

一种基于分解和协同的高维多目标进化算法*

谢承旺¹, 余伟伟², 闭应洲¹, 汪慎文³, 胡玉荣⁴



¹(南宁师范大学 计算机与信息工程学院, 广西 南宁 530299)

²(北京工业大学 软件学院, 北京 100124)

³(河北地质大学 信息工程学院, 河北 石家庄 050031)

⁴(荆楚理工学院 科技处, 湖北 荆门 448000)

通讯作者: 谢承旺, E-mail: chengwangxie@163.com; 闭应洲, E-mail: byzhou@163.com

摘要: 现实中大量存在的高维多目标优化问题对以往高效的多目标进化算法提出了严峻的挑战. 通过将分解策略和协同策略相结合提出一种高维多目标进化算法 MaOEA/DCE. 该算法利用混合水平正交实验方法在聚合系数空间产生一组均匀分布的权重向量以改善初始种群的分布性; 其次, 算法将差分进化算子和自适应 SBX 算子进行协同进化, 以产生高质量的子代个体, 并改善算法的收敛性. 该算法与另外 5 种高性能的多目标进化算法在基准测试函数集 DTLZ{1,2,4,5} 上进行对比实验, 利用改进的反转世代距离指标 IGD+ 评估各算法的性能. 实验结果表明, MaOEA/DCE 算法与其他对比算法相比, 在总体上具有较为显著的收敛性和分布性优势.

关键词: 高维多目标优化; 分解策略; 混合水平正交实验设计; 高维多目标进化算法

中图法分类号: TP181

中文引用格式: 谢承旺, 余伟伟, 闭应洲, 汪慎文, 胡玉荣. 一种基于分解和协同的高维多目标进化算法. 软件学报, 2020, 31(2): 356-373. <http://www.jos.org.cn/1000-9825/5617.htm>

英文引用格式: Xie CW, Yu WW, Bi YZ, Wang SW, Hu YR. Many-objective evolutionary algorithm based on decomposition and coevolution. Ruan Jian Xue Bao/Journal of Software, 2020, 31(2): 356-373 (in Chinese). <http://www.jos.org.cn/1000-9825/5617.htm>

Many-objective Evolutionary Algorithm Based on Decomposition and Coevolution

XIE Cheng-Wang¹, YU Wei-Wei², BI Ying-Zhou¹, WANG Shen-Wen³, HU Yu-Rong⁴

¹(School of Computer and Information Engineering, Nanning Normal University, Nanning 530299, China)

²(School of Software Engineering, Beijing University of Technology, Beijing 100124, China)

³(School of Information Engineering, Hebei Geo University, Shijiazhuang 050031, China)

⁴(Department of Science and Technology, Jingchu University of Technology, Jingmen 448000, China)

Abstract: In real-world, there exist lots of many-objective optimization problems (MaOPs), which severely challenge well-known multi-objective evolutionary algorithms (MOEAs). A many-objective evolutionary algorithm combining decomposition and coevolution

* 基金项目: 国家自然科学基金(61763010, 61402481, 61165004); 广西八桂学者项目; 河北青年拔尖人才支持计划(冀字[2013]17); 河北省自然科学基金(F2015403046); 河北省教育厅科技重点项目(ZD2018083); 湖北省教育厅科研项目(B2015240); 荆楚理工学院科学研究重点基金(ZR201402); 荆楚理工学院科学研究引进人才科研启动金(QDB201605)

Foundation item: National Natural Science Foundation of China (61763010, 61402481, 61165004); "BAGUI Scholar" Program of Guangxi Zhuang Autonomous Region of China; Hebei Youth Top Talent Support Program (冀字[2013]17); Natural Science Foundation of Hebei Province(F2015403046); Major Scientific Research Program of Education Bureau of Hebei Province (DZ2018083); Science and Technology Project of Education Bureau of Hubei Province (B2015240); Major Scientific Research Fund of Jingchu University of Technology (ZR201402); Startup Fund for Talents of Jingchu University of Technology (QDB201605)

收稿时间: 2018-04-01; 修改时间: 2018-05-25; 采用时间: 2018-06-18; jos 在线出版时间: 2019-01-21

CNKI 网络优先出版: 2019-01-22 13:49:08, <http://kns.cnki.net/kcms/detail/11.2560.TP.20190122.1348.012.html>

(MaOEA/DCE) is presented in this paper. MaOEA/DCE adopts mix-level orthogonal experimental design to produce a set of weight vectors evenly distributed in weight coefficient space, so as to improve the diversity of initial population. In addition, the MaOEA/DCE integrates differential evolution (DE) with the adaptive SBX operator to generate high-quality offspring for enhancing the convergence of evolutionary population. Some comparative experiments are conducted among MaOEA/DCE and other five representative MOEAs to examine their IGD+ performance on four MaOPs of DTLZ{1,2,4,5}. The experimental results show that the proposed MaOEA/DCE has overall performance advantage over the other peering MOEAs in terms of convergence, diversity, and robustness.

Key words: many-objective optimization; decomposition strategy; mix-level orthogonal experimental design; many-objective evolutionary algorithm

现实中很多优化问题需要同时优化多个目标,这类问题通常被称为多目标优化问题(multi-objective optimization problem,简称 MOP).MOP 问题的特征在于多个目标函数之间相互冲突,而且一般不存在唯一的解能够同时满足多个最优化的目标函数.因此,MOP 问题的求解结果通常是一组折中的解,即 Pareto 解集^[1].鉴于 MOP 问题的复杂性,使得传统的数学解析方法难以处理,而进化算法(evolutionary algorithm,简称 EA)是一类通过模拟生物进化机制发展而来的基于群体的随机优化方法.EA 算法具有适于求解 MOP 问题的若干良好特性,因而在多目标优化领域受到广泛关注.迄今为止,研究者基于不同的研究背景和视角发展出了多种多目标进化算法(multi-objective evolutionary algorithm,简称 MOEA),其中,基于 Pareto 最优性的 NSGA-II^[2]和 SPEA2^[3]算法即是其中的经典算法.近年来,一些新型的进化模型陆续引入到多目标优化和高维多目标优化中,包括多目标粒子群算法 MOPSO^[4]、改进速度更新的 SMPSO 算法^[5]、基于分布估计的 RM-MEDA 算法^[6]和多目标烟花爆炸算法等^[7].近几年来,混合型 MOEA 算法相继产生,其中代表性的工作包括 Molina^[8]、Nebro^[9]和 Solima^[10]等人的工作.基于混合策略的 MOEA 算法根据算法或算子的优劣,通过取长补短的做法发展出高效的 MOEA 算法.实践证明,混合型方法是一种很有效的策略.

随着经济与社会的发展,更多个优化目标的优化问题大量涌现,研究者通常把目标数大于等于 4 的优化问题称为高维多目标优化问题(many-objective optimization problem,简称 MaOP).一般而言,MaOP 问题与 MOP 问题相比,具有更加难以求解的特征,主要表现在:1) Pareto 支配关系在高维目标空间者难以区分解个体的优劣,从而严重削弱了种群的进化压力;2) 为了表征 MaOP 问题的 Pareto 前沿,所需使用的非支配解数量依目标数呈指数增加(例如,对于 m 个目标的 MaOP 问题,按每个目标上分布 k 个解来计算,则需要 mk^{m-1} 个解来表示 Pareto 前沿);3) 难以可视化 Pareto 前沿,因为按照人类的认识水平,对 4 维及以上空间其图形特征无法准确地刻画.此外,一旦多样性保持机制在群体进化中占据主导,则有很大可能迟滞种群逼近真实的 Pareto 前沿,从而对进化过程带来较大的负面影响^[11-13].近期的一些研究也表明,当优化问题的目标数目增至 10 个或更多时,基于 Pareto 支配的 MOEA 算法甚至比随机搜索算法表现得更差^[14].

为了应对高维多目标问题带来的挑战,研究者从不同方面开展了研究,概括起来可以将其分成如下几种类型.

- 1) 引入数学分析中的降维思想,将高维多目标问题降为低维目标问题.这类方法假定 MaOP 问题的目标集中存在冗余目标,通过分析目标之间的关系,在尽可能保持解集支配结构的前提下,消除与其他目标不冲突的冗余目标或者组合彼此不冲突的目标.但是这种方法并非适合所有的 MaOP 问题,某些问题并不能很好地保证降维的有效性.
- 2) 基于评估指标的方法.这类方法通过直接优化 Pareto 近似解集的评估指标(例如超体积指标 Hypervolume 等)来优化待解问题.其实质是将 MaOP 问题转化为一个优化指标函数的单目标优化问题.然而,基于指标的方法也存在指标,尤其是超体积指标计算复杂度很高的缺点.
- 3) 利用偏好的方法.在多目标优化中,偏好意味着决策者给目标赋予了不同的值来表示其重要程度或者优先处理的顺序.它使得搜索过程或者优化结果可以处于决策者感兴趣的区域.这一类方法需要用户预先指定偏好或要求与搜索进程进行交互,因而增大了用户使用算法的难度.
- 4) 使用新的支配关系.新的支配关系主要利用目标优劣的计数或者目标之间的差距来区分非劣解,通过

细粒化区分解个体来促进种群进化压力的改善,加快算法收敛.但这些新的支配关系仍存在不满足传递性和/或需要预先设置合适参数的问题.

- 5) 利用分解的方法将 MOP 问题分解成多个子问题,每个子问题通过交换各自解的信息来加快搜索,MOEA/D 算法^[15]则是此类算法的典型范例.MOEA/D 中的分解策略可采用加权和方法、切比雪夫方法和边界交叉法等进行聚合.一般地,聚合的标量优化问题的最优解是原 MOP 问题的一个 Pareto 最优解,而原 MOP 问题的一个 Pareto 最优解也可能是某个聚合标量优化问题的最优解.因此,对 MOP 问题 Pareto 前沿的逼近即可转化成对多个标量优化问题的求解,这也是很多传统数学规划方法逼近 Pareto 前沿的基本思想.

MOEA/D 算法框架的易用性和包容性吸引了多目标优化领域研究的关注.有关 MOEA/D 算法及其变种的研究大致可分为以下几类.

- 1) 将 MOEA/D 算法同其他启发式算法相结合.Li 等人^[16]将 MOEA/D 和模拟退火算法相结合来求解组合优化问题;Moubayed 等人^[17]将 MOEA/D 和粒子群优化算法相结合求解连续优化问题;Ke 等人^[18]将 MOEA/D 和蚁群算法相结合求解多目标 0-1 背包问题和 2 目标旅行售货商问题,取得了较好的效果.
- 2) 将新的分解机制融入到 MOEA/D 框架.Zhang 等人^[19]提出一种 NBI 型切比雪夫分解方法求解具有不同比例的 MOP 问题;Hisao 等人^[20]建议同时使用不同的聚合方法来构建标量优化问题;Wang 等人^[21]提出对分解子问题施以一定的约束,并适应性地调整约束,以平衡算法的收敛性和多样性.
- 3) 新的权重向量方法.Tan 等人^[22]提出利用均匀设计方法设置聚合系数,以求解高维多目标优化问题;Qi 等人^[23]提出一种周期性调整权重向量的 MOEA/D 算法,其通过对切比雪夫聚合函数做几何分析,进而提出一种新的聚合系数初始化技术,并根据复杂 Pareto 前沿的特点,自适应调整权重向量来改善解的均匀性.
- 4) 在 MOEA/D 中加入新的重组和/或变异算子.Li 等人^[24]将差分进化算法引入 MOEA/D 中,形成了 MOEA/D-DE 算法.实验结果表明,MOEA/D-DE 算法的性能显著地优于 NSGA-II.Zhou 等人^[25]利用高斯概率模型的重组方法对种群进行建模和采样,实验结果表明,该方法对具有复杂 Pareto 前沿的 MOP 问题有效.文献[26]进一步提出了结合混合高斯模型的 MOEA/D 算法.该方法通过计算邻域个体的相似性来重用模型,既保证了模型的质量,也降低了建模的次数,改善了算法的效率.Chen 等人^[27]通过利用引导的变异算子代替差分进化算子以及使用优先级队列的更新机制来增强 MOEA/D 算法的性能.
- 5) 设计 MOEA/D 的并行版本.Nebro 等人^[28]研究了多目标元启发式 MOEA/D 算法的并行化.文献[29]在 GPU 上实现了 MOEA/D-ACO 算法,以求解多目标旅行售货商问题.

综上,基于分解的 MOEA/D 算法不依赖于 Pareto 支配,避免了使用 Pareto 支配关系所存在的缺陷.另外,MOEA/D 算法使用聚合系数将 MOP 问题的各目标聚合成标量优化的子问题,因而优化目标数目增多不会加大 MOEA/D 算法解题难度.因此,基于分解的 MOEA/D 算法具有解决高维多目标优化问题的潜力.

综上所述,本文根据 MaOP 问题的特点,将分解策略和协同进化策略相结合提出 MaOEA/DCE(many-objective evolutionary algorithm based on decomposition and coevolution)算法.MaOEA/DCE 算法的特点在于:1) 利用混合水平正交设计方法产生合适规模且均匀分布的初始种群;2) 将差分进化算子和自适应 SBX 算子实施协同进化以生成子代个体,提高算法的收敛性.这两种策略在种群的进化过程中有效地协同,以改善算法的总体性能.

本文第 1 节是 MaOP 问题的基本知识.第 2 节是 MaOEA/DCE 算法的设计细节和流程.第 3 节是实验部分.第 4 节是本文的结论.

1 基本知识

根据对偶原理,以最小化问题为例,可以将一个 MOP 问题形式化地描述如下^[30].

$$\begin{cases} \min y = F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ x = (x_1, x_2, \dots, x_n) \in X \subset R^n \\ y = (y_1, y_2, \dots, y_m) \in Y \subset R^m \end{cases} \quad (1)$$

其中, n 是决策变量的数目, x 是 n 维的决策向量, X 表示决策空间; m 是 MOP 问题的目标个数, y 是 m 维的目标向量, Y 表示目标空间, F 表示需要同时优化 m 个目标的映射函数. 通常, 当 $m \geq 4$ 时, 式(1)被称为高维多目标优化问题. 需要说明的是, 这里仅考虑连续优化问题, 离散型优化问题不在本文的讨论范围之内.

以下是多目标优化中几个重要的概念. 1) Pareto 支配: 假设 x_1 和 x_2 是决策空间 X 中的两个解点, 如果 x_1 的目标函数都不大于并且至少存在 1 个小于 x_2 的目标函数, 则称 x_1 Pareto 支配 x_2 , 记为 $x_1 \prec x_2$. 2) Pareto 解集: 设 x^* 为决策空间 X 中的点, 如果 x^* 不被 X 中其他解点 Pareto 支配, 则称 x^* 为非支配解(亦称有效解或非劣解). 3) Pareto 解集: 决策空间 X 中的所有非支配解得集合称为 Pareto 解集(Pareto set, 简称 PS). 4) Pareto 前沿: PS 中所有的点在目标空间 Y 中的映象点的集合构成 Pareto 前沿(Pareto front, 简称 PF).

2 MaOEA/DCE 算法

2.1 混合水平正交实验设计方法

在决策者偏好信息未知的情况下, 构造一组均匀分布的权重向量, 以期获得一组均匀分布于 Pareto 前沿的解, 是 MOEA/D 算法及其变种的常用做法. 设 N 为 MOEA/D 算法中权重向量的数目(亦即种群的规模), m 为 MaOP 问题的目标数, $\lambda^1, \lambda^2, \dots, \lambda^N$ 为 N 个不同的权重向量, 其中, 每个权重向量 $\lambda^i = (\lambda_1^i, \lambda_2^i, \dots, \lambda_m^i)$ 中的分量 λ_j^i 从 $\left\{0, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H}\right\}$ 中不重复取值, 且满足 $\sum_{j=1}^m \lambda_j^i = 1$. 需要指出的是, 权重向量个数 N 是由用户定义的正整数 H 和目标数目 m 确定的, 且满足公式(2).

$$N = C_{H+m-1}^{m-1} \quad (2)$$

由公式(2)可知, 随着目标个数 m 的增大, 种群规模 N 将呈非线性方式增长. 在高维目标空间中, 由于目标数较大, 如果沿用这种权重向量生成方式, 则需要一个规模庞大的种群参与进化过程, 从而导致巨大的计算资源耗费. 表 1 给出了在不同的 (H, m) 取值下, 利用公式(2)产生的种群规模 N 的大小.

Table 1 Population size N derived from H and m

表 1 由 H 和 m 得到的种群规模 N

m	H				
	12	25	149	199	249
2	13	26	150	200	250
3	91	351	11 325	20 100	31 375
4	455	3 276	573 800	1 353 400	2 635 500
10	293 930	52 451 256	1.3408E14	8.1008E12	10212E16

由表 1 可知, MOEA/D 算法产生权重向量的方法并不适合高维目标空间, 需要采用新的权重向量生成策略生成满足用户指定规模且均匀分布于聚合系数空间的权重向量. 这里采用文献[31]中提出的混合水平正交设计方法 MLOA(mixed level orthogonal array)产生接近于使用者期望规模的初始种群. 不妨设 $M = L_{M_1}(Q_1^{N_1}) Q_2^{N_2}$ 表示一个混合水平的正交表, 其中 $L_{M_1}(Q_1^{N_1})$ 和 $L_{M_2}(Q_2^{N_2})$ 分别表示两个相混合的等水平正交表, M_1, M_2 分别表示 $L_{M_1}(Q_1^{N_1})$ 和 $L_{M_2}(Q_2^{N_2})$ 的行数, 而且 $M = M_1 \times M_2$, $M_1 = Q_1^{J_1}$, $M_2 = Q_2^{J_2}$, N_1, N_2 分别表示两个等水平正交表的行数, Q_1, Q_2 则表示它们的划分水平的数目. 如果用 J_1 和 J_2 表示两个等水平正交表的正交指数, 则其满足如下关系的正整数:

$$\begin{cases} \frac{Q_1^{J_1} - 1}{Q_1} \geq N_1 \\ \frac{Q_2^{J_2} - 1}{Q_2} \geq N_2 \end{cases} \quad (3)$$

不难看出, $L_M(Q_1^{N_1} Q_2^{N_2})$ 可以通过调整 $L_{M_1}(Q_1^{N_1})$ 和 $L_{M_2}(Q_2^{N_2})$ 的因素个数和水平划分数目来达到使用者所需要的权重向量的数目^[31]. 其中, 文献[32]给出了创建等水平正交表的过程, 在创建了等水平正交表的基础上, 可采用算法 1 的方法构造混合水平正交表 $L_M(Q_1^{N_1} Q_2^{N_2})$. 根据文献[31]的研究, 本文的 J_1 取值为 2, J_2 将视待解 MaOP 问题的不同取值为 1 或 2.

算法 1. Construction_mixed_level_orthogonal_array.

Input: 两个相混合的等水平正交表的划分水平 Q_1, Q_2 ; 正交指数 J_1, J_2 .

Output: 混合正交矩阵 $(c_{i,j})_{M \times (N_1 + N_2)}$.

1. 利用算法 1 构造等水平正交表 $L_{M_1}(Q_1^{N_1}) = (a_{i,j})_{M_1 \times N_1}$ 和 $L_{M_2}(Q_2^{N_2}) = (b_{i,j})_{M_2 \times N_2}$.
2. **for** ($k=0; k < M_1; k++$)
3. **for** ($i=0; i < M_2; i++$)
4. $c_{(k-1)M_2+i} = [a_k, b_i]$
5. **end**
6. **end**
7. 输出混合水平正交矩阵 $(c_{i,j})_{M \times (N_1 + N_2)}$.

当混合水平正交产生后即可利用算法 2 生成规模为 N 的权重向量集合 $\{\lambda^1, \lambda^2, \dots, \lambda^N\}$, 其中, 每一个权重 $\lambda^i = (\lambda_1^i, \lambda_2^i, \dots, \lambda_m^i)$ 的各分量 λ_j^i ($i=1, 2, \dots, N; j=1, 2, \dots, m$) 需满足 $\sum_{j=1}^m \lambda_j^i = 1$.

算法 2. Generation_weight_vectors_with_MLOA.

Input: 种群规模 N ; 目标数目 m ; 正交指数 J_1, J_2 .

Output: N 个均匀分布的权重向量.

1. 根据用户设置的种群规模 N 和问题的目标数 m 确定相混合的两个等水平正交表的划分水平 Q_1, Q_2 .
2. 基于划分水平 Q_1, Q_2 和正交指数 J_1, J_2 , 利用算法 2 构造混合水平正交表 $L_M(Q_1^{J_1} Q_2^{J_2})$.
3. 对混合水平正交表均匀采样, 得到 N 个均匀分布的权重向量 $\{\lambda^1, \lambda^2, \dots, \lambda^N\}$.
4. 输出 N 个均匀分布的权重向量.

综上, 混合水平正交表可以通过调整两个等水平正交表的关键参数来生成使用者所需要的实验组数, 然后再均匀采样获得 N 个权重向量. 这样, 既得到了合适的权重向量集, 又节省了计算资源.

2.2 分解机制

把一个 MaOP 问题分解为一系列子问题 $\min g^i(x), i=1, 2, \dots, N$, 每个子问题的最优解对应于 MaOP 问题的一个 Pareto 最优解. 其中, 每个子问题的目标函数是各个目标分量的聚合函数, 每个子问题对应一个权重向量 $\lambda^i = (\lambda_1^i, \lambda_2^i, \dots, \lambda_m^i), i=1, 2, \dots, N$. 常用的分解方法有加权和方法、切比雪夫方法和边界相交法. 加权和方法虽然简单, 但对于多目标优化问题的 Pareto 前沿为非凸的情况, 该方法难以找到 Pareto 最优解 (针对最小化问题, 反之亦然). 另外, 加权和方法的权重系数也不易确定, 需要人为的经验或通过实验确定. 边界相交法虽然能够找到均匀的 Pareto 解集合, 但它需要处理等式约束, 需要预先设定惩罚系数 θ 值, 其也需要人为指定或通过实验确定. 对比加权和方法与边界相交法, 切比雪夫方法不仅能够处理 Pareto 前沿为非凸的复杂情形, 而且能够减少人为因素和反复实验的不利方面. 鉴于此, 这里利用切比雪夫方法作为本文的分解机制.

设种群的规模为 N , 目标个数为 m , $A = (\lambda^1, \lambda^2, \dots, \lambda^N)$ 为权重向量集合, 其中, $\lambda^i = (\lambda_1^i, \lambda_2^i, \dots, \lambda_m^i)$ 且满足 $\sum_{j=1}^m \lambda_j^i = 1$,

则第 i 个子问题可表示如下:

$$g^{ie}(x|\lambda^i, z^*) = \min_{1 \leq j \leq m} \{\lambda_j^i | f_j(x) - z_j^* \} \quad (4)$$

其中, $z^* = (z_1^*, z_2^*, \dots, z_m^*)$ 是最优参考点.一般地,最优参考点较难获得,通常利用 $z_j^* = \min\{f_j(x) | x \in X\}$ ($j=1, 2, \dots, m$) 作为近似最优参考点.

为了避免因 MaOP 问题各目标函数取值范围的差别过大,以及不同量纲等带来误差而影响算法的性能,需要对目标函数进行规范化,将其函数值映射到[0,1]区间内.例如, $f_j(j=1, 2, \dots, m)$ 可以用下式进行替换.

$$\bar{f}_j = \frac{f_j - z_j^*}{z_j^{\max} - z_j^*} \quad (5)$$

这里的 z^* 同公式(4)中的规定, $z^{\max} = (z_1^{\max}, z_2^{\max}, \dots, z_m^{\max})$, 而 $z_j^{\max} = \max\{f_j(x) | x \in X\}$.进一步地,为减少计算 z^* 和 z^{\max} 的开销,用 z_j 来近似 z_j^* , 用 \bar{z}_j^{\max} 来近似 z_j^{\max} , 且 $z_j = \min\{f_j(x) | x \in pop\}$, 而 $\bar{z}_j^{\max} = \max\{f_j(x) | x \in pop\}$.这里的 pop 表示进化算法当前世代种群.由此,公式(4)可进一步表示成下面的形式.

$$g^{ie}(x|\lambda^i, z^*) = \min_{1 \leq j \leq m} \left\{ \lambda_j^i \left| \frac{f_j - z_j}{\bar{z}_j^{\max} - z_j} \right| \right\} \quad (6)$$

2.3 邻域分配

设子问题 $g^i(x)$ 对应一个权重向量 λ^i , 则 $g^i(x)$ 的邻域可通过在权重向量集合 Λ 中选择与 λ^i 的欧式距离最近的 T 个权重向量构成.即子问题 $g^i(x)$ 的邻域 $B(i) = \{i_1, i_2, \dots, i_T\}$, 其中, $\lambda^{i_1}, \lambda^{i_2}, \dots, \lambda^{i_T}$ 是与 λ^i 最近的 T 个权重向量.

图 1 以邻域规模为 5 的 2 目标优化问题为例示意邻域的概念.

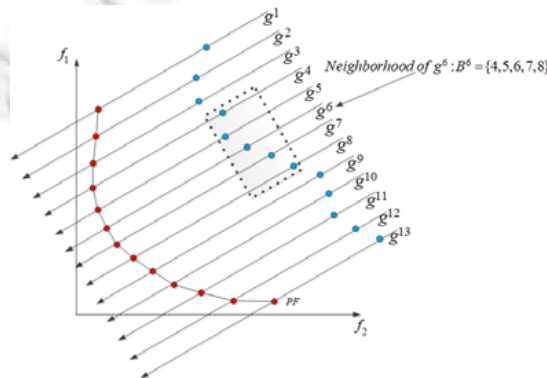


Fig.1 An illustration of neighborhood in MOEA/D

图 1 MOEA/D 中邻域示意图

每一代种群由每个子问题的当前最优解构成,对每个子问题的进化操作限制在其邻域内进行.对产生的新个体不仅更新其父代个体,而且会更新其邻域个体,使得更优的新个体尽可能地保留到下一代.正因为子问题在重组和更新操作中的相互合作,使得 MOEA/D 算法能够同时优化所有子问题.

2.4 子代生成的协同策略

子代的生成策略对进化算法的搜索性能具有重要影响,文献[33]将仿二进制交叉算子(SBX)用于 MOEA/D 算法生成子代,虽然 MOEA/D-SBX 算法在 ZDT 和 DTLZ 系列测试题上表现良好,但该算法存在探索能力不足、种群丧失多样性以及 SBX 算子产生劣解等缺点.针对这些不足, Li 等人将 DE/rand/1 差分进化和多项式变异算子用于 MOEA/D 算法,以生成质量较好的后代.实验结果表明, MOEA/D-DE 算法在处理连续的、具有复杂 PS 的 MOP 问题上具有显著的优势.必须指出,差分进化算子结合多项式变异的子代生成策略仍有较大的改善空

间,其原因在于:(1) DE算子中的差分信息类似于梯度,当随机选择的差分个体之间的差异较大时,差分信息并不能作为一种梯度信息来有效地指导算法搜索;(2) MOEA/D 算法的种群在不同的进化阶段具有不同的分布状态,有必要根据种群的分布状况自适应地选择不同的变异方式,以更加高效地搜索 Pareto 前沿。

基于此,本文利用 *DE/best/2* 差分进化算子和自适应的 SBX 算子(SA-SBX)^[34]共同形成高维多目标进化算法的子代生成策略.由于 *DE/best/2* 算子在子问题邻域内选择了更多个父代个体进行重组,丰富了子代的基因库,有利于生成多样性的个体;其次,SA-SBX 算子根据种群的分布情况自适应地选择父代-中心亦或均值-中心的方式进行变异,可促进算法较快收敛.*DE/best/2*算子和 SA-SBX 算子相互合作,协同进化,有利于生成高质量的子代个体。

设 x_{best}^i 是子问题 $g^i(x)$ 当前邻域 $B(i)$ 内最好的个体, $x_{j,c}^i, x_{k,c}^i, x_{g,c}^i, x_{h,c}^i$ 为在邻域 $B(i)$ 内随机选取的不同于 x_{best}^i 的个体, x_{target}^i 为邻域内目标个体, $j, k, g, h \in [1, 2, \dots, N]$ 为个体对应的序号,则 *DE/best/2* 差分进化算子生成的新个体 $\bar{y}^i = (\bar{y}_1^i, \bar{y}_2^i, \dots, \bar{y}_n^i)$ 的各分量 $\bar{y}_c^i (c \in [1:n])$ 可表示如下:

$$DE/best/2: \bar{y}_c^i = \begin{cases} x_{best,c}^i + F_1(x_{j,c}^i - x_{k,c}^i) + F_2(x_{g,c}^i - x_{h,c}^i), & \text{with probability } CR \\ x_{target,c}^i, & \text{with probability } 1 - CR \end{cases} \quad (7)$$

这里, CR, F_1 和 F_2 为差分进化算子的控制参数。

与 *DE/rand/1* 算子产生新个体的方式不同,*DE/best/2* 算子结合了邻域内最优个体的信息,有利于指导后代个体更有效地搜索.在沿用公式(7)若干记号的基础上,公式(8)给出了 *DE/rand/1* 差分进化方程。

$$DE/rand/1: \bar{y}_c^i = \begin{cases} x_{j,c}^i + F(x_{k,c}^i - x_{g,c}^i), & \text{with probability } CR \\ x_{target,c}^i, & \text{with probability } 1 - CR \end{cases} \quad (8)$$

图 2 和图 3 分别在 2-维决策变量空间示意了 *DE/rand/1* 和 *DE/best/2* 两种算子的搜索模式,两幅图中最内层的深色圆形区域表示最优解区域,利用 *DE/rand/1* 算子产生的后代位于最优区域的边界(如图 2 所示),而采用 *DE/best/2* 算子生成的后代则位于最优区域的中心位置,表明 *DE/best/2* 算子生成的个体质量更高。

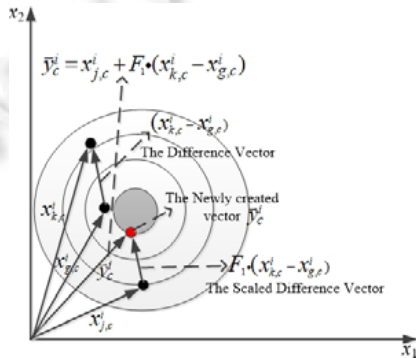


Fig.2 Search mode in *DE/rand/1*
图 2 *DE/rand/1* 搜索模式

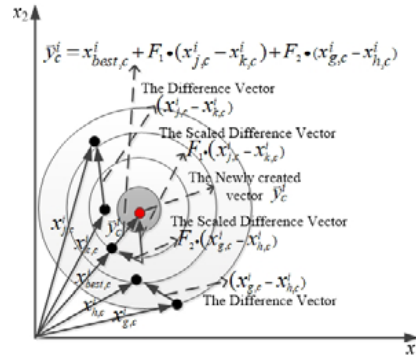


Fig.3 Search mode in *DE/best/2*
图 3 *DE/best/2* 搜索模式

实参进化算法一般有两种产生后代的方式。

- 一种是产生的后代分布于父本的附近,即父代-中心方式.其动机在于:父代个体是经历了优胜劣汰的选择过程存活下来的,因而父本有较大可能携带了与待解问题相关的信息.若生成的子代个体分布在父本周围,则有更大的可能搜索到最优解。
- 另一种则是产生的后代分布于父本的均值的附近,即均值-中心方式.其动机在于:若种群分布于最优值附近,则以父本的均值为中心搜索,会有更大的可能发现最优解。

然而在进化算法运行的整个过程中,何时采用何种后代生成策略产生高质量的子代个体,则是改善算法性能的重要问题。

设 p_1 和 p_2 是执行 SBX 算子的两个父本, d_{best} 和 d_{avg} 分别表示决策空间中种群最好的个体离种群中心点的距离以及种群各个体离种群中心点的距离之平均值, SA-SBX 算子首先根据每一代种群的 d_{best} 值和 d_{avg} 值确定出一参数 θ (如公式(9)所示), 以表征群体的分布状况; 然后定义两个虚拟的个体 v_1 和 v_2 (如公式(10)和公式(11)所示); 最后, 在此基础上计算子代个体 c_1 和 c_2 (如公式(12)、公式(13)所示). 而公式(12)和公式(13)中的参数 $\beta(u)$ 则依据公式(14)产生, 其中, u 为区间 $[0,1]$ 内均匀分布的随机数.

$$\theta = \frac{d_{best}}{d_{avg}} \quad (9)$$

$$v_1 = \frac{p_1 + p_2}{2} - \theta \frac{p_2 - p_1}{2} \quad (10)$$

$$v_2 = \frac{p_1 + p_2}{2} + \theta \frac{p_2 - p_1}{2} \quad (11)$$

$$c_1 = 0.5(1 - \beta(u))v_1 + 0.5(1 + \beta(u))v_2 \quad (12)$$

$$c_2 = 0.5(1 + \beta(u))v_1 + 0.5(1 - \beta(u))v_2 \quad (13)$$

$$\beta(u) = \begin{cases} (2u)^{\frac{1}{\eta+1}}, & \text{if } u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}}, & \text{otherwise} \end{cases} \quad (14)$$

公式(9)的 θ 值可以度量种群逼近或簇拥最优值的程度, 而且 θ 值可自动计算, 无需引入额外参数. 在公式(10)和公式(11)中, 若 $\theta=0$, 则 $v_1=v_2$ 且二者处于两个原始父本(p_1 和 p_2)的中间位置, 这样就实现了均值-中心方式的变化, 使得子代个体多样性减少, 算子的开采能力得以增强. 若 $\theta=1$, 则 $v_1=p_1, v_2=p_2$, 此时, SA-SBX 即等同于 SBX 算子. 由于 SBX 算子是一种父代-中心的变化, 因此, SA-SBX 算子就实现了以父代为中心生成后代的策略. 若 $\theta>1$, 则意味种群最好解位于群体的边界位置, v_1 或 v_2 就可以越过 p_1 或 p_2 所在位置, 使子代个体的多样性以及算法的勘探能力都能得到改善.

2.5 外部档案更新

MaOEA/DCE 算法在种群外部设置一个档案集, 用于保留算法在搜索过程中获得的非支配解. 这里采用渐进方法对档案进行更新. 算法 3 给出了外部档案的更新策略.

算法 3. Updation_of_external_archive.

Input: 产生的非劣解 x 、外部档案最大容量 N' .

Output: 外部档案中全部非劣解.

1. **if** 档案规模未达到最大容量 N'
2. 将个体 x 加入档案中.
3. 利用 Pareto 支配关系更新档案, 剔除档案中被支配的个体.
4. **else**
5. **if** 个体 x 支配了档案中部分个体
6. 剔除档案中被 x 支配的所有个体, 将新解 x 加入档案.
7. **else if** 个体 x 与档案中所有成员是非支配关系
8. 将个体 x 与档案成员一起执行个体密度估计, 并剔除多样性最差的个体.
9. **else**
10. 个体 x 被档案成员支配, 舍弃个体 x .
11. **end if**
12. **end if**
13. **end if**

14. 输出更新后的档案成员.

算法 3 的第 8 步执行个体密度估计,采用我们此前提出的 3 点最短路径策略^[35].

2.6 MaOEA/DCE 算法流程

以下为算法 4 给出的 MaOEA/DCE 算法流程.

算法 4. Flow_of_MaOEA/DCE.

Input: MaOP 问题(1);最大迭代次数 C_{max} ;分解子问题数目 N ;外部档案规模 N' ;权重向量 $\{\lambda^1, \lambda^2, \dots, \lambda^N\}$;邻域大小 T ;从邻域内选择父本的概率 δ ;邻域内父代被子代替换的最大数目 n_r ;分布指数 η .

Output: 近似 Pareto 解集 $PS: \{x^1, x^2, \dots, x^N\}$; 近似 Pareto 前沿 $PF: \{F(x^1), F(x^2), \dots, F(x^N)\}$.

1. 初始化:

1.1. 利用权重向量之间的欧氏距离确定每个权重向量的邻域.不妨设每个权重向量的邻域规模为 T ,且设 $B(i) = \{i_1, i_2, \dots, i_T\}, i = 1, 2, \dots, N$, 其中, $\lambda^{i_1}, \lambda^{i_2}, \dots, \lambda^{i_T}$ 为距离 λ^i 最近的 T 个权重向量;

1.2. 采用随机化方法生成初始种群 $\{x^1, x^2, \dots, x^N\}$, 设 $FV^i = F(x^i), i = 1, 2, \dots, N$;

1.3. 初始化 $z = (z_1, z_2, \dots, z_m)$, 其中, $z_j = \min \{f_j(x) | x \in pop\}, j = 1, 2, \dots, m$;

1.4. 初始化迭代计数器 $t = 0$;

2. 更新:

for $i = 1, \dots, N$ **do**

2.1. 在区间 $[0, 1]$ 内均匀地产生随机数 $rand$, 并置 $P = \begin{cases} B(i), & \text{if } rand < \delta \\ \{1, \dots, N\}, & \text{otherwise} \end{cases}$;

2.2. 繁殖: 从 P 内选出合适的个体执行 $DE/rand/2$ 算子, 产生中间个体 \bar{y} , 并对 \bar{y} 执行 SA-SBX 变异算子, 以生成子代新个体 y ;

2.3. 修复: 利用问题相关的启发式知识修复 y , 得到 y' ;

2.4. 更新 z : 若 $z_j > f_j(y')$, 则 $z_j = f_j(y'), j = 1, 2, \dots, m$;

2.5. 更新邻域解: 设 $k = 0$, 并执行以下操作:

1) 如果 $k = n_r$ 或者 P 为空, 则执行步骤 3; 否则, 从 P 中随机选一个下标 j ;

2) 若 $g^{te}(y' | \lambda^j, z) \leq g^{te}(x^j | \lambda^j, z)$, 则 $x^j = y', FV^j = F(y')$, 且 $k = k + 1$;

3) 从 P 中删除下标为 j 的个体并转步骤 1);

2.6. 更新外部档案: 利用算法 3 更新外部档案, 并置迭代计数器 $t = t + 1$;

3. 停止判断: 若满足停止条件, 则算法停止, 输出 $\{x^1, x^2, \dots, x^N\}$ 和 $\{F(x^1), F(x^2), \dots, F(x^N)\}$.

算法 4 的第 2.3 步修复 y 的方法如下: 如果 y 的第 j 维的值不在 y_j 的范围之内, 则对 y_j 的值按公式(15)进行修复.

$$y_j = \begin{cases} l_j + r(l_j - y_j), & y_j < l_j \\ u_j - r(u_j - y_j), & y_j > u_j \end{cases} \quad (15)$$

其中, u_j, l_j 分别为 y_j 的上、下界, $r \in (0, 1)$ 区间内均匀分布的随机数.

3 实验结果与分析

3.1 实验设置

1) 对等比较算法

为了检验 MaOEA/DCE 算法的性能, 利用 3 种 MOEA/D 算法的变种 MOEA/D-SBX^[33], MOEA/D-DE^[24] 和 MOEA/D-ACD^[21] 以及 2 个未使用分解策略的典型算法: 改进型非劣分类遗传算法 NSGA-II^[2] 和基于档案的混合分散搜索算法 AbYSS^[9] 共计 5 种经典的 MOEA 算法与本文算法进行性能对比, 以评估 MaOEA/DCE 算法的性能.

2) 高维多目标测试函数集

本文实验将 MaOEA/DCE 算法与 MOEA/D-SBX、MOEA/D-DE、MOEA/D-ACD、NSGA-II 和 AbYSS 算法一同在 4 目标、10 目标和 30 目标的 DTLZ{1,2,4,5} 基准多目标测试问题^[36]上进行实验.需要说明的是,本文的所有实验均在 ThinkPad E565 电脑上运行,电脑的配置如下:1) CPU: AMD A10-8700P; 2) 1.8GHz 主频;3) 8.0G 内存;4) Windows 7 64 位操作系统.

实验中算法和测试函数的参数设置参考文献[37]中的取值,亦即,(1) 当目标数 $m=4$,决策向量的长度 $n=10$,算法的迭代的最大次数为 2 000 代;(2) 当目标数 $m=10$ 时,决策向量的长度 $n=20$,算法的迭代的最大次数为 5 000 代;(3) 当目标数 $m=30$ 时,决策向量的长度 $n=50$,算法的迭代的最大次数为 10 000 代.

3) 性能指标

在高维多目标优化的研究中,超体积(hypervolume,简称 HV)经常用来评估算法的性能,其原因在于 HV 指标是已知的与 Pareto 关系兼容的一元评估指标,但其不足之处在于,随着优化目标数目的增大(例如目标个数超过 10)以及非支配解数目较大(例如解个体数目超过 1 000)时,HV 值的计算将耗费大量的计算资源.反转世代距离(inverted generational distance,简称 IGD)通过在真实 Pareto 前沿上均匀抽样一定数目的解点,然后计算这些解点与近似 Pareto 最优解之间的欧氏距离来评估算法的收敛性和多样性.与 HV 指标相比,IGD 指标的计算效率更高,这是由于 IGD 指标是通过计算两个点之间的距离获得的,而距离的计算一般较为高效,即便在高维多目标空间中亦是如此.但必须指出,IGD 指标与 HV 指标相比,最大的缺陷在于其不具有 Pareto 兼容性.为使利用 IGD 指标计算效率高,同时克服其不兼容 Pareto 最优性的缺陷,文献[38]利用改进的 IGD 指标,即 IGD^+ 指标来评估算法的性能. IGD^+ 不仅计算两个点之间的距离,同时还考虑到它们之间的 Pareto 关系,因而 IGD^+ 指标兼具弱 Pareto 兼容性,从而提高了度量算法性能的准确性.

对于一个满足式(1)的最小化多目标优化问题,设 m 为优化目标的个数, Z 是从其已知的真实 Pareto 前沿上均匀采样获得的参考点集合, A 为算法获得的近似 Pareto 解集, $z_j=(z_1, z_2, \dots, z_m) \in Z, j=1, 2, \dots, |Z|, a_i=(a_1, a_2, \dots, a_m) \in A, i=1, 2, \dots, |A|$, 则 IGD^+ 指标的计算方式如下:

$$IGD^+(A) = \frac{1}{|Z|} \sum_{j=1}^{|Z|} \min_{a_i \in A} d_{IGD^+}(a_i, z_j) \tag{16}$$

其中, $d_{IGD^+}(a_i, z_j) = \sqrt{\sum_{k=1}^m \left(\max \left\{ \frac{a_k - z_k}{f_k^{\max} - f_k^{\min}}, 0 \right\} \right)^2}$ 表示归一化的欧氏距离, f_m^{\max} 和 f_m^{\min} 分别表示集合 Z 在第 m 个目标函数上的最大值和最小值.从公式(16)不难得知,如果 IGD^+ 指标值越小,则近似解集的质量(收敛性和多样性)就越好.为了计算 IGD^+ 的值,这里的实验在各个测试函数的真实 Pareto 前沿上都均匀地采样 10 000 个解点.

4) 实验参数

本文中参与对等比较的算法的主要参数见表 2.表 2 中各参数的含义可通过查阅相应的参考文献而获得.

Table 2 Parameter settings of six peer algorithms

表 2 6 种对比算法的参数设置

Algorithm	Parameter settings
NSGA-II	$N=100, p_c=0.9, p_m=1/n, \eta_c=20, \eta_m=20$
AbYSS	$N=100, N_{RefSet1}=10, N_{RefSet2}=10, p_c=0.9, p_m=1/n, \eta_c=20, \eta_m=20$
MOEA/D-SBX	$N=100, p_m=1/n, \eta_m=20, T=20, \delta=0.9, n_r=2$
MOEA/D-DE	$N=100, CR=1.0, F=0.5, p_m=1/n, \eta_m=20, T=20, \delta=0.9, n_r=2$
MOEA/D-ACD	$N=100, CR=1.0, F=0.5, p_m=1/n, \eta_m=20, T=20, \delta=0.9, n_r=2$
MaOEA/DCE	$N=100, CR=1.0, F=0.5, p_m=1/n, \eta_m=20, T=20, \delta=0.9, n_r=2$

为使实验结果客观、合理和可信,本文的每一种算法在每一个测试实例上均独立执行 30 次,以此统计实验的结果,而且利用 Wilcoxon's rank sum test 来统计和比较算法之间水平为 $\alpha=0.5$ 的显著性差异.

3.2 实验与分析

表3~表6给出了6种算法在DTLZ₁{1,2,4,5}测试函数上分别具有4目标、10目标和30目标时得到的IGD⁺指标的平均值(mean)和标准方差(std).各表中不同的算法在相同的测试问题上所取得的最佳的IGD⁺的均值用粗体凸显.各表中的“+”、“-”和“≈”符号分别用于标记对比算法获得的结果要显著地优于、劣于和无差别于MaOEA/DCE算法所获得的结果,这里,用采样水平 $\alpha=0.05$ 的Wilcoxon秩和检验来统计获得的数据.

Table 3 Comparison of IGD⁺ on DTLZ1 instances for six algorithms表3 6种算法在DTLZ1函数上获得IGD⁺值的比较

目标个数/算法		NSGA-II	AbYSS	MOEA/D-SBX	MOEA/D-DE	MOEA/D-ACD	MaOEA/DCE
4目标	mean	7.7911e-02	6.9832e-02	2.4356e-02	1.8187e-02	2.0701e-02	6.1942e-03
	std	5.6606e-04	5.4157e-04	8.1132e-04	2.6219e-03	1.0252e-03	6.6735e-04
	rank	6-	5-	4-	2-	3-	1
10目标	mean	2.2968e-01	3.3845e-01	7.2919e-02	3.4618e-02	6.4427e-02	1.8430e-02
	std	6.5522e-04	4.1108e-04	9.2253e-04	2.1961e-03	2.0277e-03	5.9821e-04
	rank	5-	6-	4-	2-	3-	1
30目标	mean	6.7734e-01	9.4144e-01	3.0130e-01	2.9392e-01	2.6554e-01	5.1883e-02
	std	9.4339e-04	5.6814e-04	1.3106e-03	4.2951e-03	3.1288e-03	4.8649e-04
	rank	5-	6-	4-	3-	2-	1
rank suam		16	17	12	7	8	3
final rank		5	6	4	2	3	1
better/worst/similar		0/3/0	0/3/0	0/3/0	0/3/0	0/3/0	-

Table 4 Comparison of IGD⁺ on DTLZ2 instances for six algorithms表4 6种算法在DTLZ2函数上获得IGD⁺值的比较

目标个数/算法		NSGA-II	AbYSS	MOEA/D-SBX	MOEA/D-DE	MOEA/D-ACD	MaOEA/DCE
4目标	mean	6.3109e-03	2.5341e-02	3.8412e-02	1.9241e-02	4.7625e-02	8.5415e-03
	std	5.1819e-04	7.1915e-04	6.8459e-04	4.1851e-04	1.74861e-04	4.874e-04
	rank	1+	4-	5-	3-	6-	2
10目标	mean	5.2851e-02	6.6171e-02	2.0151e-01	8.5241e-02	1.0741e-01	4.8169e-02
	std	5.6740e-04	6.9151e-04	5.9871e-04	5.1762e-04	3.5061e-04	5.5651e-04
	rank	2≈	3-	6-	4-	5-	1
30目标	mean	3.5548e-01	6.8851e-01	4.8158e-01	2.7941e-01	5.9967e-01	1.9181e-01
	std	6.5141e-03	5.9210e-04	4.2225e-04	6.2131e-03	5.1101e-04	3.6518e-03
	rank	3-	6-	4-	2-	5-	1
rank suam		6	13	15	9	16	4
final rank		2	5	6	3	4	1
better/worst/similar		1/1/1	0/3/0	0/3/0	0/3/0	0/3/0	-

Table 5 Comparison of IGD⁺ on DTLZ4 instances for six algorithms表5 6种算法在DTLZ4函数上获得IGD⁺值的比较

目标个数/算法		NSGA-II	AbYSS	MOEA/D-SBX	MOEA/D-DE	MOEA/D-ACD	MaOEA/DCE
4目标	mean	6.3138e-02	7.6824e-02	6.9656e-03	9.5511e-03	1.7779e-02	5.4201e-03
	std	2.16e-04	4.1382e-04	5.8299e-04	6.2751e-03	2.21e-03	7.0124e-04
	rank	5-	6-	2≈	3-	4-	1
10目标	mean	2.1433e-01	1.9838e-01	2.1656e-03	3.0675e-03	5.0197e-02	2.5610e-03
	std	5.6443e-04	6.9283e-04	5.9915e-04	5.1867e-04	3.5119e-04	5.5721e-04
	rank	6-	5-	1+	3≈	4-	2
30目标	mean	8.5513e-01	6.8923e-01	1.2295e-02	2.4986e-02	3.9911e-01	2.8872e-02
	std	6.1546e-04	5.8228e-04	4.2329e-04	9.2289e-03	8.2117e-03	3.7587e-04
	rank	6-	5-	1+	2≈	4-	3
rank suam		17	16	4	8	12	6
final rank		6	5	1	3	4	2
better/worst/similar		0/3/0	0/3/0	2/0/1	0/1/2	0/3/0	-

Table 6 Comparison of IGD⁺ on DTLZ5 instances for six algorithms
表 6 6 种算法在 DTLZ5 函数上获得 IGD⁺值的比较

目标个数/算法		NSGA-II	AbYSS	MOEA/D-SBX	MOEA/D-DE	MOEA/D-ACD	MaOEA/DCE
4 目标	mean	1.5116e-03	8.1541e-03	4.6114e-03	6.3841e-03	2.9141e-03	7.1981e-04
	std	6.8917e-04	4.1123e-04	5.6101e-04	3.2987e-04	4.5005e-04	3.3176e-04
	rank	2-	6-	4-	5-	3-	1
10 目标	mean	6.6810e-03	3.5141e-02	1.9184e-02	4.8584e-02	7.9515e-03	3.4214e-03
	std	7.0151e-04	4.4414e-04	6.6516e-04	5.8771e-04	4.6001e-04	3.6011e-04
	rank	2-	5-	4-	6-	3-	1
30 目标	mean	3.5945e-02	9.2494e-02	5.5441e-02	2.5541e-01	3.0541e-02	2.0713e-02
	std	4.6054e-03	8.5104e-04	2.9414e-03	8.8151e-03	5.1112e-03	5.6807e-03
	rank	3-	5-	4-	6-	2≈	1
rank suam		7	16	12	17	8	3
final rank		2	5	4	6	3	1
better/worst/similar		0/3/0	0/3/0	0/3/0	0/3/0	0/2/1	-

从表 3 可以看出, MaOEA/DCE 算法分别在 DTLZ1(4,10,30)测试实例上获得了最佳的 IGD⁺指标的平均值,而其他对比算法 NSGA-II、AbYSS、MOEA/D-SBX、MOEA/D-ACD 无一能够获得最佳的 IGD⁺均值;其次,通过统计各算法在 DTLZ1(4,10,30)这 3 个测试例上获得的 IGD⁺值的排名,表现最好的是本文的算法,其次是 MOEA/D-DE,随后依次是 MOEA/D-ACD、MOEA/D-SBX、NSGA-II 和 AbYSS.从表 3 的“better/worst/similar”结果来看,5 种参与对比的 MOEA 算法获得的结果都为“0/3/0”,由此表明本文算法 MaOEA/DCE 在这 3 个高维目标的 DTLZ1 测试函数上具有比其他对比算法显著更优的 IGD⁺性能,亦即 MaOEA/DCE 算法与其他对比算法相比,具有更好的收敛性与多样性的综合性能.由于 DTLZ1 测试函数的 Pareto 前沿是线性、多模态的,这也表明 MaOEA/DCE 在求解具有线性、多模态的高维多目标优化问题时具有较好的性能.

从表 4 可以看出, MaOEA/DCE 算法在 DTLZ2(10,30)两个测试实例上获得了最好的 IGD⁺平均值, NSGA-II 算法在 DTLZ2(4)测试问题上得到了最佳的 IGD⁺平均值,而 AbYSS、MOEA/D-SBX、MOEA/D-DE 和 MOEA/ACD 等则未能获得最好的 IGD⁺均值.在 4 目标的 DTLZ2 测试问题上,尽管 NSGA-II 算法获得了最好的 IGD⁺均值,但本文算法在该问题上获得的 IGD⁺均值仅略逊于 NSGA-II 算法(二者处于相同的数量级 10⁻²),而优于 MOEA/D-SBX、MOEA/D-DE、MOEA/D-ACD 和 AbYSS 算法.此外,通过统计各算法在 DTLZ2(4,10,30)这 3 个测试例上获得 IGD⁺值的排名可以发现,排名第一的为 MOEA/D-SBX 算法,后面依次为 MOEA/D-DE、MOEA/D-ACD、AbYSS 和 MOEA/D-SBX 算法.从表 4 的“better/worst/similar”结果来看,各对比算法在 DTLZ2(4,10,30)这 3 个测试函数上只有 NSGA-II 算法的统计结果为“1/1/1”,其他 4 种对比算法所获得的统计结果都是“0/3/0”,这表明 MaOEA/DCE 算法在 DTLZ2(4,10,30)这 3 个测试实例上与其他对比算法相比,具有统计学意义上显著较优的 IGD⁺性能,即 MaOEA/DCE 算法在这 3 个测试实例上获得了相对最优的收敛性与多样性.考虑到 DTLZ2 的 Pareto 前沿为凹的,这也表明 MaOEA/DCE 算法能够较好地求解 Pareto 前沿为凹的一类 MaOP 问题.

从表 5 可以看出, MaOEA/DCE 算法在 4 目标的 DTLZ4 问题上获得了最好的 IGD⁺均值, MOEA/D-SBX 算法在 DTLZ4(10,30)两个测试实例上获得了最好的 IGD⁺的平均值,而其他几种对等比较算法则未能获得最优的 IGD⁺平均值.虽然 MaOEA/DCE 算法未能在 DTLZ(10,30)两个测试问题上获得最好的 IGD⁺平均值,但 MaOEA/DCE 算法获得的 IGD⁺均值仅稍逊于 NSGA-II 算法获得的最优值.它们具有相同的数量级,即在 10 目标的 DTLZ4 问题上,二者的数量级为 10⁻³;在 30 目标的 DTLZ4 问题上,二者的数量级为 10⁻².通过统计各算法在 DTLZ4(4,10,30)这 3 个测试问题上获得的 IGD⁺值的排名, MOEA/D-SBX 算法排名第 1,其次是 MaOEA/DCE 算法,随后是 MOEA/D-DE、MOEA/D-ACD、AbYSS 和 NSGA-II.从表 5 的“better/worst/similar”结果来看,只有 MOEA/D-SBX 的结果是“2/0/1”以及 MOEA/D-DE 的结果是“0/1/2”,而其他对比算法的结果均是“0/3/0”.这表明本文算法在 DTLZ4(4,10,30)这 3 个测试问题上仅稍逊于 MOEA/D-SBX 算法,而优于其他几种对比算法. DTLZ4 问题的 Pareto 前沿为凹形、有偏之特点,说明本文算法在求解这种问题特征的 MaOP 问题上具有一定的优势.

从表 6 可以看出,本文的 MaOEA/DCE 在 DTLZ5(4,10,30)这 3 个测试问题上均获得了最优的 IGD⁺均值.通

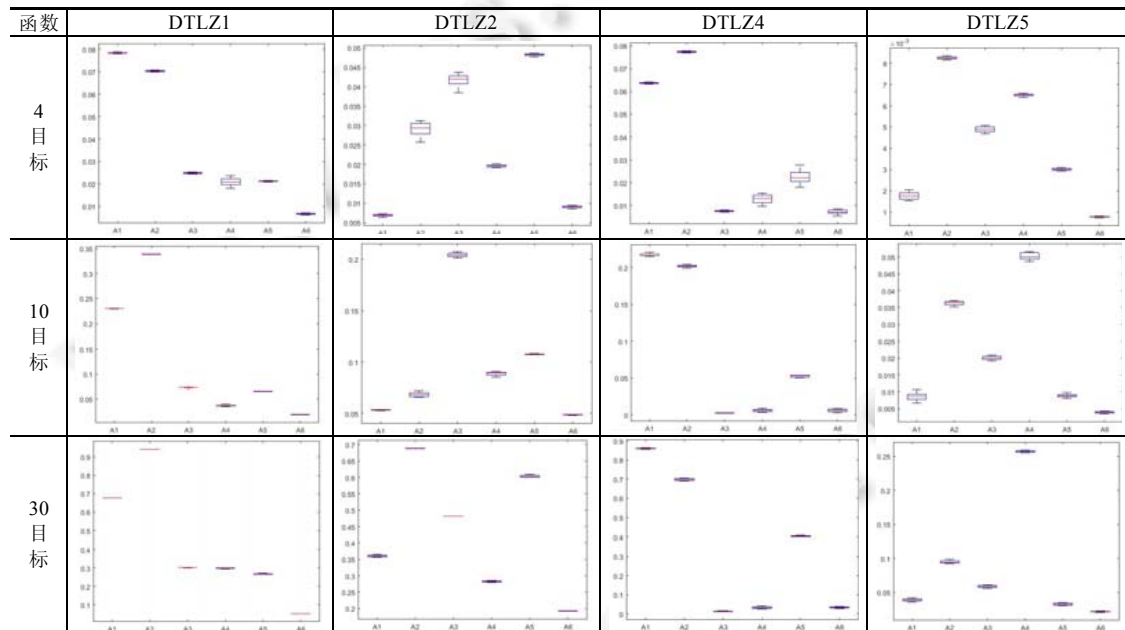
过统计各算法在 DTLZ5(4,10,30)这 3 个测试例上获得的 IGD^+ 值的最终排序,表现最好的是 MaOEA/DCE 算法,其次是 NSGA-II,随后依次是 MOEA/D-ACD、MOEA/D-SBX、AbYSS 和 MOEA/D-DE.从表 6 的“better/worst/similar”结果来看,各算法在 DTLZ5(4,10,30)这 3 个测试函数上只有 MOEA/D-ACD 的结果是“0/2/1”,而其他 4 种对比算法的结果均是“0/3/0”,这表明, MaOEA/DCE 算法在这 3 个高维多目标 DTLZ5 问题上具有显著的 IGD^+ 性能优势. DTLZ5 问题具有凹形且退化的 Pareto 前沿,而本文算法能够在这种类型的 MaOP 问题上获得较好的结果.

综合表 3~表 6 的实验结果, MaOEA/DCE 算法相对于其他 5 种代表性的 MOEA 算法,总体上具有显著的收敛性和多样性的优势.究其原因,本文算法利用混合水平正交实验设计方法产生适当规模且均匀分布的权重向量,改善了算法的分布性;其次, MaOEA/DCE 算法结合差分进化算子和自适应的 SBX 算子产生后代,提高了算法的收敛性能.这些策略相互协同,显著地改善了算法求解 MaOP 问题的性能.

为了进一步考察各算法解题的稳定性,表 7 用盒图(box plot)的形式给出了各算法在 DTLZ{1,2,4,5}系列函数上分别具有 4,10 和 30 目标问题上独立运行 30 次获得的 IGD^+ 值的统计结果.为了便于图示,这里分别将 NSGA-II、AbYSS、MOEA/D-SBX、MOEA/D-DE、MOEA/D-ACD 和 MaOEA/DCE 简记为 A1、A2、A3、A4、A5 和 A6.

Table 7 Box plots of IGD^+ derived from 30 independent runs for six algorithms

表 7 6 种算法独立运行 30 次获得的 IGD^+ 结果的盒图



从表 7 可以看出,对于 4 目标的 DTLZ{1,2,4,5}问题而言, MaOEA/DCE 算法在 DTLZ1、DTLZ4 和 DTLZ5 测试例上获得的 IGD^+ 的值具有最佳的稳定性, NSGA-II 算法在 DTLZ2 问题上获得了最好的稳定性,而本文的算法在 DTLZ2 问题上的稳定性仅次于 NSGA-II 算法,而优于其他几种对比算法.对于 10 目标的 DTLZ{1,2,4,5}问题,本文算法在 DTLZ1、DTLZ2、DTLZ5 这 3 个测试问题上的 IGD^+ 值的稳定性最好,而在 DTLZ4 问题上, MaOEA/DCE 算法的 IGD^+ 值的稳定性略逊于 MOEA/D-SBX 算法,而与 MOEA/D-DE 算法相当.对于 30 目标的 DTLZ{1,2,4,5}问题而言, MaOEA/DCE 算法在 DTLZ1、DTLZ2 和 DTLZ5 这 3 个函数上获得的 IGD^+ 值具有最好的稳定性, MOEA/D-SBX 算法在 DTLZ4 上的 IGD^+ 值具有最好的稳定性,而本文算法在 DTLZ4 问题上获得 IGD^+ 值的稳定性稍逊于 MOEA/D-SBX,而与 MOEA/D-DE 算法相当.从表 7 不难看出, MaOEA/DCE 算法在绝大多数的 MaOP 问题上获得了最好的 IGD^+ 值的稳定性.究其原因, MaOEA/DCE 算法将差分进化算法和自适应的

SBX 算法进行协同产生后代,从而使得子代个体的质量得以改善,因而算法解题的鲁棒性也得到提高。

3.3 进一步讨论

通过上面的实验,验证了 MaOEA/DCE 算法总体上具有较显著的收敛性、多样性和鲁棒性优势,表明了利用混合水平正交实验设计方法产生适当规模且均匀分布的权重向量的策略,以及将差分进化算子和自适应 SBX 算子进行协同以产生高质量的后代个体的方法相结合,有效地提高了算法的整体性能。为了进一步验证差分进化算子和自适应 SBX 算子进行协同进化的有效性,设计如下实验方案:首先,基于 MOEA/D 算法框架构造 3 种算法 B1、B2 和 B3,并将这 3 种算法分别在 4 目标、10 目标和 30 目标的 $DTLZ\{1,2,4,5\}$ 测试函数集上进行 IGD^+ 性能指标的比较。需要说明的是,这里的 B1 算法指采用差分进化作为变异算子算法,B2 算法指使用自适应 SBX 算子作为变异策略的算法,B3 算法为运用本文提出的差分进化和自适应 SBX 算子相互协同的算法。实验中,3 种算法除了所使用的变异策略不同以外,这些算法的其他要素是相同的。表 8~表 11 分别给出了 B1、B2 和 B3 这 3 种算法分别在 $DTLZ1$ 、 $DTLZ2$ 、 $DTLZ4$ 和 $DTLZ5$ 系列函数上获得的 IGD^+ 指标值。为了减少随机误差对统计结果的影响,这里所有的实验各执行 30 次。

从表 8 可以看出,B3 算法在 4 目标、10 目标和 30 目标的 $DTLZ1$ 函数上均获得了最佳的 IGD^+ 值,而且从 IGD^+ 值的排名来看,B3 算法排名第一。此外,从表 8 的“better/worst/similar”结果来看,B1 和 B2 在 $DTLZ1(4,10,30)$ 这 3 个测试题上的结果均是“0/3/0”,表明 B3 算法对比其他两种算法具有显著的 IGD^+ 性能优势。从表 9 可以看出,B3 算法分别在 $DTLZ2(4,10,30)$ 这 3 个测试函数上获得了最好的 IGD^+ 值,而且从 IGD^+ 值的排名来看,B3 算法排名第 1。此外,从表 9 的“better/worst/similar”结果来看,B1 和 B2 算法在 $DTLZ2(4,10,30)$ 上的对比结果均为“0/3/0”,表明 B3 算法在 $DTLZ2$ 函数上具有显著的 IGD^+ 性能的优势。从表 10 来看,B3 分别在 4 目标和 10 目标的 $DTLZ4$ 测试问题上获得了最好的 IGD^+ 值,而 B1 在 30 目标的 $DTLZ4$ 问题上获得了最佳的 IGD^+ 值。从 3 种算法的 IGD^+ 值的排名来看,B3 算法排名第 1。而从表 10 的“better/worst/similar”结果来看,B1 算法是“1/1/1”,而 B2 算法是“0/3/0”,表现出 B3 算法在 $DTLZ4$ 问题上具有一定的 IGD^+ 性能优势。从表 11 来看,B3 算法分别在 4 目标和 10 目标的 $DTLZ5$ 问题上获得了最佳的 IGD^+ 值,B2 在 30 目标的 $DTLZ5$ 问题上获得了最好的 IGD^+ 值。从 3 种算法的 IGD^+ 值的排名来看,B3 算法排名第 1。另外,从表 11 的“better/worst/similar”结果来看,B1 和 B2 算法的结果分别是“0/3/0”和“1/2/0”,表明 B3 算法在 $DTLZ5$ 问题上具有一定的 IGD^+ 的性能优势。综合表 8~表 11 的实验结果不难发现,采用差分进化和自适应 SBX 算子进行协同的 B3 算法在 $DTLZ\{1,2,4,5\}$ 系列测试问题上具有较显著的 IGD^+ 性能优势。由此表明,本文提出利用差分进化算子和自适应 SBX 算子进行协同的策略是有效的。

Table 8 Comparison of IGD^+ performance on $DTLZ1$ instances for three algorithms

表 8 3 种算法在 $DTLZ1$ 函数上进行 IGD^+ 性能比较

目标个数/算法		B1	B2	B3
4 目标	mean	2.0511e-02	2.2647e-02	5.9942e-03
	std	3.8281e-03	2.5581e-03	5.6684e-04
	rank	2-	3-	1
10 目标	mean	4.6771e-02	5.8115e-02	1.8430e-02
	std	3.0114e-03	3.2147e-03	4.7798e-04
	rank	2-	3-	1
30 目标	mean	2.5411e-01	3.0414e-01	5.1883e-02
	std	5.8128e-03	2.9358e-03	5.3514e-04
	rank	2-	3-	1
rank sum		6	9	3
final rank		2	3	1
better/worst/similar		0/3/0	0/3/0	-

Table 9 Comparison of IGD⁺ performance on DTLZ2 instances for three algorithms**表 9** 3 种算法在 DTLZ2 函数上进行 IGD⁺性能比较

目标个数/算法		B1	B2	B3
4 目标	mean	1.5114e-02	4.7625e-02	7.1556e-03
	std	3.9188e-04	1.74861e-04	4.9874e-04
	rank	2-	3-	1
10 目标	mean	9.0121e-02	1.0741e-01	5.1689e-02
	std	4.9873e-04	3.5061e-04	4.9551e-04
	rank	2-	3-	1
30 目标	mean	3.0052e-01	4.8817e-01	2.0111e-01
	std	5.9327e-03	4.7980e-04	4.0848e-03
	rank	2-	3-	1
rank sum		6	9	3
final rank		2	3	1
better/worst/similar		0/3/0	0/3/0	-

Table 10 Comparison of IGD⁺ performance on DTLZ4 instances for three algorithms**表 10** 3 种算法在 DTLZ4 函数上进行 IGD⁺性能比较

目标个数/算法		B1	B2	B3
4 目标	mean	8.9602e-03	2.3658e-02	5.7813e-03
	std	5.9812e-03	2.7210e-03	6.8231e-04
	rank	2-	3-	1
10 目标	mean	2.8091e-03	4.9817e-02	2.3515e-03
	std	4.8879e-04	4.0108e-04	4.9382e-04
	rank	2≈	3-	1
30 目标	mean	3.0171e-02	4.1238e-01	3.0963e-02
	std	8.1099e-03	7.1721e-03	4.5796e-04
	rank	1+	3-	2
rank sum		5	9	4
final rank		2	3	1
better/worst/similar		1/1/1	0/3/0	-

Table 11 Comparison of IGD⁺ performance on DTLZ5 instances for three algorithms**表 11** 3 种算法在 DTLZ5 函数上进行 IGD⁺性能比较

目标个数/算法		B1	B2	B3
4 目标	mean	8.1541e-03	3.1244e-03	2.0224e-03
	std	4.1123e-04	3.8884e-04	4.7881e-04
	rank	3-	2-	1
10 目标	mean	3.5141e-02	8.0141e-03	6.6810e-03
	std	4.4414e-04	3.9777e-04	7.0333e-04
	rank	3-	2-	1
30 目标	mean	9.8440e-02	2.9941e-02	3.6885e-02
	std	7.9459e-04	4.9888e-03	5.1523e-03
	rank	3-	1+	2
rank sum		9	5	4
final rank		3	2	1
better/worst/similar		0/3/0	1/2/0	-

4 结 论

高维多目标优化问题巨大的目标空间使得一些经典的多目标进化算法面临极大的挑战,迫切需要发展新的高维多目标进化算法予以应对.本文提出一种基于分解和协同进化策略的高维多目标进化算法 MaOEA/DCE.该算法利用混合水平正交实验设计方法产生权重向量集合,并将差分进化算子和自适应 SBX 算子进行协同,以产生后代个体.MaOEA/DCE 算法在 12 个基准高维多目标优化问题上与另 5 种代表性 MOEA 算法进行 IGD⁺性能实验,结果表明,本文算法具有比较显著的收敛性、多样性和稳定性方面的优势.未来将利用一些更加复杂的 MaOP 问题测试 MaOEA/DCE 算法,以及利用 MaOEA/DCE 算法求解一些实践中的高维多目标优化问

题,以不断测试和改进算法的性能.

References:

- [1] Gong MG, Jiao LC, Yang DD, Ma WP. Research on evolutionary multi-objective optimization algorithms. *Ruan Jian Xue Bao/ Journal of Software*, 2009,20(2):271–289 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.03483]
- [2] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2002,6(2):182–197. [doi: 10.1109/4235.996017]
- [3] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. In: Giannakoglou K, Tsahalis DT, Periaux J, Papailious KD, Fogarty T, eds. *Proc. of the Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*. Berlin: Springer-Verlag, 2002. 95–100.
- [4] Coello CCA, Pulido GT, Lechuga MS. Handling multiple objectives with particles swarm optimization. *IEEE Trans. on Evolutionary Computation*, 2004,8(3):256–279. [doi: 10.1109/TEVC.2004.826067]
- [5] Nebro AJ, Durillo JJ, García-Nieto J, Coello CCA, Luna F, Alba E. SMPSO: A new PSO metaheuristic for multi-objective optimization. In: *Proc. of the IEEE Symp. on Computer Int'l Multi-criteria Decision Making*. 2009. 66–73. [doi: 10.1109/MCDM.2009.4938830]
- [6] Zhang QF, Zhou AM, Jin Y. RM-MEDA: A regularity model based multi-objective estimation of distribution algorithm. *IEEE Trans. on Evolutionary Computations*, 2007,12(1):1–23. [doi: 10.1109/TEVC.2007.894202]
- [7] Xie CW, Xu L, Zhao HR, Xia XW, Wei B. Multi-objective fireworks optimization algorithm using elite opposition-based learning. *Acta Electronica Sinica*, 2016,44(5):1180–1188 (in Chinese with English abstract).
- [8] Molina J, Laguna M, Martí R, Caballero R. SSPMO: A scatter tabu search procedure for non-linear multi-objective optimization. *Inform Journal on Computing*, 2007,19(1):91–100. [doi: 10.1287/ijoc.1050.0149]
- [9] Nebro AJ, Luna F, Alba E, Dorronsoro B, Durillo JJ, Beham A. AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Trans. on Evolutionary Computation*, 2008,12(4):439–457. [doi: 10.1109/TEVC.2007.913109]
- [10] Soliman O, Bui LT, Abbass H. A memetic coevolutionary multi-objective differential evolution algorithm. In: *Proc. of the Multi-objective Memetic Alg*. Berlin: Springer-Verlag, 2009. 369–388. [doi: 10.1007/978-3-540-88051-6_17]
- [11] Adra SF, Fleming PJ. A diversity management operator for evolutionary many-objective optimization. In: *Proc. of the Evolutionary Multi-criterion Optimization*. Springer-Verlag, 2009. 81–94.
- [12] Wagner T, Beume N, Naujoks B. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: *Proc. of the Evolutionary Multi-criterion Optimization*. Springer-Verlag, 2007. 742–756.
- [13] Li MQ, Yang SX, Liu XH. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2014,18(3):348–365.
- [14] Mostaghim S, Schmeck H. Distance based ranking in many-objective particle swarm optimization. In: *Proc. of the Parallel Problem Solving from Nature*. Springer-Verlag, 2008. 753–762.
- [15] Zhang QF, Li H. MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Trans. on Evolutionary Computation*, 2007,11(6):712–731. [doi: 10.1109/TEVC.2007.892759]
- [16] Li H, Landa-Silva D. An adaptive evolutionary multi-objective approach based on simulated annealing. *Evolutionary Computation*, 2011,19(4):561–595.
- [17] Moubayed NA, Petrovski A, McCall J. A novel smart multi-objective particle swarm optimization using decomposition. In: *Proc. of the 11th PPSN*. 2010. 1–10.
- [18] Ke L, Zhang Q, Battiti R. *Multi-objective Combinatorial Optimization by Using Decomposition and Ant Colony*. Colchester: University of Essex, 2010.
- [19] Zhang Q, Li H, Maringer D, *et al.* MOEA/D with NBI-style Tchebycheff approach for portfolio management. In: *Proc. of the Evolutionary Computation*. IEEE, 2010. 1–8.
- [20] Ishibuchi H, Sakane Y, Tsukamoto N, *et al.* Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multi-objective evolutionary algorithm. In: *Proc. of the Int'l Conf. on Evolutionary Multi-criterion Optimization*. Heidelberg: Springer-Verlag, 2009. 438–452.

- [21] Wang LP, Zhang QF, Zhou AM, Gong M, Jiao L. Constrained subproblems in a decomposition-based multi-objective evolutionary algorithm. *IEEE Trans. on Evolutionary Computation*, 2016,20(3):475–480. [doi: 10.1109/TEVC.2015.2457616]
- [22] Tan Y, Jiao Y, Li H, *et al.* MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. *Computers & Operations Research*, 2013,40(6):1648–1660.
- [23] Qi Y, Ma X, Liu F, *et al.* MOEA/D with adaptive weight adjustment. *Evolutionary Computation*, 2014,22(2):231–263.
- [24] Li H, Zhang Q. Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2009,13(2):284–302.
- [25] Zhou AM, Zhang QF, Zhang GX. A multi-objective evolutionary algorithm based on decomposition and probability model. In: *Proc. of the IEEE Congress of Evolutionary Computation (CEC 2012)*. Brisbane: IEEE Press, 2012. 1–8. [doi: 10.1109/CEC.2012.6252954]
- [26] Zhou AM, Zhang QF, Zhang GX. Multi-objective evolutionary algorithm based on mixture Gaussian models. *Ruan Jian Xue Bao/ Journal of Software*, 2014,25(5):913–928 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4514.htm> [doi: 10.13328/j.cnki.jos.004514]
- [27] Chen CM, Chen Y, Zhang Q. Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization. In: *Proc. of the 2009 IEEE Congress on Evolutionary Computation (CEC 2009)*. IEEE, 2009. 209–216.
- [28] Nebro AJ, Durillo JJ. A study of the parallelization of the multi-objective metaheuristic MOEA/D. In: *Proc. of the 4th Int'l Conf. on Learning and Intelligent Optimization (LION 2010)*. 2010. 303–317.
- [29] Souza MZD, Pozo ATR. A GPU implementation of MOEA/D-ACO for the multi-objective traveling salesman problem. In: *Proc. of the Intelligent Systems*. IEEE, 2014. 324–329.
- [30] Xie CW, Zhang FL, Lu JB, Xiao C, Long GL. Multi-objective firefly algorithm based on multiply cooperative strategies. *Acta Electronica Sinica*, 2019,47(11):2359–2367 (in Chinese with English abstract).
- [31] Xie CW, Xiao C, Ding LX, Xia XW, Zhu JY, Zhang FL. HMOFA: A hybrid multi-objective firefly algorithm. *Ruan Jian Xue Bao/ Journal of Software*, 2018,29(4):1143–1162 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5275.htm> [doi: 10.13328/j.cnki.jos.005275]
- [32] Leung YW, Wang YP. An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Trans. on Evolutionary Computation*, 2001,5(1):41–53. [doi: 10.1109/4235.910464]
- [33] Deb K. *Multi-objective Optimization Using Evolutionary Algorithms*. Chichester: Wiley, 2001.
- [34] Jain H, Deb K. Parent to mean-centric self-adaptation in SBX operator for real-parameter optimization. In: *Proc. of the Int'l Conf. on Swarm, Evolutionary, and Memetic Computing*. Berlin, Heidelberg: Springer-Verlag, 2011. 299–306.
- [35] Xie CW, Wang ZJ, Xia XW. Multi-objective evolutionary algorithm based on archive-elite learning and opposition-based learning. *Chinese Journal of Computers*, 2017,40(3):757–772 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2017.00757]
- [36] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems. In: Fogel DB, ed. *Proc. of the IEEE Congress on Evolutionary Computation (CEC)*. Piscataway: IEEE Service Center, 2002. 825–830. [doi: 10.1109/CEC.2002.1007032]
- [37] Yu WW, Xie CW, Bi YZ, Xia XW, Li X, Ren KY, Zhao HR, Wang SF. Many-objective particle swarm optimization based on adaptive fuzzy dominance. *Acta Automatica Sinica*, 2018,44(12):2278–2289 (in Chinese with English abstract).
- [38] Liu Y, Gong D, Jing S, *et al.* A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Trans. on Cybernetics*, 2017,47(9):2689–2702.

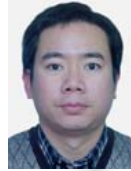
附中文参考文献:

- [1] 公茂果,焦李成,杨咚咚,马文萍.进化多目标算法研究.软件学报,2009,20(2):271–289. <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.03483]
- [7] 谢承旺,许雷,赵怀瑞,夏学文,魏波.应用精英反向学习的多目标烟花爆炸算法.电子学报,2016,44(5):1180–1188.
- [26] 周爱民,张青富,张桂戎.一种基于混合高斯模型的多目标进化算法.软件学报,2014,25(5):913–928. <http://www.jos.org.cn/1000-9825/4514.htm> [doi: 10.13328/j.cnki.jos.004514]
- [30] 谢承旺,张飞龙,陆建波,肖驰,龙广林.一种多策略协同的多目标萤火虫算法.电子学报,2019,47(11):2359–2367.

- [31] 谢承旺,肖驰,丁立新,夏学文,朱建勇,张飞龙.HMOFA:一种混合型多目标萤火虫算法.软件学报,2018,29(4):1143-1162. <http://www.jos.org.cn/1000-9825/5275.htm> [doi: 10.13328/j.cnki.jos.005275]
- [35] 谢承旺,王志杰,夏学文.应用档案精英学习和反向学习的多目标进化算法.计算机学报,2017,40(3):757-772. [doi: 10.11897/SP.J.1016.2017.00757]
- [37] 余伟伟,谢承旺,闭应洲,夏学文,李雄,任柯燕,赵怀瑞,王少锋.一种基于自适应模糊支配的高维多目标粒子群算法.自动化学报,2018,44(12):2278-2289.



谢承旺(1974—),男,湖北武汉人,博士,教授,CCF 高级会员,主要研究领域为智能计算,多目标优化.



汪慎文(1979—),男,博士,教授,主要研究领域为智能计算,机器学习.



余伟伟(1993—),男,硕士,主要研究领域为智能计算.



胡玉荣(1970—),女,博士,教授,主要研究领域为数据挖掘,智能计算.



闭应洲(1967—),男,博士,教授,CCF 专业会员,主要研究领域为智能计算,自然语言处理.