

控制与决策

Control and Decision

面向复杂超多目标优化问题的自适应增强学习进化算法

呼子宇, 李玉林, 魏之慧, 杨景明

引用本文:

呼子宇, 李玉林, 魏之慧, 杨景明. 面向复杂超多目标优化问题的自适应增强学习进化算法[J]. *控制与决策*, 2022, 37(11): 2849–2859.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.0725>

您可能感兴趣的其他文章

Articles you may be interested in

[基于多区域中心点预测的动态多目标优化算法](#)

Dynamic multi-objective optimization algorithm based on multi-regional center point prediction

控制与决策. 2022, 37(10): 2477–2486 <https://doi.org/10.13195/j.kzyjc.2021.0268>

[基于弱关联的自适应高维多目标进化算法](#)

A weak association-based adaptive evolutionary algorithm for many-objective optimization

控制与决策. 2021, 36(8): 1804–1814 <https://doi.org/10.13195/j.kzyjc.2019.1723>

[基于多种群分解预测的动态多目标引力搜索算法](#)

Dynamic multi-objective gravitational searching algorithm based on multi-population decomposition prediction

控制与决策. 2021, 36(12): 2910–2918 <https://doi.org/10.13195/j.kzyjc.2020.1002>

[基于向量角分解的高维多目标进化算法](#)

Many-objective evolutionary algorithm based on vector angle decomposition

控制与决策. 2021, 36(3): 761–768 <https://doi.org/10.13195/j.kzyjc.2019.0925>

[一种求解约束多目标问题的协作进化算法](#)

A collaborative evolutionary algorithm for solving constrained multi-objective problems

控制与决策. 2021, 36(11): 2656–2664 <https://doi.org/10.13195/j.kzyjc.2020.0791>

面向复杂超多目标优化问题的自适应增强学习进化算法

呼子宇[†], 李玉林, 魏之慧, 杨景明

(1. 燕山大学 电气工程学院, 河北 秦皇岛 066004;

2. 燕山大学 智能控制系统与智能装备教育部工程研究中心, 河北 秦皇岛 066004)

摘要: 在解决超多目标优化问题中, 基于分解的进化算法是一种较为有效的方法. 传统的分解方法依赖于一组均匀分布的参考向量, 它借助聚合函数将多目标优化问题分解为一组单目标子问题, 然后对这些子问题同时进行优化. 然而, 由于参考向量分布和 Pareto 前沿形状的不一致性, 导致这些预定义的参考向量在解决复杂超多目标优化问题时表现较差. 对此, 提出一种基于自适应增强学习的超多目标进化算法 (MaOEA-ABL). 该算法主要分为两个阶段: 第 1 阶段, 采用一种自适应增强学习算法对预定义的参考向量进行调整, 在学习过程中删除无用向量, 增加新的向量; 第 2 阶段, 设计一种对 Pareto 形状无偏好的分解方法. 为验证所提出算法的有效性, 选取具有复杂 Pareto 前沿的 MaF 系列测试函数进行仿真研究, 结果显示, MaOEA-ABL 算法的 IGD (inverted generational distance) 均值在 67% 的测试函数上超过了对比算法, 从而表明该算法在复杂超多目标优化问题中表现良好.

关键词: 超多目标优化; 进化算法; 自适应增强学习; 分解; 参考向量调整; 复杂 Pareto 前沿

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.0725

引用格式: 呼子宇, 李玉林, 魏之慧, 等. 面向复杂超多目标优化问题的自适应增强学习进化算法 [J]. 控制与决策, 2022, 37(11): 2849-2859.

Adaptive boosting learning evolutionary algorithm for complex many-objective optimization problems

HU Zi-yu[†], LI Yu-lin, WEI Zhi-hui, YANG Jing-ming

(1. School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China; 2. Engineering Research Center of the Ministry of Education for Intelligent Control System and Intelligent Equipment, Yanshan University, Qinhuangdao 066004, China)

Abstract: The evolutionary algorithm based on decomposition is an effective method in dealing with many-objective optimization problems. The traditional decomposition method relies on a set of uniformly distributed reference vectors, which decomposes the multi-objective optimization problem into a set of single-objective subproblems through aggregation functions, and then optimizes these subproblems simultaneously. However, these predefined reference vectors perform poorly in solving complex many-objective optimization problems because of the inconsistency of the distribution of reference vectors and the shape of the Pareto front. Aiming at the above problems, a many-objective evolutionary algorithm based on adaptive boosting learning (MaOEA-ABL) is proposed. The algorithm can be divided into two stages. In the first stage, an adaptive boosting learning algorithm is used to adjust the predefined reference vectors. In the learning process, useless vectors are deleted and new vectors are added. In the second stage, an unbiased decomposition method of Pareto shape is designed. Simulation has been conducted on the MaF test problems. The experimental results show that the IGD (inverted generational distance) mean value of MaOEA-ABL is better than that of the comparison algorithms in 67% of the test functions, which indicates that the MaOEA-ABL performs well in many-objective optimization problems with complex Pareto front.

Keywords: many-objective optimization; evolutionary algorithm; adaptive boosting learning; decomposition; adjustment of reference vector; complex Pareto front

收稿日期: 2021-04-25; 录用日期: 2021-08-18.

基金项目: 国家自然科学基金项目 (62003296); 河北省自然科学基金项目 (F2020203031); 河北省高等学校科学技术研究项目 (QN2020225).

责任编辑: 巩敦卫.

[†]通讯作者. E-mail: hzy@ysu.edu.cn.

0 引言

在科学研究和实际工程中,存在大量具有多个目标的优化问题,要求每一个目标达到最小(最大值),这种问题被称为多目标优化问题(multi-objective optimization problems, MOPs)^[1]. 其中,当优化问题的目标个数大于3时,称其为超多目标优化问题(many-objective optimization problems, MaOPs)^[2],用来解决MaOPs的进化算法被称为超多目标进化算法(many-objective evolutionary algorithm, MaOEA)^[3].

以目标函数最小化为例,无约束、静态、连续的超多目标优化问题定义如下:

$$\begin{aligned} \min F(X) &= (f_1(X), f_2(X), \dots, f_m(X)); \\ \text{s.t. } X &\in \Omega. \end{aligned} \quad (1)$$

其中: $X = (x_1, x_2, \dots, x_D)$ 是决策变量, Ω 是决策变量的可行域, m 是目标函数的个数, $F(\cdot)$ 是由决策变量 Ω 到目标函数空间 R^m 的映射.

超多目标优化问题的目标函数之间存在一定的冲突性,因此不存在唯一的最优解^[4]. 为了解决这些互相矛盾的目标,采用Pareto解集来定义超多目标优化问题的最优解. 以式(1)为例,超多目标优化问题中有关Pareto的定义如下.

定义1 支配(dominates): 设 $X = (x_1, x_2, \dots, x_n)$ 和 $Y = (y_1, y_2, \dots, y_n)$ 为决策空间的两个向量,若 $f_i(X) \leq f_i(Y)$, 当且仅当 $\forall i \in 1, 2, \dots, m$, 且 $\exists j \in 1, 2, \dots, m$ 使得 $f_j(X) < f_j(Y)$, 则称 X 支配 Y , 记为 $X \prec Y$.

定义2 Pareto最优解(Pareto optimal solution): 设 $X^* \in \Omega$ 为决策空间中的一个向量,若 $Y \prec X^*$, 当且仅当 $\neg \exists Y \in \Omega$, 则称 X^* 为非支配解,即Pareto最优解.

定义3 Pareto最优解集: 在决策空间 Ω 中,所有Pareto最优解构成的集合被称为Pareto最优解集(PS),表示为 $PS = \{X^* \in R^n | \neg \exists X \in R^n : X \succ X^*\}$.

在实际应用中,随着优化问题目标个数的增多,Pareto前沿形状复杂度的提高^[5],传统的进化算法的搜索能力降低^[6]. 其面临的挑战主要可分为以下3种.

1) 种群规模. 超多目标进化算法期望得到一组良好收敛性与多样性的Pareto近似解集^[7]. 然而,种群规模越大,直接造成算法搜索到的Pareto近似解集越无法均匀分布在Pareto前沿上^[8].

2) 算法性能. 收敛性与多样性是用来评价超多目标进化算法的两个重要性能指标^[9]. 但是,当优化问题目标个数增多时,在超多目标空间中算法的收敛

性与多样性难以同时维持,即当算法收敛性提高时反而会降低算法的多样性,反之亦然^[10].

3) 复杂前沿. 由于Pareto前沿的复杂性,预定义的参考向量分布与真实Pareto前沿形状高度不匹配,如图1所示. 由图1可知,预定义的参考向量中存在一部分消极参考向量(或称为无效参考向量). 目前,已有的超多目标进化算法在处理具有复杂前沿的问题时,无法取得较好的优化效果^[11].

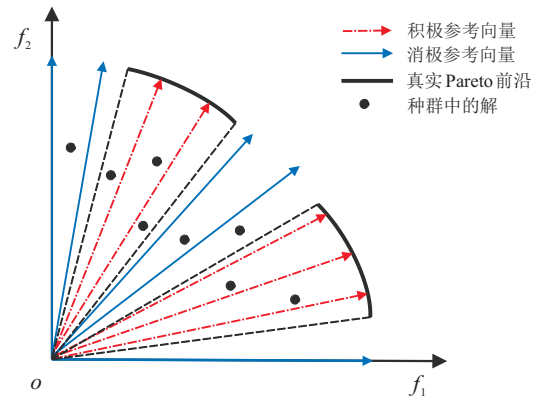


图1 预定义参考向量与真实前沿不匹配图

基于分解的进化算法是最典型的一种方法,该算法性能对参考向量的分布情况非常敏感,它们依赖参考向量的分布来提高算法的多样性. 然而,在处理复杂超多目标优化问题时,由于部分预定义生成的均匀分布参考向量与真实的Pareto前沿之间无交叉,在优化过程中,会出现部分区域内无解的现象^[12]. 因此,寻找合适的方法来估计参考向量单位平面上的有效面积显得尤为重要.

为了解决以上问题,一些学者在基于分解的超多目标进化算法基础上进行了研究. 主要可分为以下3类.

1) 分解方法的改进. Jiang等^[13]提出了一种基于改进分解的多目标进化算法来解决复杂前沿的优化问题,该算法采用两阶段策略,在目标维度较低时表现较好,维度增加时算法性能可能会降低. Asafuddoula等^[14]提出了基于自适应参考向量的增强分解进化算法,旨在同时解决规则与不规则的优化问题. 但是,该算法与学习周期相关的参数需要进一步研究以提高算法的性能.

2) 参考点与参考向量的调整. Jiang等^[15]设计了基于自适应权重分解的超多目标进化算法(MOEA/AWD),该算法使用的参考向量是在搜索过程中从种群中产生的,能够解决一些复杂前沿的优化问题,缺点在于参考向量的自适应生成可能会引起种群过早

收敛. Elarbi等^[16]利用一组预定义的法向边界点方向来近似不规则前沿的形状. Zhang等^[17]提出了一种改进的基于惩罚的边界交叉法(MPBI)和一种调整参考点(ARP)策略来处理具有复杂前沿面的超多目标优化问题,但该算法受到了参数设置的影响. Ge等^[18]针对复杂多目标优化问题,提出了基于级联聚类与参考点增量学习的进化算法(CLIA). 在基于级联的种群选择过程中使用参考向量进行聚类;在参考点增量学习中,使用基于级联聚类的过程反馈逐步获得适当的参考点分布.

3) 其他方法. 李学强等^[19]通过专门设计的一组权重得到适应值函数,同时采用一种随进化代数变化的杂交、变异概率来解决复杂多目标优化问题.

针对以上复杂超多目标优化问题面临的挑战,本文选取一种自适应增强学习(adaptive boosting learning)算法,利用参考向量的活动性训练一个强分类器,将预定义生成的均匀分布参考向量分为积极参考向量与消极参考向量. 参考向量的活动可以反映真实Pareto前沿在目标空间中的分布,从而估计参考向量单位平面上的有效面积. 并设计一种对Pareto形状无偏好的分解方法(convergence and diversity metric, CDM),该策略对真实Pareto前沿的曲率没有要求,可以求解前沿曲率不一致的问题,并且CDM分解所用的向量是经过基于自适应增强学习调整后的参考向量. 因此,本文所提出的方法能够适应于前沿形状不同的复杂超多目标优化问题.

1 自适应增强学习算法

1.1 自适应增强学习算法分析

“Adaptive boosting”(自适应增强)一般简称为“Adaboost”. 自适应增强算法是由Freund等^[20]提出的一种集成学习算法,它与单纯的机器学习算法不同,该算法采用多个机器学习器结合的方法来完成任任务. Adaboost算法可以用来解决分类问题的集成、特征选取的集成等问题.

自适应增强学习算法是迭代算法的一类,算法主要目的是在同一个训练集的基础上训练不同的弱分类器,然后选择某种方式将这些弱分类器组合成一个强分类器. Adaboost学习算法具有很强的自适应性,主要表现在:若一个样本在前一个基本分类器中被分类错误,则该样本的权值增大;若被分类正确,则减小该样本的权重,在训练下一个基本分类器时还会使用到该样本. 在每一次迭代过程中,都会有一个新的弱分类器加入进来,若完成最大的迭代次数或者

错误率达到设定值时,则最终的强分类器将被确定下来. 总之,自适应增强学习就是把多个具有一定权值的弱分类器累加得到一个强分类器的过程.

1.2 自适应增强学习算法流程

1) 对参考向量集的权重分布进行初始化. 设有 N 个参考向量训练样本,给每一个样本赋予相同的权重 $\omega_1 = 1/N$.

2) 开始训练弱分类器 h_i . 对于某一个参考向量而言,若被弱分类器错误分类,则在下一次训练中,该向量的权重随之增大;若被正确分类,则它所对应的权重要减小. 权重经过更新后的参考向量集将会用于下一个分类器的训练过程,训练样本需要经过多次的迭代.

3) 将每次迭代训练得到的弱分类器组合成一个强分类器. 在所有弱分类器训练结束后,分类误差率小的弱分类器占据的权重将会被增大,目的是为了使其在最终的分类器中发挥较大的作用;分类误差率大的弱分类器占据的权重将会被降低,这样它在最终的分类器中将会发挥较小的作用. 自适应增强学习算法的步骤如下所示.

算法1 自适应增强学习算法框架.

输入: 迭代次数 K , N 个训练参考向量数据 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, $y \in \{-1, +1\}$, 基础分类器 $G_t(x)$;

输出: 参考向量的强分类器.

step 1: 初始化训练参考向量的权重 $\omega_i^{(i)} = 1/N$, $i = 1, 2, \dots, N$;

step 2: for $t = 1, \dots, K$ do

step 3: 根据训练参考向量的分布获得基础分类器

$$G_t(x) = \arg \min_{G(x)} \sum_{i=1}^N \omega_i^{(t)} (y_i \neq G(x_i));$$

step 4: 计算权重分布 $\omega_i^{(t)}$ 所对应的分类器 $G_t(x)$ 错误率

$$\varepsilon_t = \frac{\sum_{i=1}^N \omega_i^{(t)} (y_i \neq G(x_i))}{\sum_{i=1}^N \omega_i^{(t)}};$$

step 5: 计算分类器 $G_t(x)$ 的权重系数

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t};$$

step 6: 更新训练参考向量的权重

$$\omega_i^{(t+1)} = \frac{\omega_i^{(t)} e^{-y_i \alpha_t G_t(x_i)}}{Z(t)}, \quad i = 1, 2, \dots, N,$$

其中 $Z^{(t)} = \sum_{i=1}^N \omega_i^{(t)} e^{-y_i \alpha_t G_t(x_i)}$ 是一个标准化因子;

step 7: 计算最终参考向量的强分类器

$$G(x) = \text{sign} \left[\sum_{t=1}^T \alpha_t G_t(x) \right];$$

step 8: end for.

2 基于自适应增强学习的超多目标进化算法 MaOEA-ABL

2.1 基于自适应增强学习的参考向量调整策略

基于分解的超多目标进化算法的性能对于参考向量的分布十分敏感. 在解决复杂超多目标优化问题时, 由于 Pareto 前沿具有不同的特性, 例如非连续、倒置等形状, 预定义的参考向量与真实 Pareto 前沿无法相交, 这样只有一部分参考向量在种群向真实的 Pareto 前沿进化时起到积极的引导作用. 因此, 本文设计的算法 MaOEA-ABL 采用了基于自适应增强学习的参考向量调整策略.

在基于分解的超多目标进化算法中, 参考向量的

活动可以反映真实的 Pareto 前沿在目标空间的分布情况. 基于上述情况, MaOEA-ABL 利用参考向量与种群个体的关联特性, 将参考向量分为积极向量和消极向量; 再利用自适应增强学习算法对积极和消极向量进行训练分类, 将参考向量的超平面划分为积极区域和消极区域; 识别引导种群进化的参考向量有效面积, 删除消极向量, 在积极区域内增加向量个数到原来的向量集大小.

图2给出了基于自适应增强学习的参考向量调整的具体过程, 以 MaF3 测试函数的 Pareto 前沿形状为例, 图2(a) 两种颜色不同的点分别表示在种群进化过程中的积极向量与消极向量. 图2(b)~图2(f) 表示参考向量自适应调整的过程, 利用种群中个体与参考向量之间的关联特性或者称为参考向量的活动性能, 将参考向量分为积极和消极两类, 通过自适应增强学习来训练一个强分类器, 识别将要保留下的区域和删除的向量. 由图2可以看出, 一次自适应增强学习很难得到足够数量的积极参考向量, 因此, 在算法整个搜索过程中, 需要进行多次自适应增强学习.

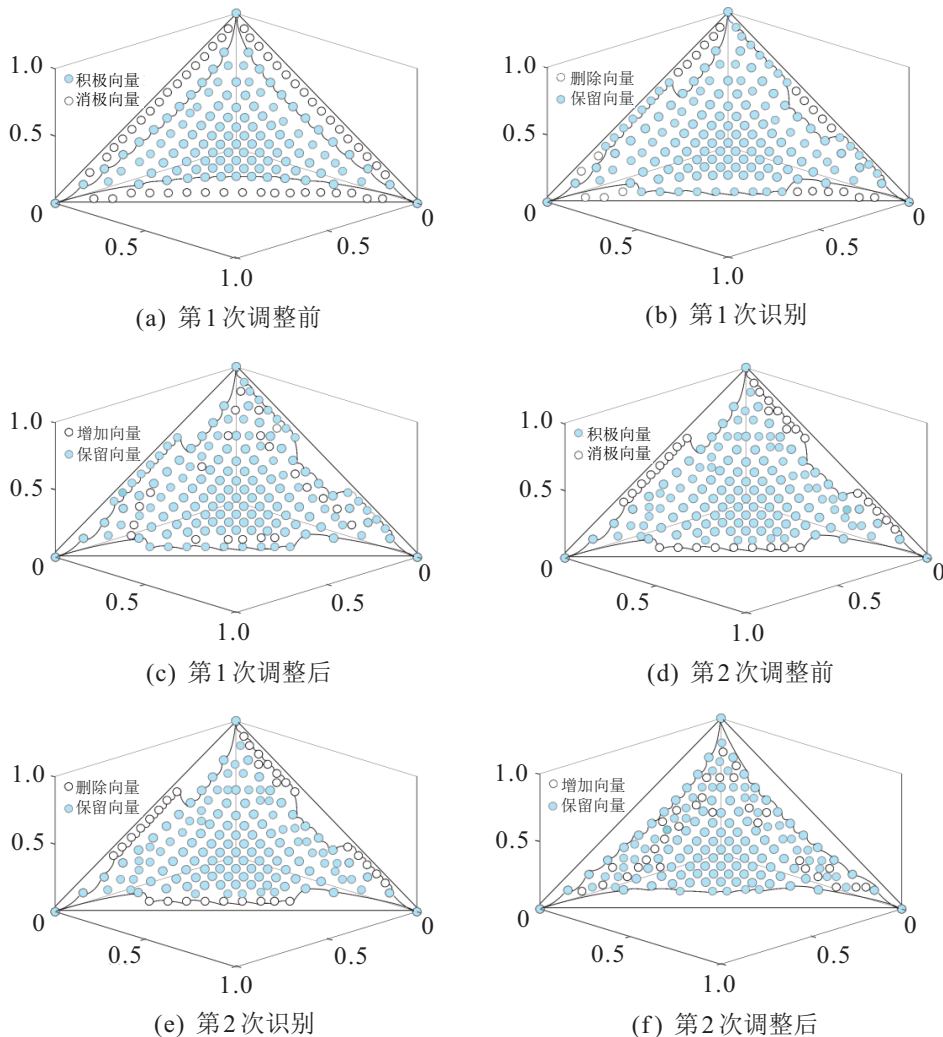


图2 基于自适应增强学习的参考向量调整策略

为了更加清晰地描述基于自适应增强学习的参考向量调整策略,其伪代码如下所示.

算法 2 基于自适应增强学习的参考向量调整策略框架.

输入: 种群大小 N , 预定义参考向量集 W_0 , 自适应调整周期 t , 自适应函数评价次数 T_FES , 集合 Ω ;

输出: 第 $t + 1$ 代参考向量集 W_{t+1} .

step 1: while $FES \geq T_Gen$

step 2: If $Gen(mod) == 0$ then

step 3: for $i = 1, \dots, N$ do

step 4: for $j = 1, \dots, N$ do

step 5: 计算第 i 个体与第 j 个向量 λ_j 之间的夹角 φ_i ;

step 6: $\varphi_i = \arccos \frac{F(x_i)\lambda_j}{\|F(x_i)\|}$;

step 7: $\Phi = \Phi \cup \varphi_i$;

step 8: 找出 Φ 中的最小值, 即与第 i 个体相关联的向量;

step 9: end for

step 10: end for

step 11: 与个体相关联的向量称为积极向量 W , 反之称为消极向量 \bar{W} ;

step 12: 将积极向量 W (标记为 1) 和消极向量 \bar{W} (标记为 -1) 作为样本进行训练, 利用自适应增强学习算法训练出参考向量的强分类器, 识别出参考向量超平面的有效区域;

step 13: 在有效区域内随机生成参考向量, 直到向量个数达到 N ;

step 14: end if

step 15: end while.

2.2 基于自适应 CDM 分解的个体更新策略

在解决复杂超多目标优化问题时, 真实 Pareto 前沿的曲率可能是不规则的, 如凸、凹或更复杂. 为了解决这些问题, MaOEA-ABL 算法采用一种基于 CDM 分解的个体更新策略. 该策略对真实 Pareto 前沿的曲率没有要求, 可以解决不同曲率的问题, 并且在 CDM 分解所用的向量是经过基于自适应增强学习调整后的参考向量, 具体如下所示:

$$CDM(x, \lambda) =$$

$$CM(x) + DM(x, \lambda) =$$

$$\frac{\sum_{i=1}^m f_i(x)}{m} + \alpha \frac{\sqrt{\|f(x)\|_2^2 \|\lambda\|_2^2 - (f(x)^T \lambda)^2}}{\|\lambda\|_2} \sin(f(x), \lambda) =$$

$$\frac{\sum_{i=1}^m f_i(x)}{m} + \alpha \frac{\sqrt{\|f(x)\|_2^2 \|\lambda\|_2^2 - (f(x)^T \lambda)^2}}{\|\lambda\|_2}. \quad (2)$$

其中: x 为种群中的一个随机个体; m 为目标函数的个数; $f_i(x)$ 为个体 x 所对应的第 i 个目标函数值; λ 为离个体 x 最近的参考向量; α 为系数, 用来平衡种群个体的收敛性与多样性 (在这里与 PBI 方法中设置值相同, 设置为 5).

图 3 给出了 CDM 搜索方向分解方法示意图. 由图 3 可知: CM 计算了所有目标函数的平均值, CM 值越小, 代表边界个体离真实前沿越近, 个体的收敛程度越好, 可以引导种群中的个体向垂直于单位平面的方向进化; DM 表示个体到最近参考向量的欧氏距离, 代表了个体与参考向量之间的分布误差, DM 值越小, 代表个体与向量之间的距离越近, 种群个体的多样性越好. 在 MaOEA-ABL 算法流程中, 基于 CDM 分解的个体更新策略主要用来进行环境选择, 个体的 CDM 聚合函数值越小, 代表个体的性能越好, 更期望被保留下来进行下一次进化. 利用该策略可以提高算法的收敛性.

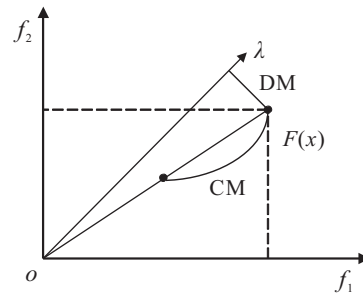


图 3 自适应 CDM 分解方法示意图

不同分解方法对于真实 Pareto 前沿曲率的偏好是不同的, 为了更加清晰地看出 CDM 分解方法的优势, 图 4 给出了 3 种传统分解方法: 权重求和法 (weighted sum, WS)、切比雪夫法 (tchebycheff, TCH)、基于惩罚的边界交叉法 (penalty-based boundary intersection, PBI), 以及 CDM 分解方法的等高线示意图.

以最小化超多目标优化问题为例, 权重求和法在前沿非凸的情况下, 无法搜索到全部的 Pareto 最优解; 通过切比雪夫法得到的近似前沿形状是不光滑的. 如图 4(c) 所示, 基于惩罚的边界交叉法的等高线在凹同心弧周围波动, 表明该方法对于凹面 Pareto 前沿有偏好, 在前沿为凸或者线性形状时则无法起到积极的作用. 从图 4(d) 可以看出, 自适应 CDM 分解方法的等高线在水平线左右波动, 对于不同的 Pareto 前沿曲率的问题没有偏好.

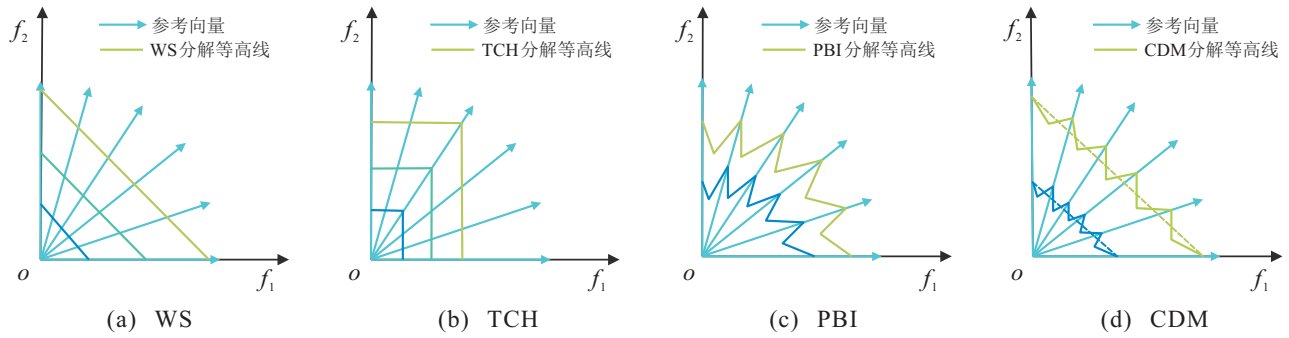


图4 4种分解方法等高线示意图

基于自适应CDM分解的个体更新策略步骤的伪代码如算法3所示.

算法3 基于自适应增强学习的参考向量调整策略框架.

输入: 种群 P , 参考向量 λ , 对应的邻域索引值 $B(i)$, 父代个体 x , 子代个体 y ;

输出: 精英解集 P' .

step 1: if $g^{\text{CDM}}(x_j|\lambda_i, \alpha) > g^{\text{CDM}}(y|\lambda_i, \alpha)$ then

step 2: $x_j = y$;

step 3: $F(x_j) = F(y)$;

step 4: end if

step 5: return P' .

2.3 MaOEA-ABL算法步骤

为了更加清晰描述MaOEA-ABL算法的过程, 算法伪代码如下所示.

算法4 MaOEA-ABL算法框架.

输入: 种群大小 N , 邻域大小 T , 函数评价次数 $FES = N$, 最大函数评价次数 maxGen , 自适应周期 t ;

输出: 精英解集 P .

step 1: 初始化种群 P_0 , 并对每个个体进行评价;

step 2: 生成参考向量 W_0 , 计算每个向量的邻域 $B(i)$;

step 3: while $FES \leq \text{maxGen}$

step 4: for $i = 1, \dots, N$ do

step 5: 交配选择得到 P' ;

step 6: 对 P' 进行交叉变异;

step 7: 更新理想点 Z^{\min} ;

step 8: 根据算法3更新种群 P ;

step 9: end for

step 10: if $FES \geq T_Gen$ 且 $\text{Gen}(\text{mod})t == 0$

then

step 11: 参考向量自适应得到 W ;

step 12: 更新向量邻域;

step 13: end if

step 14: end while

step 15: return P_0 .

2.4 MaOEA-ABL算法计算复杂度分析

MaOEA-ABL算法的计算复杂度主要集中在参考向量自适应增强学习过程和基于自适应CDM分解的个体更新策略中. 设种群中个体数为 N , 目标函数个数为 M . 在参考向量自适应增强学习算法中, 计算了个体与向量之间的夹角, 计算复杂度为 $O(MN^2)$; 自适应增强学习算法的计算复杂度为 $O(\log(N))$, 所以整个参考向量自适应增强学习过程的计算复杂度为 $O(\log(N) \times MN^2)$. 基于自适应CDM分解的个体更新策略的计算复杂度为 $O(MN^2)$. 因此, MaOEA-ABL算法的计算复杂度为 $O(\log(N) \times MN^2)$.

3 数值仿真与结果分析

3.1 仿真参数设置

为了验证MaOEA-ABL算法的有效性, 本文采用具有复杂多变Pareto前沿且目标可拓展的MaF系列测试集进行仿真.

MaF系列函数的表达式和真实Pareto参见文献[21]. 本文测试函数考虑目标个数 M 为 {5, 8, 10, 15}. 为了方便对算法结果进行比较, 所有算法的种群数量大小根据VaEA算法^[22]的种群大小进行微调, 在目标个数为 {5, 8, 10, 15} 时, 分别设置为 {210, 156, 275, 136}. 所有测试函数以300000次函数评价次数作为仿真的终止条件.

为了验证MaOEA-ABL算法性能, 选取4个算法进行对比, 分别为ARMOEA^[23]、MaOEAIT^[24]、VaEA、PICEAg^[25].

算法采用反向世代距离 (inverted generational distance, IGD) 和超体积指标 (hyper volume, HV) 作为性能评价指标. IGD作为一种综合性评价指标, 用来

表 1 算法 IGD 指标统计值

函数	M	MaOEA-ABL	ARMOEA	MaOEAIT	VaEA	PICEAg
MaF1	5	1.02e-01 (3.74e-04)	1.14e-01 (1.95e-03)	3.26e-01 (7.14e-02)	1.07e-01 (6.69e-04)	1.04e-01 (5.41e-04)
	8	2.59e-01 (6.70e-03)	2.16e-01 (1.53e-03)	6.10e-01 (1.95e-01)	2.10e-01 (1.47e-03)	2.22e-01 (3.04e-03)
	10	2.46e-01 (1.57e-03)	2.82e-01 (2.97e-02)	5.21e-01 (1.12e-01)	2.70e-01 (1.84e-03)	2.53e-01 (2.89e-03)
	15	2.85e-03 (2.92e-03)	3.69e-01 (1.05e-02)	7.47e-01 (2.10e-01)	3.02e-01 (1.75e-02)	3.00e-01 (4.11e-03)
MaF2	5	7.42e-02 (1.02e-03)	9.46e-02 (1.30e-03)	1.20e-01 (3.33e-02)	8.94e-02 (1.63e-03)	9.12e-2 (1.93e-03)
	8	1.60e-01 (1.72e-03)	1.74e-01 (5.35e-03)	3.42e-01 (5.62e-02)	1.70e-01 (2.59e-03)	2.70e-01 (2.49e-02)
	10	2.68e-01 (3.95e-03)	2.33e-01 (2.68e-02)	3.62e-01 (6.54e-02)	2.19e-01 (3.82e-03)	3.28e-01 (1.82e-02)
	15	1.91e-01 (1.52e-03)	2.53e-01 (1.99e-02)	4.53e-01 (4.94e-02)	2.14e-01 (2.27e-03)	4.14e-01 (2.56e-02)
MaF3	5	8.27e-02 (1.58e-03)	7.31e-02 (1.30e-03)	1.15e+03 (1.72e+03)	3.58e-01 (5.93e-01)	1.19e-01 (2.00e-02)
	8	7.56e-02 (2.08e-03)	9.84e-02 (3.75e-03)	2.58e+08 (8.18e+08)	4.34e+02 (6.31e+02)	5.54e+01 (1.09e+02)
	10	9.42e-01 (6.21e-01)	1.12e+00 (3.23e+00)	9.79e+08 (2.02e+09)	2.96e+03 (4.31e+03)	3.90e+02 (8.43e+02)
	15	9.67e-01 (5.84e-01)	6.11e+00 (6.97e+00)	2.70e+08 (8.33e+08)	5.29e+03 (6.63e+03)	1.64e+03 (2.58e+03)
MaF4	5	2.13e+00 (6.98e-01)	1.54e+02 (7.05e-01)	3.38e+02 (3.16e+02)	2.04e+00 (3.02e-01)	2.32e+00 (3.25e-01)
	8	1.82e+01 (1.35e+00)	3.47e+01 (2.96e+00)	2.29e+03 (2.81e+03)	1.63e+01 (1.17e+00)	1.67e+01 (5.26e+00)
	10	2.76e+01 (1.29e+00)	1.62e+02 (4.06e+00)	1.30e+04 (2.23e+04)	5.29e+01 (1.70e+00)	6.57e+01 (2.97e+00)
	15	8.01e+02 (1.73e+02)	5.98e+03 (5.81e+03)	2.32e+05 (2.86e+05)	1.88e+03 (2.07e+02)	1.99e+03 (1.87e+02)
MaF5	5	7.59e-02 (1.58e-03)	7.31e-02 (1.30e-03)	1.04e+01 (2.48e+00)	1.75e+00 (1.73e-02)	1.65e+00 (5.88e-03)
	8	7.56e-02 (2.08e-03)	9.84e-02 (3.75e-03)	6.83e+01 (1.89e+01)	1.47e+01 (5.76e-01)	2.38e+01 (9.30e+00)
	10	9.42e-01 (6.21e-01)	1.12e+00 (3.23e+00)	2.46e+02 (8.34e+01)	4.53e+01 (9.05e-01)	1.03e+02 (3.52e+01)
	15	9.67e-01 (5.84e+00)	6.11e+00 (6.97e+00)	5.39e+03 (2.71e+03)	1.53e+03 (1.08e+02)	2.61e+03 (3.69e+02)
MaF7	5	6.06e-03 (3.81e-04)	2.00e-03 (1.17e-05)	1.82e-03 (1.11e-05)	2.52e-03 (7.40e-05)	4.35e-01 (2.11e-01)
	8	9.83e-01 (3.62e-02)	9.80e-01 (4.13e-02)	1.72e+01 (3.08e+00)	7.10e-01 (9.88e-03)	4.23e+00 (5.80e-02)
	10	5.98e-01 (2.08e-02)	3.07e+00 (8.74e-01)	2.53e+01 (3.09e+00)	1.10e+00 (3.01e-02)	5.51e+00 (7.34e-02)
	15	7.45e-01 (1.06e-01)	4.62e+00 (5.92e-01)	4.51e+01 (3.27e+00)	2.28e+00 (1.65e-01)	1.11e+01 (8.64e-02)
MaF8	5	9.93e-02 (4.18e-02)	8.89e-02 (1.94e-03)	7.07e+01 (1.16e+02)	2.70e-01 (5.10e-03)	1.59e-01 (4.45e-02)
	8	1.02e-01 (3.59e-03)	1.47e-01 (5.70e-03)	8.87e+01 (1.33e+02)	7.21e+00 (6.52e+00)	2.59e-01 (7.75e-02)
	10	6.29e-02 (3.03e-03)	1.21e-01 (3.27e-03)	5.00e+01 (8.19e+01)	8.01e-01 (6.90e-02)	2.83e-01 (9.62e-02)
	15	2.48e-02 (1.67e-03)	2.22e-01 (7.93e-03)	1.06e+02 (1.11e+02)	7.39e-01 (4.83e+01)	3.06e-01 (3.26e-02)
MaF9	5	3.14e-01 (2.62e-02)	8.85e-02 (4.22e-03)	4.19e+01 (2.78e+01)	2.70e-01 (5.10e-03)	1.44e-01 (5.06e-02)
	8	5.62e-01 (6.08e-02)	2.08e-01 (9.93e-03)	1.65e+02 (1.32e+02)	7.21e+00 (6.52e+00)	3.23e+00 (1.91e+00)
	10	1.25e-01 (3.86e-03)	1.60e-01 (4.33e-03)	1.60e+02 (1.12e+02)	8.01e-01 (6.90e-02)	4.94e+00 (5.99e+00)
	15	1.28e-01 (3.94e-03)	2.01e-01 (5.71e-03)	1.44e+02 (1.49e+02)	7.39e-01 (4.83e+01)	6.72e+00 (5.04e+00)
MaF10	5	4.25e-01 (5.63e-02)	2.59e-01 (4.69e-03)	2.06e+00 (2.07e+0)	2.70e-01 (5.10e-03)	3.92e-01 (4.83e-03)
	8	6.81e-01 (2.17e-02)	8.99e-01 (2.85e-02)	2.87e+00 (3.58e-01)	1.34e+00 (1.31e-01)	1.62e+00 (9.59e-02)
	10	5.06e-01 (3.83e-02)	1.30e+00 (4.58e-02)	3.30e+00 (1.85e-01)	1.32e+00 (8.40e-02)	1.68e+00 (1.06e-01)
	15	6.97e-01 (2.45e-02)	1.85e+00 (3.96e-02)	4.35e+00 (4.83e-02)	2.14e+00 (1.55e-01)	2.43e+00 (1.69e-01)
MaF11	5	7.20e-01 (5.81e-03)	3.93e-01 (6.23e-03)	1.77e+00 (8.48e-01)	3.89e-01 (1.86e-03)	4.24e-01 (4.98e-03)
	8	9.68e-01 (8.07e-02)	9.31e-01 (1.06e-02)	3.13e+00 (6.60e-01)	9.35e-01 (1.57e-02)	1.33e+00 (8.88e-02)
	10	9.25e-01 (7.05e-02)	1.33e+00 (1.30e-01)	5.20e+00 (3.04e+00)	1.30e+00 (7.75e-02)	1.73e+00 (1.23e-01)
	15	9.44e-01 (4.97e-02)	1.72e+00 (5.60e-02)	7.91e+00 (2.72e+00)	1.75e+00 (5.65e-02)	1.44e+01 (5.35e-01)
MaF12	5	9.92e-01 (8.05e-03)	9.44e-01 (8.62e-03)	2.14e+00 (7.54e-02)	9.31e-01 (7.98e-03)	9.12e-01 (6.12e-03)
	8	9.01e-01 (5.42e-02)	2.92e+00 (1.01e-01)	5.79e+00 (3.36e-01)	3.04e+00 (5.97e-02)	3.56e+00 (3.49e-01)
	10	8.49e-01 (3.27e-02)	5.65e+00 (4.18e-01)	7.91e+00 (2.92e-01)	4.88e+00 (4.83e-02)	6.36e+00 (5.87e-01)
	15	9.60e-01 (3.64e-03)	9.09e+00 (3.28e-02)	1.63e+01 (6.90e-01)	7.74e+00 (6.74e-02)	1.44e+01 (5.35e-01)
1 st / 2 nd / 3 rd		29/6/4	8/20/13	2/3/5	6/12/10	1/3/12

表2 算法HV指标统计值

函数	M	MaOEA-ABL	ARMOEA	MaOEAIT	VaEA	PICEAg
MaF1	5	1.54e-03(1.52e-04)	1.14e-02(2.56e-04)	1.75e-03(1.36e-03)	1.16e-02(1.40e-04)	4.65e-03(8.03e-04)
	8	7.58e-02(1.14e-05)	2.67e-05(4.11e-06)	8.55e-02(3.67e-06)	2.62e-05(5.61e-06)	2.12e-05(4.23e-06)
	10	3.49e-03(2.18e-08)	2.00e-07(4.22e-07)	1.47e-08(2.44e-08)	2.97e-07(6.68e-07)	7.84e-07(4.05e-06)
	15	1.75e-03(2.92e-03)	2.82e-04(6.07e-01)	2.48e-04(6.72e-03)	3.46e-04(5.39e-02)	0.00e+00(0.00e+00)
MaF2	5	2.08e-01(1.24e-03)	1.89e-01(1.61e-03)	1.41e-01(2.09e-02)	1.92e-01(1.27e-03)	1.57e-01(2.99e-03)
	8	2.03e-01(3.75e-03)	2.06e-01(3.30e-03)	1.42e-01(3.34e-02)	2.19e-01(2.60e-03)	1.91e-01(3.01e-03)
	10	3.19e-01(5.82e-04)	1.77e-01(1.49e-02)	1.43e-01(5.27e-02)	1.93e-01(4.27e-03)	1.08e-01(4.62e-03)
	15	2.76e-01(3.07e-03)	1.83e-01(3.97e-03)	1.06e-01(2.90e-02)	1.84e-01(4.88e-03)	1.51e-01(3.64e-03)
MaF3	5	7.57e-01(1.81e-04)	9.99e-01(6.22e-05)	6.64e-02(2.57e-01)	8.57e-01(2.64e-01)	7.04e-02(6.83e-01)
	8	1.02e+00(1.06e-05)	9.98e-01(1.29e-05)	6.07e-02(8.54e-04)	7.59e-01(3.45e-04)	5.98e-02(1.71e-01)
	10	9.52e-01(6.99e-02)	8.99e-01(3.16e-01)	4.96e-02(7.18e-01)	7.01e-02(6.82e-01)	6.88e-02(3.57e-01)
	15	8.35e-01(4.23e-02)	4.55e-01(4.77e-01)	4.64e-02(6.16e-01)	2.13e-02(4.90e-01)	0.00e+00(0.00e+00)
MaF4	5	9.20e-01(3.74e-03)	1.05e+00(1.58e-03)	9.15e-03(1.72e-01)	1.07e-01(1.84e-02)	7.84e-05(5.61e-02)
	8	6.08e-02(1.71e-04)	5.71e-02(6.05e-03)	3.47e-02(8.18e-02)	1.72e-03(3.03e-04)	9.73e-05(4.33e-02)
	10	5.98e-02(3.56e-02)	1.12e-02(4.13e-02)	6.91e-02(2.02e-02)	9.04e-05(2.02e-02)	7.88e-05(6.93e-02)
	15	9.67e-02(5.29e-02)	3.42e-02(7.99e-02)	1.42e-02(8.33e-02)	3.38e-08(5.49e-02)	0.00e+00(0.00e+00)
MaF5	5	8.27e-02(1.72e-01)	9.31e-02(1.69e-01)	1.00e-01(1.09e-01)	7.89e-02(2.32e-01)	8.26e-02(7.61e-01)
	8	9.36e-01(2.08e-03)	7.04e-02(3.75e-02)	7.87e-02(3.12e-02)	8.95e-01(6.61e-03)	7.76e-01(3.02e-02)
	10	9.58e-01(2.53e-02)	7.12e-02(3.23e-02)	5.92e-02(6.97e-02)	9.22e-01(7.45e-02)	9.02e-01(4.79e-02)
	15	9.71e-01(3.96e-03)	3.29e-02(6.57e-02)	1.53e-02(2.23e-02)	9.57e-01(5.89e-03)	7.59e-01(2.76e-02)
MaF7	5	2.14e-01(2.58e-03)	2.54e-01(1.76e-03)	6.09e-02(6.87e-02)	2.48e-01(2.89e-03)	2.50e-01(4.97e-03)
	8	1.65e-01(2.89e-03)	1.82e-01(2.77e-03)	1.74e-02(8.34e-03)	1.87e-01(2.58e-03)	1.50e-01(2.52e-03)
	10	2.29e-01(1.26e-03)	1.37e-01(2.57e-02)	1.04e-02(9.71e-02)	1.36e-01(7.72e-03)	1.74e-01(2.52e-02)
	15	4.79e-01(3.47e-03)	5.35e-02(2.33e-02)	8.46e-03(4.29e-02)	1.00e-01(4.97e-03)	1.18e-01(1.92e-03)
MaF8	5	1.18e-01(5.62e-02)	1.23e-01(3.72e-04)	4.18e-03(1.16e-02)	1.25e-01(2.53e-04)	4.07e-02(5.72e-03)
	8	6.41e-01(1.09e-04)	2.98e-02(3.83e-04)	2.28e-04(2.94e-04)	3.04e-02(1.65e-04)	2.27e-02(3.11e-03)
	10	1.08e-02(3.84e-05)	1.03e-02(7.87e-05)	5.66e-04(5.47e-04)	1.10e-02(1.04e-05)	6.81e-03(9.13e-03)
	15	7.25e-01(1.38e-05)	5.50e-04(2.06e-05)	2.77e-07(6.23e-05)	5.65e-04(1.90e-05)	2.70e-04(5.78e-05)
MaF9	5	3.14e-01(2.62e-02)	3.17e-01(2.12e-03)	3.70e-03(1.17e-02)	1.95e-01(7.93e-02)	5.96e-04(4.26e-02)
	8	5.81e-01(5.24e-04)	4.22e-02(9.68e-04)	5.35e-03(6.94e-03)	4.31e-02(1.14e-03)	6.00e-04(7.32e-04)
	10	7.49e-01(1.24e-04)	1.63e-02(1.64e-04)	6.27e-03(7.08e-02)	1.55e-02(3.27e-03)	4.73e-04(6.85e-04)
	15	5.61e-01(2.05e-05)	1.08e-03(3.09e-05)	4.28e-03(5.62e-02)	7.78e-04(3.38e-04)	4.99e-04(1.08e-04)
MaF10	5	8.97e-01(5.75e-02)	9.12e-01(2.94e-02)	1.81e-01(7.07e-02)	9.21e-01(1.99e-02)	9.04e-01(6.98e-02)
	8	9.64e-01(4.26e-02)	9.88e-01(2.15e-02)	7.70e-02(5.87e-02)	6.14e-01(4.63e-02)	9.98e-01(2.73e-04)
	10	9.74e-01(5.75e-02)	9.62e-01(8.56e-02)	4.00e-02(2.47e-02)	8.27e-01(5.48e-02)	5.62e-01(3.16e-01)
	15	9.98e-01(3.26e-04)	9.97e-01(6.16e-04)	4.43e-02(1.09e-02)	5.95e-01(5.16e-02)	9.26e-01(7.55e-02)
MaF11	5	1.04e+00(1.01e-03)	9.95e-01(1.08e-03)	5.91e-01(7.46e-02)	9.88e-01(2.02e-03)	9.58e-01(4.81e-03)
	8	9.07e-01(2.15e-03)	9.97e-01(1.24e-03)	5.90e-01(5.07e-02)	9.91e-01(2.14e-03)	9.92e-01(3.03e-03)
	10	9.98e-01(1.57e-03)	9.97e-01(2.08e-03)	5.54e-01(1.18e-01)	9.94e-01(1.71e-03)	9.52e-01(6.80e-02)
	15	9.97e-01(1.67e-03)	9.95e-01(2.36e-03)	5.25e-01(9.19e-02)	9.92e-01(3.24e-03)	9.81e-01(8.19e-03)
MaF12	5	6.85e-01(9.01e-02)	7.36e-01(7.72e-03)	2.76e-01(1.47e-02)	7.23e-01(8.63e-03)	7.89e-01(3.27e-03)
	8	7.62e-01(6.56e-02)	7.99e-01(4.45e-02)	2.95e-01(1.21e-02)	7.38e-01(7.03e-02)	7.49e-01(5.09e-02)
	10	8.89e-01(1.28e-02)	8.42e-01(1.74e-02)	3.25e-01(1.35e-02)	7.46e-01(6.43e-02)	7.57e-01(8.17e-02)
	15	9.16e-01(2.59e-02)	8.07e-01(4.77e-02)	2.85e-01(3.42e-02)	7.34e-01(7.31e-02)	6.11e-01(6.95e-02)
$1^{st}/2^{nd}/3^{rd}$		29/6/7	7/20/13	1/3/2	6/12/10	1/3/12

反映算法搜索到解集的收敛性与多样性的好坏, IGD 越小, 表示算法得到的近似 Pareto 解集与真实 Pareto 前沿之间的距离越近且分布越均匀. HV 是一种用来衡量算法解集收敛性与多样性的综合性能指标, HV 越大, 表示在目标空间中非支配解覆盖的区域越大.

在 5 个算法中, 模拟二项式交叉 (simulated binary crossover, SBX) 和多项式变异 (polynomial mutation, PM) 用于生成子代个体, 其操作参数设置为: $p_c = 1.0$, $\eta_c = 30$ (p_c 为交叉概率, η_c 为交叉参数); $p_m = 1/n$, $\eta_m = 1.0$ (p_m 为变异概率, η_m 为变异参数). 其他算法参数按照原始文献设置. 本文 MaOEA-ABL 算法相关的参数设置为: $T_Gen = 5000$, 自适应周期 t 为 1000.

3.2 仿真结果数值分析

表 1 和表 2 给出了 5 个算法独立运行 30 次得到的 IGD 与 HV 的均值及标准差 (括号内为标准差). 每一个测试函数的各项最优值用粗体显示, 并在表格最后一行统计了各算法获得最优值的个数. 从表 1 和表 2 可以看出, 本文所提出算法 MaOEA-ABL 在 44 个测试函数中分别得到了 29 个 IGD 最小值和 30 个 HV 最大值, 说明 MaOEA-ABL 算法比其他 4 个算法表现更好. 尤其在处理目标个数较多时, MaOEA-ABL 算法的优势更加明显. 与其他 4 个算法相比, 本文算法采用自适应增强学习算法对预定义参考向量进行调整, 估计参考向量单位平面上的有效面积, 通过删除旧向量、增加新向量, 引导种群向真实 Pareto 前沿进化, 并设计了对 Pareto 形状无偏好的分解方法, 从而提高了

种群的多样性和收敛性.

3.3 仿真结果图分析

为了更加形象地比较算法的性能, 图 5 和图 6 绘制了 MaOEA-ABL、ARMOEA、MaOEAIT 和 VaEA 四种算法在 5 和 10 个目标的部分 MaF 系列测试函数上的最优解的平行坐标系图. 由于 PICEAg 的性能较差, 此处不再给出该算法的解集图.

MaF3 测试函数的真实 Pareto 前沿是一个强凹的平面. 从图 5 中可以看出, MaOEA-ABL 和 ARMOEA 由于自身的自适应性, 获得了收敛性与多样性较好的解集. VaEA 算法的性能较差, 在某些目标上没有达到最优收敛. 对于 MaF3 而言, 由于预定义的参考向量与前沿之间有一部分无交叉, 参考向量集中有一部分无效向量, 对算法没有起到积极的引导作用. 随着优化问题目标个数的增多, MaOEA-ABL 算法所设计的自适应分解策略的优势逐步表现出来, 经过自适应增强学习调整后的参考向量集可以更好地引导种群向真实的 Pareto 前沿靠近, 能够得到一组具有良好收敛性与多样性的解集.

MaF12 是目标函数具有不同规模的优化问题, 随着目标个数的增多, MaOEA-ABL 算法性能比其他两种算法性能好. 该算法对预定义的参考向量进行调整, 识别出参考向量超平面的有效面积, 提高积极向量的数目, 从而更好地引导种群进化. 并且设计了一种对 Pareto 形状无偏好的分解方法, 提高了种群的收敛性与多样性.

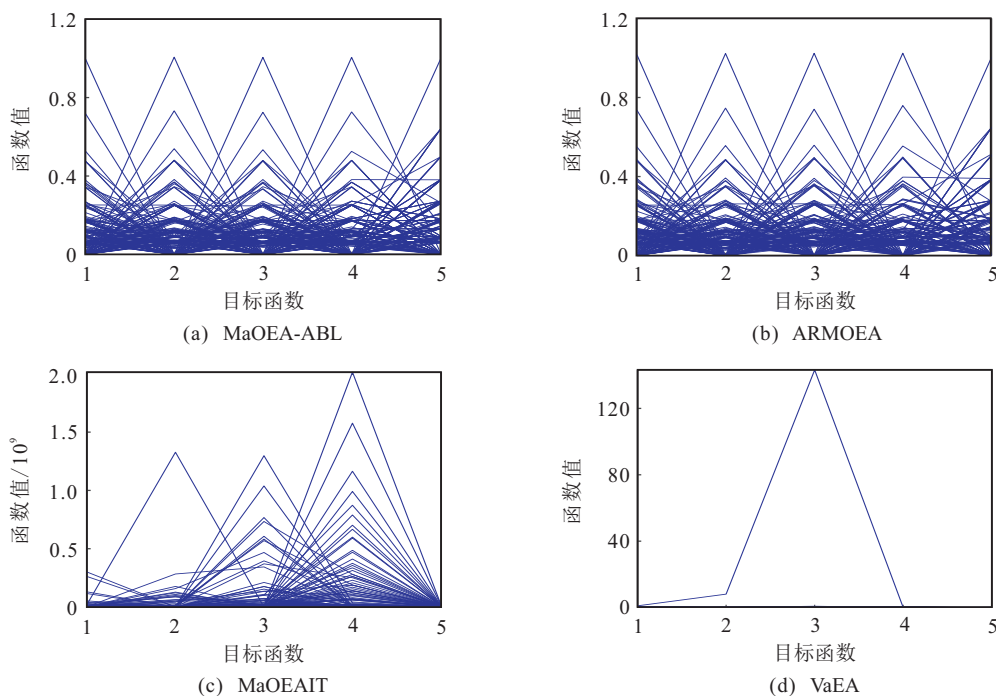


图 5 MaOEA-ABL、ARMOEA、MaOEAIT 和 VaEA 在目标为 5 的测试函数 MaF3 的解集

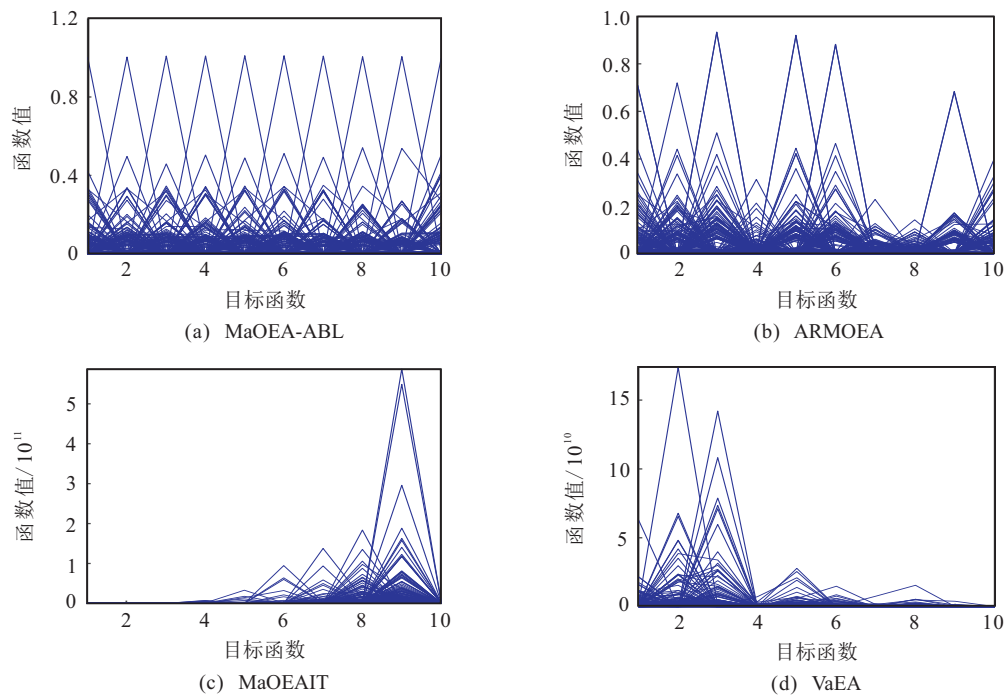


图6 MaOEA-ABL、ARMOEA、MaOEAIT和VaEA在目标为10的测试函数MaF3的解集

4 结论

本文在分析现有复杂超多目标优化问题面临的挑战后,提出了基于自适应增强学习的超多目标进化算法. 算法第1阶段在充分考虑参考向量活动性的基础上,利用自适应增强学习算法训练一个强分类器,将预定义生成的均匀分布参考向量分为积极参考向量与消极参考向量. 参考向量的活动可以反映真实Pareto前沿在目标空间中的分布,从而估计参考向量单位平面上的有效面积. 第2阶段在充分考虑真实前沿不规则的基础上,设计了基于CDM分解的个体更新策略. 将两种策略结合起来进行选解,基于自适应增强学习算法对参考向量进行调整,再选取CDM分解的方式进行优质解的选择,形成一种自适应分解的进化算法. 仿真结果表明,本文所提出的算法MaOEA-ABL在解决复杂超多目标优化问题时能够得到收敛性与多样性良好的Pareto解集. 目前存在的超多目标进化算法一般是在经典测试函数中进行仿真,但是,在实际工作中由于问题的复杂多变性,算法有待进一步改进. 因此,下一步的工作主要是对超多目标进化算法在实际工程中的应用展开研究.

参考文献(References)

[1] Hu Z Y, Wei Z H, Sun H, et al. Optimization of metal rolling control using soft computing approaches: A review[J]. Archives of Computational Methods in Engineering, 2021, 28(2): 405-421.
 [2] 田红军, 汪镭, 吴启迪. 一种求解多目标优化问题

的进化算法混合框架[J]. 控制与决策, 2017, 32(10): 1729-1738.

(Tian H J, Wang L, Wu Q D. A hybrid framework of evolutionary algorithm for solving multi-objective optimization problems[J]. Control and Decision, 2017, 32(10): 1729-1738.)

[3] 刘建昌, 赵阳杰, 李飞, 等. 基于R2指标的昂贵多目标进化算法[J]. 控制与决策, 2020, 35(4): 823-832.

(Liu J C, Zhao Y J, Li F, et al. Expensive multi-objective optimization algorithm based on R2 indicator[J]. Control and Decision, 2020, 35(4): 823-832.)

[4] Hu Z Y, Wei Z H, Ma X M, et al. Multi-parameter deep-perception and many-objective autonomous-control of rolling schedule on high speed cold tandem mill[J]. ISA Transactions, 2020, 102: 193-207.

[5] 顾清华, 莫明慧, 卢才武, 等. 求解约束高维多目标问题的分解约束支配NSGA-优化算法[J]. 控制与决策, 2020, 35(10): 2466-2474.

(Gu Q H, Mo M H, Lu C W, et al. Decomposition-based constrained dominance principle NSGA- for constrained many-objective optimization problems[J]. Control and Decision, 2020, 35(10): 2466-2474.)

[6] Cai X J, Hu Z M, Chen J J. A many-objective optimization recommendation algorithm based on knowledge mining[J]. Information Sciences, 2020, 537: 148-161.

[7] 赵玉亮, 宋业新, 康丽文. 基于向量角分解的高维多目标进化算法[J]. 控制与决策, 2021, 36(3): 761-768.

(Zhao Y L, Song Y X, Kang L W. Many-objective

- evolutionary algorithm based on vector angle decomposition[J]. *Control and Decision*, 2021, 36(3): 761-768.)
- [8] Wu M Y, Li K, Kwong S, et al. Evolutionary many-objective optimization based on adversarial decomposition[J]. *IEEE Transactions on Cybernetics*, 2020, 50(2): 753-764.
- [9] Asafuddoula M, Verma B, Zhang M J. A divide-and-conquer-based ensemble classifier learning by means of many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2018, 22(5): 762-777.
- [10] Zhao H T, Zhang C S, Zhang B. A decomposition-based many-objective ant colony optimization algorithm with adaptive reference points[J]. *Information Sciences*, 2020, 540: 435-448.
- [11] 刘建昌, 李飞, 王洪海, 等. 进化高维多目标优化算法研究综述[J]. *控制与决策*, 2018, 33(5): 879-887.
(Liu J C, Li F, Wang H H, et al. Survey on evolutionary many-objective optimization algorithms[J]. *Control and Decision*, 2018, 33(5): 879-887.)
- [12] Cai X Y, Mei Z W, Fan Z. A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors[J]. *IEEE Transactions on Cybernetics*, 2018, 48(8): 2335-2348.
- [13] Jiang S Y, Yang S X. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts[J]. *IEEE Transactions on Cybernetics*, 2016, 46(2): 421-437.
- [14] Asafuddoula M, Singh H K, Ray T. An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors[J]. *IEEE Transactions on Cybernetics*, 2018, 48(8): 2321-2334.
- [15] Jiang S Y, He X Y, Zhou Y R. Many-objective evolutionary algorithm based on adaptive weighted decomposition[J]. *Applied Soft Computing*, 2019, 84: 105731.
- [16] Elarbi M, Bechikh S, Coello Coello C A, et al. Approximating complex Pareto fronts with predefined normal-boundary intersection directions[J]. *IEEE Transactions on Evolutionary Computation*, 2020, 24(5): 809-823.
- [17] Zhang Q S, Zhu W, Liao B, et al. A modified PBI approach for multi-objective optimization with complex Pareto fronts[J]. *Swarm and Evolutionary Computation*, 2018, 40: 216-237.
- [18] Ge H W, Zhao M D, Sun L, et al. A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(4): 572-586.
- [19] 李学强, 刘海林. 一种新的求解复杂多目标优化问题的进化算法[J]. *系统仿真学报*, 2011, 23(9): 1860-1865.
(Li X Q, Liu H L. New evolutionary algorithm for complicated multiobjective optimization[J]. *Journal of System Simulation*, 2011, 23(9): 1860-1865.)
- [20] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting[J]. *Journal of Computer and System Sciences*, 1997, 55(1): 119-139.
- [21] Cheng R, Li M Q, Tian Y, et al. A benchmark test suite for evolutionary many-objective optimization[J]. *Complex & Intelligent Systems*, 2017, 3(1): 67-81.
- [22] Tian Y, Cheng R, Zhang X Y, et al. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility[J]. *IEEE Transactions on Evolutionary Computation*, 2018, 22(4): 609-622.
- [23] Liu Q, Jin Y, Heiderich M, et al. An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems[J]. *IEEE Transactions on Cybernetics*, 2020, 50(6): 1-14.
- [24] Liu Y P, Gong D W, Sun J, et al. A many-objective evolutionary algorithm using A one-by-one selection strategy[J]. *IEEE Transactions on Cybernetics*, 2017, 47(9): 2689-2702.
- [25] Wang R, Purshouse R C, Fleming P J. Preference-inspired coevolutionary algorithms for many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2013, 17(4): 474-494.

作者简介

呼子宇(1989—), 男, 讲师, 博士, 从事复杂系统建模、多目标进化算法、人工智能等研究, E-mail: hzy@ysu.edu.cn;

李玉林(1998—), 女, 硕士生, 从事多任务优化的研究, E-mail: ilmf103523@163.com;

魏之慧(1996—), 女, 硕士生, 从事多目标优化的研究, E-mail: weizhahui1036@163.com;

杨景明(1957—), 男, 教授, 博士生导师, 从事轧制过程建模、人工智能与神经网络应用、轧制过程自动控制等研究, E-mail: yangjm@ysu.edu.cn.

(责任编辑: 李君玲)