - 20 \geq 0 \\ g_2(\mathbf{x}) &= 30 - 2.5(x_4 + 1) \geq 0 \\ g_3(\mathbf{x}) &= 0.4 - \frac{x_3}{3.14(x_2^2 - x_1^2)} \geq 0 \\ g_4(\mathbf{x}) &= 1 - \frac{2.22 \times 10^{-3} x_3 x_4 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)^2} \geq 0 \\ g_5(\mathbf{x}) &= \frac{2.66 \times 10^{-2} x_3 x_4 (x_2^3 - x_1^3)}{x_2^2 - x_1^2} - 900 \geq 0 \end{aligned}$$

决策变量的取值范围为： $55 \leq x_1 \leq 80, 75 \leq x_2 \leq 110, 1000 \leq x_3 \leq 3000, 2 \leq x_4 \leq 20$ 。本文所提算法采用可行性法则约束处理机制^[40]对算法进

行处理，使 MOFA-GD 有效处理约束。

为对比 MOFA-GD 的优化效果，选择 5 种近几年求解约束多目标优化问题的算法进行仿真对比实验，5 种算法分别为求解大规模多目标优化问题的改进 Pareto 建模优化算法 (AGEMOEA-II)^[41]、求解约束多目标优化问题的竞争与协作群智能算法 (CMOCSO)^[42]、基于问题转换处理的约束多目标优化算法 (DCNSGA-III)^[43]、基于演进多任务的约束多目标优化算法 (EMCMO)^[44] 和基于新型协作机制的约束多种群进化算法 (MCCMO)^[45]，算法参数均取自 PLATEMO 的默认参数，实验设置与 3.3 保持一致，以 IGD 综合性指标为评价指标。为确保实验的公平，评价指标所得数据为各算法独立运行 50 次取平均值 (mean) 和标准差 (std)，结果由表 11 给出。

表 11 MOFA-GD 与 5 种算法基于 IGD 指标上的实验结果
Table 11 Experimental results of MOFA-GD and five algorithms on IGD

指标	MOFA-GD	AGEMOEA-II	CMOCSO	DCNSGA-III	EMCMO	MCCMO
mean	3.1203×10^{-1}	6.8582×10^{-1}	2.1691×10^0	4.6271×10^0	7.6959×10^{-1}	8.4755×10^{-1}
std	7.2206×10^{-2}	5.3153×10^{-1}	8.7031×10^{-1}	5.5657×10^{-1}	6.7288×10^{-1}	6.9658×10^{-1}

根据表 11 给出的 IGD 平均值来看，相比其他算法，MOFA-GD 面对此问题表现出优良的性能，其 IGD 平均值取得了最优。图 4 给出了各算法在盘式制动器优化问题上获得的 Pareto 最优前沿，可以从图 4 中清楚地看到，在收敛性和分

布性上，MOFA-GD 均优于其他算法，计算得到的 Pareto 前沿均匀分布在真实 Pareto 前沿上。由此可见，MOFA-GD 在众多算法中展现出卓越的性能，是一个解决实际工程优化问题的可行选择。

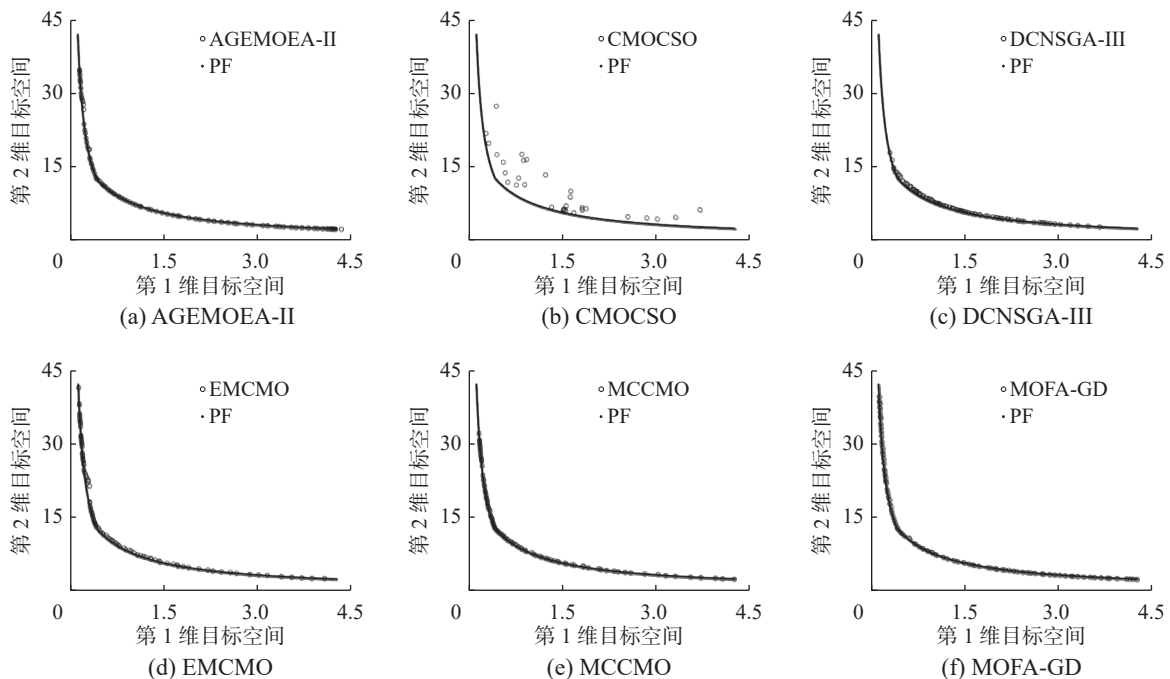


图 4 MOFA-GD 与 5 种对比算法在盘式制动器优化问题上获得的 Pareto 前沿

Fig. 4 Pareto front obtained by MOFA-GD and 5 comparison algorithms for disc brake optimization

5 结束语

MOFA 采用整体维度更新策略,未能充分考虑决策变量对算法的影响,导致某几维变量上优化效果不佳,进而影响算法的收敛性和多样性。针对上述问题,本文提出决策变量分组优化的多目标萤火虫算法。MOFA-GD 引入决策变量分组机制,按照决策变量对算法性能的不同影响,将整体变量分成收敛性变量组和多样性变量组。设计决策变量分组优化模型:学习行为优化收敛性变量组,使萤火虫能够快速靠近最优个体;非均匀变异算子优化多样性变量组,使萤火虫能够克服种群约束,充分探索决策空间,后期逐渐减小的变异幅度和概率使萤火虫精细化探索,不仅提升了算法寻优精度,还增强了算法的稳定性。档案截断策略能够精准删除拥挤的个体,使萤火虫均匀分布在 Pareto 前沿。在 15 个广泛使用的多目标优化测试函数上,MOFA-GD 与 6 种经典多目标优化算法和 10 种新近多目标优化算法进行性能对比,通过收敛性指标 GD、分布性指标 MS 和综合性指标 IGD 的性能检测,充分验证 MOFA-GD 具有优秀的收敛性和分布性,是一种求解多目标优化问题的可靠选择。最后将 MOFA-GD 应用到优化盘式制动器设计上,与 5 种约束多目标优化算相比展现出卓越的性能。鉴于 MOFA-GD 的有效性和稳定性,在今后的研究方向有望进一步扩大 MOFA-GD 的应用范围,以应对更多样化的工程优化问题。

参考文献:

- [1] FARAG M A, EL-SHORBAGY M A, MOUSA A A, et al. A new hybrid metaheuristic algorithm for multiobjective optimization problems[J]. *International journal of computational intelligence systems*, 2020, 13(1): 920–940.
- [2] BAROCIO E, REGALADO J, CUEVAS E, et al. Modified bio-inspired optimisation algorithm with a centroid decision making approach for solving a multi-objective optimal power flow problem[J]. *IET generation, transmission & distribution*, 2017, 11(4): 1012–1022.
- [3] 李昌响, 赵嘉, 韩龙哲, 等. 多通道 CNN-BiLSTM 的短时温度预测[J]. *江西师范大学学报(自然科学版)*, 2023, 47(3): 325–330.
LI Changxiang, ZHAO Jia, HAN Longzhe, et al. The short-time temperature prediction for multi-channel CNN-BiLSTM[J]. *Journal of Jiangxi Normal University (natural science edition)*, 2023, 47(3): 325–330.
- [4] 李松, 吴润秀, 康平, 等. 基于自适应剪辑与概率参数的 Tri-Training 算法[J]. *江西师范大学学报(自然科学版)*, 2023, 47(5): 490–496.
LI Song, WU Runxiu, KANG Ping, et al. The ADP-Tri-Training: Tri-Training with adaptive editing and probability parameters[J]. *Journal of Jiangxi Normal University (natural science edition)*, 2023, 47(5): 490–496.
- [5] LIAGKOURAS K, METAXIOTIS K. Enhancing the performance of MOEAs: an experimental presentation of a new fitness guided mutation operator[J]. *Journal of experimental & theoretical artificial intelligence*, 2017, 29(1): 91–131.
- [6] XIAO Renbin. Four development stages of collective intelligence[J]. *Frontiers of information technology & electronic engineering*, 2024, 25(7): 903–916.
- [7] MIRJALILI S, SAREMI S, MIRJALILI S M, et al. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization[J]. *Expert systems with applications*, 2016, 47: 106–119.
- [8] AKBARI R, HEDAYATZADEH R, ZIARATI K, et al. A multi-objective artificial bee colony algorithm[J]. *Swarm and evolutionary computation*, 2012, 2: 39–52.
- [9] YANG Xinse. Nature-inspired metaheuristic algorithms[M]. Frome: Luniver press, 2008.
- [10] YANG Xinshe. Multiobjective firefly algorithm for continuous optimization[J]. *Engineering with computers*, 2013, 29(2): 175–184.
- [11] 谢承旺, 肖驰, 丁立新, 等. HMOFA: 一种混合型多目标萤火虫算法[J]. *软件学报*, 2018, 29(4): 1143–1162.
XIE Chengwang, XIAO Chi, DING Lixin, et al. HMOFA: a hybrid multi-objective firefly algorithm[J]. *Journal of software*, 2018, 29(4): 1143–1162.
- [12] ZHAO Jia, CHEN Dandan, XIAO Renbin, et al. Multi-strategy ensemble firefly algorithm with equilibrium of convergence and diversity[J]. *Applied soft computing*, 2022, 123: 108938.
- [13] 赵嘉, 胡秋敏, 肖人彬, 等. 求解大规模稀疏优化问题的高维多目标萤火虫算法[J]. *控制与决策*, 2024, 39(12): 3989–3996.
ZHAO Jia, HU Qiumin, XIAO Renbin, et al. Many-objective firefly algorithm for solving large-scale sparse optimization problems[J]. *Control and decision*, 2024, 39(12): 3989–3996.
- [14] LYU Li, ZHAO Jia, WANG Jiayuan, et al. Multi-objective firefly algorithm based on compensation factor and elite learning[J]. *Future generation computer systems*, 2019, 91: 37–47.
- [15] 赵嘉, 陈丹丹, 肖人彬, 等. 一种基于最大最小策略和非均匀变异的萤火虫算法[J]. *智能系统学报*, 2022, 17(1):

- 116–130.
ZHAO Jia, CHEN Dandan, XIAO Renbin, et al. A heterogeneous variation firefly algorithm with maximin strategy[J]. *CAAI transactions on intelligent systems*, 2022, 17(1): 116–130.
- [16] 陈娟, 赵嘉, 肖人彬, 等. 基于动态反向学习和莱维飞行的双搜索模式萤火虫算法[J]. *信息与控制*, 2023, 52(5): 607–615.
CHEN Juan, ZHAO Jiao, XIAO Renbin, et al. Double search mode firefly algorithm based on dynamic reverse learning and levy flight[J]. *Information and control*, 2023, 52(5): 607–615.
- [17] ZHAO Jia, CHEN Dandan, XIAO Renbin, et al. Multi-objective firefly algorithm with adaptive region division[J]. *Applied soft computing*, 2023, 147: 110796.
- [18] WANG Hui, WANG Wenjun, CUI Laizhong, et al. A hybrid multi-objective firefly algorithm for big data optimization[J]. *Applied soft computing*, 2018, 69: 806–815.
- [19] ZHAO Jia, LV Siyu, XIAO Renbin, et al. Hierarchical learning multi-objective firefly algorithm for high-dimensional feature selection[J]. *Applied soft computing*, 2024, 165: 112042.
- [20] MA Xiaoliang, LIU Fang, QI Yutao, et al. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables[J]. *IEEE transactions on evolutionary computation*, 2016, 20(2): 275–298.
- [21] ZHAO Xinchao, GAO Xiaoshan, HU Zechun. Evolutionary programming based on non-uniform mutation[J]. *Applied mathematics and computation*, 2007, 192(1): 1–11.
- [22] ZITZLER E, LAUMANN M, THIELE L. SPEA2: improving the strength pareto evolutionary algorithm for multi-objective optimization [C]//Proceedings of the Evolutionary Methods for Design, Optimization and Control. Barcelona: IEEE, 2002: 19–26.
- [23] 赵嘉, 谢智峰, 吕莉, 等. 深度学习萤火虫算法[J]. *电子学报*, 2018, 46(11): 2633–2641.
ZHAO Jia, XIE Zhifeng, LV Li, et al. Firefly algorithm with deep learning[J]. *Acta electronica sinica*, 2018, 46(11): 2633–2641.
- [24] ZHAO Jia, CHEN Wenping, XIAO Renbin, et al. Firefly algorithm with division of roles for complex optimal scheduling[J]. *Frontiers of information technology & electronic engineering*, 2021, 22(10): 1311–1333.
- [25] TIAN Ye, CHENG Ran, ZHANG Xingyi, et al. PlatEMO: a MATLAB platform for evolutionary multi-objective optimization educational forum[J]. *IEEE computational intelligence magazine*, 2017, 12(4): 73–87.
- [26] RAQUEL C R, NAVAL P C. An effective use of crowding distance in multiobjective particle swarm optimization[C]//Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation. Washington: ACM, 2005: 257–264.
- [27] ZHANG Qingfu, LI Hui. MOEA/D: a multiobjective evolutionary algorithm based on decomposition[J]. *IEEE transactions on evolutionary computation*, 2007, 11(6): 712–731.
- [28] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE transactions on evolutionary computation*, 2002, 6(2): 182–197.
- [29] CORNE D W, JERRAM N R, KNOWLES J D, et al. PESA-II: region-based selection in evolutionary multiobjective optimization[C]//Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. [S. l. : s. n.], 2001: 283–290.
- [30] ZHANG Xingyi, ZHENG Xiutao, CHENG Ran, et al. A competitive mechanism based multi-objective particle swarm optimizer with fast convergence[J]. *Information sciences*, 2018, 427: 63–76.
- [31] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints[J]. *IEEE transactions on evolutionary computation*, 2014, 18(4): 577–601.
- [32] ELARBI M, BECHIKH S, GUPTA A, et al. A new decomposition-based NSGA-II for many-objective optimization[J]. *IEEE transactions on systems, man, and cybernetics: systems*, 2018, 48(7): 1191–1210.
- [33] TIAN Ye, HE Cheng, CHENG Ran, et al. A multistage evolutionary algorithm for better diversity preservation in multiobjective optimization[J]. *IEEE transactions on systems, man, and cybernetics: systems*, 2021, 51(9): 5880–5894.
- [34] LI Lianghao, HE Cheng, CHENG Ran, et al. A fast sampling based evolutionary algorithm for million-dimensional multiobjective optimization[J]. *Swarm and evolutionary computation*, 2022, 75: 101181.
- [35] BACK T, PREUSS M, DEUTZ A, et al. Parallel problem solving from nature-PPSN XVI[M]. Cham: Springer International Publishing, 2020: 271–284.
- [36] TIAN Ye, LU Chang, ZHANG Xingyi, et al. Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks[J]. *IEEE transactions on cybernetics*, 2021, 51(6): 3115–3128.
- [37] HUA Yicun, JIN Yaochu, HAO Kuangrong. A clustering-based adaptive evolutionary algorithm for multiobjective

- optimization with irregular Pareto fronts[J]. *IEEE transactions on cybernetics*, 2019, 49(7): 2758–2770.
- [38] RAY T, LIEW K M. A swarm metaphor for multiobjective design optimization[J]. *Engineering optimization*, 2002, 34(2): 141–153.
- [39] TANABE R, ISHIBUCHI H. An easy-to-use real-world multi-objective optimization problem suite[J]. *Applied soft computing*, 2020, 89: 106078.
- [40] DEB K. An efficient constraint handling method for genetic algorithms[J]. *Computer methods in applied mechanics and engineering*, 2000, 186(2/3/4): 311–338.
- [41] PANICHELLA A. An improved Pareto front modeling algorithm for large-scale many-objective optimization[C]// *Proceedings of the Genetic and Evolutionary Computation Conference*. Boston: ACM, 2022: 565–573.
- [42] MING Fei, GONG Wenyin, LI Dongcheng, et al. A competitive and cooperative swarm optimizer for constrained multiobjective optimization problems[J]. *IEEE transactions on evolutionary computation*, 2023, 27(5): 1313–1326.
- [43] JIAO Ruwang, ZENG Sanyou, LI Changhe, et al. Handling constrained many-objective optimization problems via problem transformation[J]. *IEEE transactions on cybernetics*, 2021, 51(10): 4834–4847.
- [44] QIAO Kangjia, YU Kunjie, QU Boyang, et al. An evolutionary multitasking optimization framework for constrained multiobjective optimization problems[J]. *IEEE transactions on evolutionary computation*, 2022, 26(2): 263–277.
- [45] ZOU Juan, SUN Ruiqing, LIU Yuan, et al. A multipopulation evolutionary algorithm using new cooperative mechanism for solving multiobjective problems with multiconstraint[J]. *IEEE transactions on evolutionary computation*, 2024, 28(1): 267–280.

作者简介:



邢文来, 硕士研究生, 主要研究方向为群智能算法。E-mail: xw19464@yeah.net。



吴润秀, 教授, 主要研究方向为群智能算法及应用、大数据与人工智能。发表学术论文 30 余篇。E-mail: wrx@nit.edu.cn。



肖人彬, 教授, 博士生导师, 主要研究方向为群体智能、涌现计算、复杂系统建模与仿真。主持国家自然科学基金项目 11 项, 获教育部自然科学基金 1 项、湖北省自然科学基金及科技进步奖 4 项。发表学术论文 300 余篇, 出版学术专著和教材 10 余部。E-mail: rbxiao@hust.edu.cn。

2025 全国高校人工智能通识与基础课程建设研讨会

为探讨人工智能与高等教育深度融合的路径, 加快人工智能的理念、知识、方法和技术更快融入高等教育的专业建设, 推动学科创新与人才培养, 由中国人工智能学会(CAAI)主办, CAAI 人工智能基础专委会、北京邮电大学人工智能学院、北京邮电大学出版社联合承办的“走进高校”系列活动将于 2025 年 8 月 7—9 日在北京邮电大学举行。本次活动主题为“全国高校人工智能通识与基础课程建设”研讨会。研讨会将通过人工智能专家分享最新研究成果、一线教师分享教学经验的方式, 探讨我国人工智能通识与基础课程建设的途径及最新发展趋势。

时间地点

会议时间: 2025 年 8 月 7 日报到, 8—9 日会议论坛。会议地点: 北京邮电大学

联系方式

中国人工智能学会 陈志成 18612752025
北京邮电大学人工智能学院 张 旭 17329570436
北京邮电大学出版社 张 超 18601092338

官网详情

<https://www.caa.cn/index.php?s=/Home/Article/detail/id/4750.html>