

全局替换的自适应权重调整MOEA/D

袁 田¹, 尹云飞^{1†}, 黄发良², 陈乙雄¹

(1. 重庆大学 计算机学院, 重庆 400044; 2. 南宁师范大学 广西人机交互与智能决策重点实验室, 广西 南宁 530100)

摘要: 当多目标问题的帕累托前沿形状较为复杂时, 基于分解的多目标进化算法MOEA/D的解的均匀性将受到很大的影响. MOEA/D利用相邻子问题的信息来优化, 但早期因为种群中的个体与子问题的关联是随机分配的, 仅在邻居间更新会浪费优秀解的信息, 影响收敛速度. 针对这些问题, 本文提出一种MOEA/D的改进算法(MOEA/D-GUAW). 该算法使用种群全局更新的策略, 来提高收敛速度; 使用自适应调整权重向量的策略来获得更均匀分布的解集. 将MOEA/D-GUAW算法与现有的MOEA/D, MOEA/D-AWA, RVEA和NSGA-III算法在10个广泛应用的测试问题上进行了实验比较. 实验结果表明, 提出的算法在大部分问题上, 反转世代距离评价指标IGD优于其他算法, 收敛速度也快于其他算法.

关键词: 多目标优化; 基于分解的进化多目标优化; 全局替换; 自适应权重调整

引用格式: 袁田, 尹云飞, 黄发良, 等. 全局替换的自适应权重调整MOEA/D. 控制理论与应用, 2023, 40(4): 653 – 662

DOI: 10.7641/CTA.2022.11231

Adaptive weight adjustment MOEA/D with global replacement

YUAN Tian¹, YIN Yun-fei^{1†}, HUANG Fa-liang², CHEN Yi-xiong¹

(1. College of Computer Science, Chongqing University, Chongqing 400044, China;

2. Guangxi Key Lab of Human-machine Interaction and Intelligent Decision, Nanning Normal University, Nanning Guangxi 530100, China)

Abstract: When the shape of pareto front is complex, the uniformity of solution in multi-objective evolutionary algorithm based on decomposition (MOEA/D) will be affected. The MOEA/D uses the information of neighboring subproblems to optimize, but in the early stage, because the association between individuals in the population and subproblems is randomly assigned, updating only among neighbors will waste the information of excellent solutions and affect the convergence speed. To address these issues, an improved algorithm for MOEA/D, MOEA/D-global uniform adaptive weight (MOEA/D-GUAW), is proposed. The algorithm uses the strategy of global population update to improve the convergence speed. And the adaptive weight vector adjustment strategy are used to obtain more uniformly distributed solutions. The MOEA/D-GUAW algorithm is compared with the existing MOEA/D, MOEA/D based on adaptive weight vector adjustment (MOEA/D-AWA), reference vector guided evolutionary algorithm (RVEA) and nondominated sorting genetic algorithm-III (NSGA-III) in 10 widely used test problems. Experimental results show that the inverted generational distance (IGD) metric of the proposed algorithm is better than other algorithms in most problems, and the convergence speed is faster than other algorithms.

Key words: multi-objective optimization; multi-objective evolutionary algorithm based on decomposition; global replacement; adaptive weight adjust

Citation: YUAN Tian, YIN Yunfei, HUANG Faliang, et al. Adaptive weight adjustment MOEA/D with global replacement. *Control Theory & Applications*, 2023, 40(4): 653 – 662

1 引言

最优化问题是科学研究中重要的研究问题之一. 相较于只有一个目标函数的单目标优化问题, 需要同时处理多个目标函数, 且目标函数之间存在冲突的多目标优化问题 (multi-objective optimization problem, MOP) 更符合实际情况. 由于多个目标间相互冲突, 所

以对于多目标优化问题, 并不能得到满足所有目标达到最优的解, 只能得到的是帕累托最优解, 即最佳权衡候选方案, 而后决策者可以根据需求从中选择最合适的方案.

大多数多目标优化问题会有许多甚至无限的帕累托最优解, 因此一次运行就可以获得多个帕累托最优

收稿日期: 2021–12–16; 录用日期: 2022–05–15.

†通信作者. E-mail: yinyunfei@cqu.edu.cn; Tel.: +86 13996491007.

本文责任编辑: 付俊.

国家自然科学基金项目(61962038), 广西八桂学者创新团队基金项目(201979)资助.

Supported by the National Natural Science Foundation of China (61962038) and the Guangxi Bagui Teams for Innovation and Research (201979).

解的进化多目标算法(multi-objective optimization evolutionary algorithms, MOEA)^[1]得到了广泛的关注. 相较于优化方法, 进化多目标算法还有可以处理不连续等复杂的帕累托最优面, 处理二进制以及整数型变量的优势^[2].

根据种群进化的选择压力来源, MOEA通常可以分为以下3类^[3]: 1) 基于支配关系的方法^[2,4], 以帕累托支配关系或者其他支配准则作为一个解的适应度, 保留非支配解; 2) 基于指标的方法^[5-6], 以一个性能指标, 常见的例如超体积, 来作为解的适应度, 倾向于选择对指标贡献更大的解; 3) 基于分解的方法^[7-8], 以子问题的聚合函数值作为解的适应度, 倾向于选择聚合函数值更小(对于最小化问题)的解.

不同于前两种方法, 基于分解的MOEA不是将多目标问题作为一个整体来处理, 而是将原多目标优化问题转化为一组单目标优化子问题, 通过寻求子问题的最优解最后组成原多目标优化问题的解集.

Zhang和Li^[8]在2007年提出了基于分解的多目标进化算法(multi-objective evolutionary algorithm based on decomposition, MOEA/D), 该算法用一组均匀分布的归一化权重向量表示一组单目标优化子问题, 每个子问题通过使用它的几个相邻子问题的信息来优化, 这样可以加快搜索速度, 在MOEA/D的进化过程中, 种群中不同的解与不同的子问题一一匹配进行优化, 然后通过维护这些子问题的多样性, 来保证整个种群的多样性.

基于MOEA/D作为一类新的多目标进化算法, 已成为近十年来的研究热点. 根据研究的方向不同, 现有的研究大致分为以下4个方面:

1) 子代繁衍机制的改进, 包括交叉、变异及选择算子的改进: 基于稳定匹配的MOEA/D(MOEA/D-stable matching, MOEA/D-STM)^[9]改进了个体与权重向量的匹配关系, 个体通过其与权重向量的角度对所有向量进行排序, 而权重向量通过其与所有个体的聚合函数值进行排序, 然后进行相互选择, 得到一个稳定的匹配, 以此来得到新的种群, 并且收敛性和多样性上也得到了很好的权衡. 基于排序和角度选择的MOEA/D(MOEA/D-sorting and angle based selection, MOEA/D-SAS)^[10]提出了一种针对MOP的带有排序和选择的MOEA/D新变种. 由基于分解的排序(decomposition-based sorting, DBS)和基于角度的选择(angle-based selection, ABS)这两个部分, 来实现收敛和多样性之间的平衡. 在自适应区域调整的MOEA/D(MOEA/D-adaptive region adjustment, MOEA/D-ARA)^[11]中, 提出使用权重向量将空间进行划分, 然后根据角度和距离来定义解与权重向量的适配度, 这样的适应度即考虑了收敛又考虑了多样性, 同时通过与等高点的比较判断新解是否更优来进行种群的更新.

2) 分解方法的研究: Ishibuchi等^[12]提出了一种根

据个体与邻居个体相同的数目是否超过用户定义的阈值, 来选择使用不同的分解方法, 当个体处于帕累托前沿的非凸区域时, 使用加权切比雪夫, 其他情况则使用加权和, 以及2010年提出的在不同时期根据种群的分布情况选择不同的权重分解方法, 并采用了两种方案来实现这个思想, 其一是使用多个权重向量网络, 对每个解都应用不同的标量化方法; 另一种是在单个权重向量网络中使用不同类型的标量函数, 其中不同的标量函数交替分配给每个权重向量^[13].

3) 权重向量的生成方式: 有相关的文献提出了各种各样的方法来获得用于标量化函数的单纯形中的权重的均匀分布子集^[14-15]. 在均匀分布(MOEA/D + uniform design, UMOEA/D)中, 作者提出了一种均匀设计方法, 种群规模再随着目标数量的增加而增大, 还能得到分布更加均匀的权重向量^[16].

4) 权重向量的调整: 文献[17]中提出了一种权重向量进行自适应调整的MOEA/D(MOEA/D based on adaptive weight vector adjustment, MOEA/D-AWA), 该算法针对切比雪夫分解法设计了一种权重向量的生成方式, 使用得到的权重向量进行优化, 直到接近帕累托前沿面时, 再根据种群的稀疏情况周期性地调整权重向量. 文献[18]以进化种群中解之间的角度作为相似度度量, 提出了多对多自适应权重更新的方法(MOEA/D-multiple to multiple, MOEA/D-M2M). 文献[19]中提出了一种利用均匀分布外部种群引导权重向量自适应调整的方法(MOEA/D-adapt weights, MOEA/D-Adaw).

MOEA/D有两大特点, 其一是将MOP分解为用一组均匀分布的归一化权重向量表示一组单目标子问题, 其二是利用相邻子问题的信息来优化. 但是预先设定好的权重向量不一定适应所有的MOP, 当帕累托前沿不规则时, MOEA/D的效果会受到影响. 另外, MOEA/D使用邻居信息来进行优化, 新产生的解仅在邻居间替换, 在进化前期, 由于解和权重向量的匹配关系是随机的, 解的信息不能得到充分利用, 因此会影响收敛速度, 浪费计算资源.

基于以上问题, 本文提出了一种种群全局更新策略, 自适应调整权重的MOEA/D改进算法. 本文的主要贡献如下:

1) 提出种群全局更新机制, 根据新解的信息计算更新概率, 对整个种群进行更新. 在保证种群多样性的前提下, 加快收敛速度.

2) 使用种群活跃度来对种群的收敛程度进行判断. 通过种群信息来决定调整权重的时间.

3) 在种群收敛到一定程度后, 计算解的最近邻距离作为解的稀疏度, 根据稀疏度自适应调整权重向量, 以解决拥有不规则帕累托前沿的MOP.

本文的其余部分组织如下: 第2节主要介绍了多目

标优化相关概念, MOEA/D的算法流程, 以及本文将要解决的问题; 第3节则详细介绍了本文提出的算法思想, 以及算法流程; 第4节是实验部分. 详细介绍了实验使用的基准测试函数和性能指标, 实验参数设置及实验结果展示与分析; 第5节是本文的结束语.

2 基础理论与问题提出

2.1 基础定义

多目标优化问题可以表述为

$$\min_{\mathbf{x} \in \Omega} F(\mathbf{x}) = (f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \cdots \ f_m(\mathbf{x}))^T, \quad (1)$$

其中: $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 是一个 n 维决策变量的向量, n 是决策变量的个数; Ω 是可行决策域; m 则是目标函数的个数.

帕累托占优^[1]: 设 $\mathbf{x}_A, \mathbf{x}_B$ 是式(1)的两个可行解, 当且仅当对于所有的 $i \in 1, \dots, m$ 都有 $f_i(\mathbf{x}_A) \leq f_i(\mathbf{x}_B)$, 且存在 $j \in 1, \dots, m$ 满足 $f_j(\mathbf{x}_A) < f_j(\mathbf{x}_B)$, 则称 \mathbf{x}_A 与 \mathbf{x}_B 相比是Pareto占优的, 并称为 \mathbf{x}_A 支配 \mathbf{x}_B .

帕累托最优解^[1]: 一个解 \mathbf{x}^* 被称为Pareto最优解, 当且仅当可行解中没有支配 \mathbf{x}^* 的解.

帕累托最优解集^[1]: 所有帕累托最优解的集合被称为帕累托最优解集.

帕累托前沿^[1]: 所有帕累托最优解所对应的目标向量组成的曲面被称为帕累托前沿(pareto front, PF).

2.2 基于分解的多目标优化算法

MOEA/D通过分解方法, 将一个多目标优化问题分解为若干个标量优化子问题, 并通过一组均匀分布的权重向量和聚合函数同时对所有子问题进行优化. 具体做法就是每个子问题的新个体除了会对自己进行替换更新^[8], 如果新个体在邻居子问题中的效果更好, 即聚合函数值更优, 也会替换邻居中的个体. MOEA/D算法的流程如表1所示, 算法流程图如图1所示.

在MOEA/D及其变体算法中常用的分解方法有: 切比雪夫(Tchebycheff)、基于惩罚的边界交叉法(penalty-based boundary intersection, PBI)以及加权和(weighted sum), 本文使用的是切比雪夫, 其数学描述如下所示:

2.3 问题分析

MOEA/D的基本假设是, 权重向量的均匀性自然会引导帕累托最优解的多样性. 为了保证子问题的多样性, MOEA/D采用预定义的均匀分布的权重向量集合与种群中的个体一一对应, 在整个进化过程中, 权重向量不会再改变.

$$\min_{\mathbf{x} \in \Omega} g^{tc}(\mathbf{x} | \boldsymbol{\lambda}, \mathbf{z}^*) = \max_{1 \leq i \leq m} (\lambda_i |f_i(\mathbf{x}) - z_i^*|), \quad (2)$$

其中 $\boldsymbol{\lambda}$ 和 \mathbf{z}^* 分别是权重向量和参考点.

当PF接近超平面 $\sum_{i=1}^m f_i(\mathbf{x}) = 1$ 时, 在目标空间中, MOEA/D可以得到均匀分布的Pareto最优解^[20], 这是

因为预设的权重向量是均匀分布在超平面 $\sum_{i=1}^m \lambda_i(\mathbf{x}) = 1$ 上的, 如果PF接近该超平面, 那么预设的均匀的权重向量就能使得种群得到均匀的解, 如图2. 但是当PF较为复杂时, 即PF不连续或具有尖峰低尾的形状, MOEA/D的基本假设会被违背^[17].

表 1 MOEA/D算法的流程

Table 1 The flow of MOEA/D algorithm

1	生成 N 个均匀分布权重向量, 根据权重向量间的欧几里得距离, 得到 T 个最近邻居权重向量, 以及对应的邻域 B^i ;
2	初始化种群 $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N$, N 为种群大小, 计算对应的 $FV^i = F(\mathbf{x}^i)$;
3	计算理想点, 记为 $\mathbf{z}^* = (z_1, z_2, \dots, z_m)$, 其中 m 为目标的维数, z_i 则是当前代的 f_i 的最小值;
4	for $i = 1, 2, \dots, N$, do:
5	根据概率 σ , 从 B_i 或整个种群中随机选择两个个体 \mathbf{k}, \mathbf{l} , 然后生成一个新个体 \mathbf{y} ;
6	更新理想点, 对于 $j = 1, 2, \dots, m$, 如果 $z_j < f_j(\mathbf{y})$, 则更新 $z_j = f_j(\mathbf{y})$;
7	更新邻域: 对于 $\mathbf{x}^j \in B^i$, 如果 \mathbf{y} 的聚合函数值小于 \mathbf{x}^j , 则 $\mathbf{x}^j = \mathbf{y}$, $FV^j = F(\mathbf{y})$;
8	若满足停止条件, 则停止迭代, 输出种群为最优解集; 否则继续执行4.

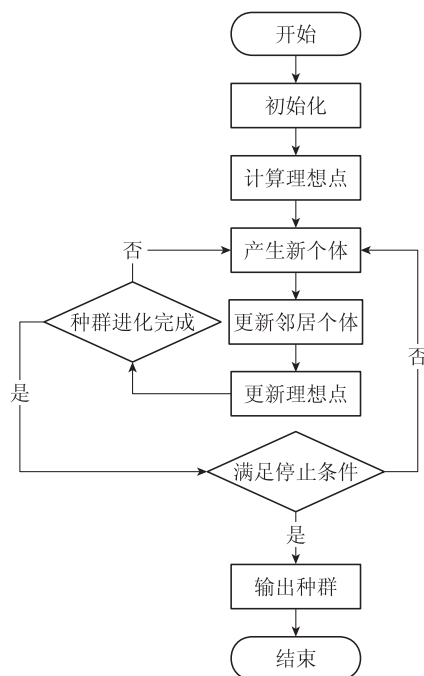


图 1 MOEA/D算法流程图

Fig. 1 Algorithm flow chart of MOEA/D

当PF不连续时, 会在断裂的端点产生多个拥有相同解的子问题. 由于子问题解的选择与权重向量以及聚合函数相关, 因此在断裂的端点处很可能会取得较优解. 如图3中 \mathbf{w}_4 对应的解只能在 \mathbf{w}_3 或 \mathbf{w}_5 的附近, 而这会影响种群的多样性, 无法得到一组均匀的解.

当PF如图4所示时,得到的解会聚集在 w_2 与 w_6 之间,而两端的解很稀疏.

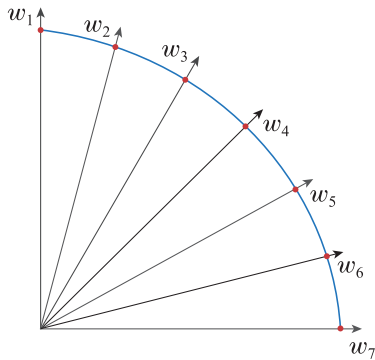


图2 PF接近超平面 $\sum_{i=1}^m f_i(\mathbf{x}) = 1$ 时权重与个体的分布
Fig. 2 Distribution of weights and individuals when PF near hyperplane $\sum_{i=1}^m f_i(\mathbf{x}) = 1$

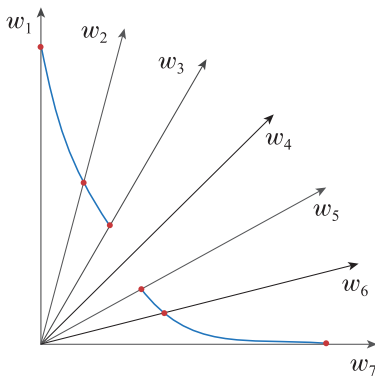


图3 不连续PF中权重与个体的分布
Fig. 3 Distribution of weights and individuals in discontinuous PF

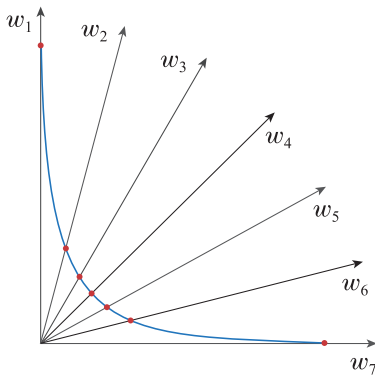


图4 凹状PF中权重与个体的分布
Fig. 4 Distribution of weights and individuals in concave PF

除此之外, MOEA/D使用的邻居信息来进行种群更新,能够在一定程度上加快收敛速度,但是因为前期的解与权重向量的对应关系是随机分配的,这就可能会使得可以保留的优秀的新解,因为聚合函数值差而被淘汰,解的信息没有被充分利用. 如图5, 向量 u 和向量 v 对应的解分别是 x_u 和 x_v , x' 是解 x_v 产生的新解, 对应向量 v 及其邻居而言, 会根据聚合函数值选择解 x_v , 而新解 x' 会被淘汰, 但是对于向量 u 而言, 新

解是优于解 x_u , 但因为 u 不是 v 的邻居而得不到新解 x' 的信息, 错失这个解.

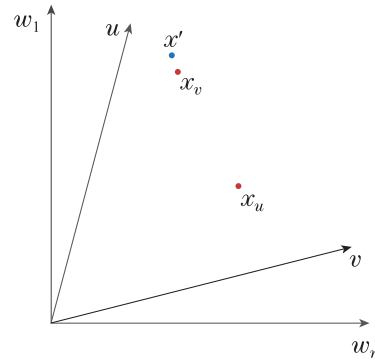


图5 早期权重向量与个体的分布
Fig. 5 Early weight vector and distribution of solutions

为了解决以上问题, 本文提出了一种全局替换, 自适应调整权重的MOEA/D改进算法(MOEA/D-global uniform adaptive weight, MOEA/D-GUAW), 来提高算法的计算效率和效果.

3 MOEA/D-GUAW算法

3.1 基本思想

本文的主要目标是针对不同PF对权重向量进行自适应调整, 提高算法的收敛速度, 为调整权重后提供更多的优化时间, 平衡好种群的收敛性和多样性, 得到分布更优的PF. 权重向量的调整主要依赖当前种群的分布情况, 这需要种群达到一定收敛程度, 能够体现出对应PF的走向. MOEA/D的收敛速度主要依赖与邻居间的信息共享, 具体来说, 就是用精英个体替换邻居内聚合函数值差的个体. 利用好精英个体则可以在保证多样性的同时加快种群收敛, 为后期调整权重和调整后进化提供更多的时间.

综上所述, 本文的基本思想是前期通过利用精英个体促进种群加快收敛, 根据种群活跃程度判断收敛状态, 达到一定收敛程度时, 根据种群中个体的拥挤程度, 对拥挤区域的子问题进行删除, 并在稀疏地区添加子问题.

3.2 全局替换策略

本文采用全局替换的策略, 即每个新产生的后代会与整个种群的权重匹配计算聚合函数值, 更差的个体被淘汰, 因为种群中的每个个体对应的权重都可以得到新个体的信息进行计算, 这样就可以避免出现第2.3节中图5的情况, 即有效的解因为和权重及其邻居的聚合函数值差而被淘汰. 但是, 如果不对新个体的替换进行限制, 可能会造成种群多样性的缺失, 因此本文使用可替换概率 Θ 来决定后代能否替换原解. Θ_j 越大, 对应的可替换的第 j 个个体就越可能被新解替换.

$$\Theta_j = \frac{1}{2} \left(\frac{r}{R} + \frac{d_j}{\max_{1 \leq i \leq R} d_i} \right), \quad (3)$$

其中: R 代表聚合函数值差于后代的解的个数, 即可被替换解的个数, r 代表还未被替换的个数, 这部分意味着被替换的解越多, 替换的可能性就越小, 保证种群的多样性. d_j 代表第 j 个可替换解与后代的聚合函数差值, 后面这部分则保证更差的原解被替换的概率更大.

由于进化后期, 个体与权重的关联关系基本固定, 全局替换的影响不大, 因此为了减少不必要的计算, 当种群进化到一定收敛程度后, 再重新使用邻居替换策略.

3.3 种群活跃度

不同于许多调整权重的MOEA/D改进算法, 需要人为设置调整权重的时机和频率, 本文使用种群活跃度来判断收敛状态, 根据种群状态进行权重调整. 本文使用种群两代个体目标值差异平均值作为种群的活跃度, 可以根据种群进化程度自适应决定调整权重的时机. 第 t 代种群活跃度 AC_t 的计算公式如下:

$$AC_t = \frac{\sum_{i=1}^N (g_{t-1}^{tc}(x^i) - g_t^{tc}(x^i))}{N}, \quad (4)$$

其中: N 为种群大小, $g(x)$ 为聚合函数值, t 代表第 t 代种群, x^i 代表第 i 个个体. AC 越大, 代表则种群越活跃. 当连续多代的种群活跃度都小于阈值时, 代表当前种群达到了一定的收敛程度. 设置连续多代的意义在于, 确定种群达到较为稳定的收敛状态.

3.4 自适应权重调整

本文使用邻近距离来评价当前解的稀疏程度, 取值为个体的前 m 个最近个体的欧式距离的平均值. 种群中第 j 个个体的稀疏程度定义如下:

$$S_j = \frac{\sum_{i=1}^m L^{NN_i^j}}{m}, \quad (5)$$

其中: $L^{NN_i^j}$ 为第 j 个解到第 i 个最近邻的欧氏距离^[21], m 为目标函数个数.

根据所有个体的稀疏度, 对稀疏度低于阈值的个体对应的子问题进行删除, 同时在稀疏度高的地方添加子问题. 需要注意的是, 每次仅调整一个子问题, 进行多次调整, 直至没有个体的稀疏度低于阈值或者达到最大调整次数. 如果一次调整多个子问题, 则可能会导致原本拥挤区域删除多个子问题变得过于稀疏.

添加权重向量则是在稀疏度最高的个体对应的子问题与其最近邻居对应子问题间添加, u, v 为两个子问题对应的权重向量, 子问题对应的权重向量计算公式如下:

$$\mathbf{W} = \left(\frac{u_1 + v_1}{2}, \frac{u_2 + v_2}{2}, \dots, \frac{u_m + v_m}{2} \right). \quad (6)$$

然后从外部种群中挑选一个与新权重的聚合函数值最小的解, 作为新的子问题的解. 外部种群保存的是被种群淘汰的非支配解, 从外部种群中选择新解, 主要是为了保证种群的多样性^[17], 同时也兼顾了收敛性.

3.5 MOEA/D-GUAW

本文算法的整体框架与MOEA/D相同, 主要差异在于后代的替换策略与基于稀疏度的自适应权重调整. 本文算法的流程如表2所示, 算法流程图如图6所示.

表2 本文算法MOEA/D-GUAW的流程

Table 2 The flow of MOEA/D-GUAW algorithm

1	生成 N 个权重向量, 得到对应邻域 B^i
2	初始化种群 P , 计算对应的 $FV^i = F(\mathbf{x}^i)$
3	计算理想点 \mathbf{z}^*
4	for $i = 1, 2, \dots, N$,
5	生成新个体 \mathbf{y} , 更新理想点
6	如果达到一定收敛, 执行7, 否则执行8
7	更新邻居
8	全局更新: 对于 $\mathbf{x}^j \in P$,
9	if $g^{tc}(\mathbf{y}, \lambda^j) < g^{tc}(\mathbf{x}^j, \lambda^j)$, 将 \mathbf{x}^j 添加到 P_r 中
10	for $\mathbf{x}^j \in P_r$, 计算可替换概率 Θ_j
11	if $\text{random}(\cdot) < \Theta_j$, 则 $\mathbf{x}^j = \mathbf{y}$, $FV^j = F(\mathbf{y})$, 以 \mathbf{x}^j 更新外部种群EP
12	计算 $AC_t, l = AC_t < \alpha?l + 1 : 0$
13	if $l \geq L$:
14	计算所有个体稀疏度 S , 调整次数 $t = 0$
15	while $\min(S) < \sigma$ and $t < \text{nus}$:
16	删除最小稀疏度的个体和对应权重
17	求最大稀疏度的个体 \mathbf{x}^i 的最近邻居 \mathbf{x}^j
18	for $\mathbf{x} \in \text{EP}$, 计算 $g^{tc}(\mathbf{x}, \lambda^k)$, 值最小的个体添加到 P 中, 并将其从EP删除
19	$t++ = 1$, 计算 S
20	若满足停止条件, 停止迭代输出种群; 否则执行4.

算法中有一组参数作为输入, 包括邻居子问题的个数 T , 从邻居中选择个体作为父代的概率, 最大迭代次数 G_{\max} , 种群最小活跃度 α , 最大连续不活跃代数 L , 最小稀疏度 σ , 最大权重调整个数 nus .

3.6 算法复杂度分析

在单次种群迭代进化过程中, 更新种群的时间复杂度为 $O(m \times N^2)$, 其中: m 为目标函数个数, N 为种群大小, 计算种群活跃度的时间复杂度是 $O(m \times N)$, 其中 m 为目标函数的个数, 调整权重的时间复杂度最大为 $O(\text{nus} \times m \times N^2)$, 其中 nus 是每次调整权重向量的最大调整个数. 因此单次迭代的时间复杂度为 $O(\text{nus} \times m \times N^2)$, 由于 m, nus 都远小于 N , 因此单次迭代的时间复杂度约为 $O(N^2)$.

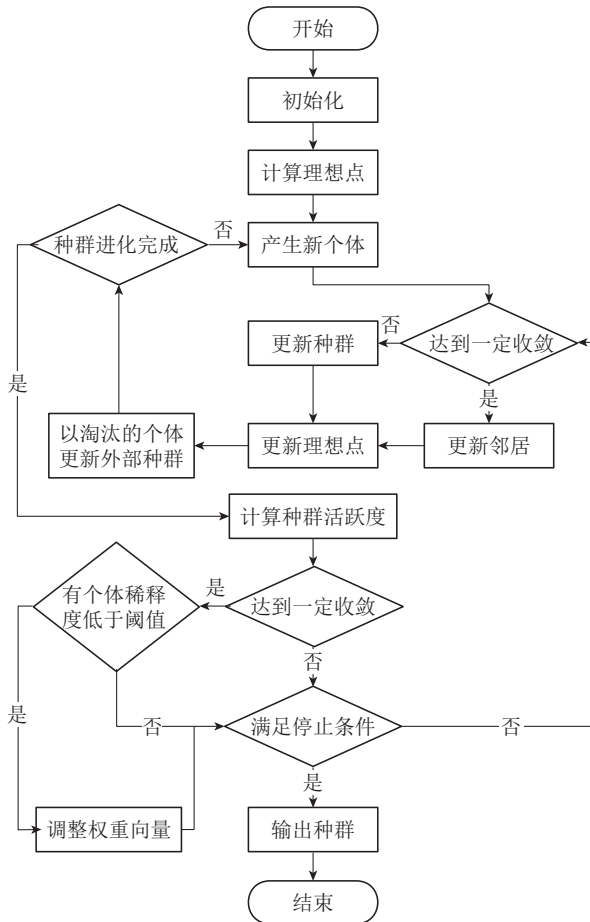


图 6 MOEA/D-GUAW算法流程图

Fig. 6 Algorithm flow chart of MOEA/D-GUAW

4 实验结果与分析

4.1 对比算法与测试函数

为了验证本文所提出算法的有效性,实验部分在二目标问题上选取了ZDT1-4, ZDT6, Kursawe, SymPart测试函数,三目标问题上选取了DTLZ1-3, DTLZ5和WFG1-3测试函数集.对比算法则选择了:MOEA/D, MOEA/D-AWA, REVA以及第3代非支配排序进化算法(nondominated sorting genetic algorithm-III, NSGA-III).

为了保证所有算法的可比较性,实验中所有相同的参数都设置了同样的值.参数包括交叉率 p_c 、变异率 p_m 、种群大小 N 、最大迭代次数 G ,MOEA/D及其改进算法的邻域大小 T ,其取值如表3,其中 n 为自变量维度.各算法中涉及的其他参数则均按照算法原文中的设置.

表 3 参数取值

Table 3 Parameter value

参数	p_c	p_m	N	G	T
二目标	0.9	$1/n$	150	150	15
三目标	0.9	$1/n$	190	200	19

本文提出的算法所需参数设置为最小活跃度 α 为

0.001,连续代数 L 为6(二目标)和8(三目标),最小稀疏度 σ 为0.005(二目标)和0.05(三目标),最大调整次数 mus 为10(二目标)与20(三目标).

4.2 性能指标

本文实验中用到了广泛用于评估多目标进化算法的指标,反转世代距离评价指标(inverted generational distance, IGD)以及超体积指标(hyper volume, HV).

IGD:使用一组真实PF*,然后计算算法获得的PF中的每个点到PF*之间的最近距离,得到所有点的最近距离平均值^[22].其计算公式如下所示:

$$IGD = \frac{\sum_{i=1}^{P^*} d_i}{P^*}, \quad (7)$$

其中: P^* 是PF*的大小, d_i 是第 i 个真实前沿点到算法所得PF的最近欧氏距离.IGD值越低,说明算法获得的近似PF的收敛性和多样性越好,越接近真实PF.

HV:算法获得的非支配解集与参照点围成的目标空间中区域的体积.HV值越大,说明算法的综合性能越好.其计算公式如下所示:

$$HV = \sigma(U_{i=1}^{P^*} v_i), \quad (8)$$

其中: σ 表示Lebesgue测度, P^* 表示算法得到的PF的大小, v_i 表示参照点与解集中第 i 个解构成的超体积.本文实验中用到的参考点值分别是二目标问题的参考点[2.0 2.0],三目标问题的参考点[2.0 2.0 2.0].

4.3 实验结果与分析

4.3.1 IGD性能指标结果分析

表4中给出了算法MOEA/D, MOEA/D-AWA, REVA和NSGA-III,及本文所提出的算法MOEA/D-GUAW分别独立运行30次后IGD的平均值以及方差.表4中,对各算法中显著性较优的数据进行了加黑操作.

从表4各算法的IGD性能指标的平均值可以看出:在二目标问题的7个测试函数中,有5个表现最好的算法都是MOEA/D-GUAW.这5个测试函数分别是ZDT1, ZDT2, ZDT4, Kursawe和SymPart. ZDT3上表现最好的算法是NSGA-III, MOEA/D-GUAW紧跟其后,与NSGA-III的差距也不大. ZDT6上表现最好的算法是MOEA/D-AWA, MOEA/D-GUAW算法的效果也不错,仅比MOEA/D-AWA差一点.

在三目标问题的7个测试函数中,MOEA/D-GUAW算法在DTLZ1, DTLZ3, DTLZ5, WFG1和WFG3这5个测试函数上都是表现最好的. NSGA-III算法在DTLZ2上略胜一筹, MOEA/D-GUAW虽然不是表现最好的,但不管是IGD平均值还是方差都优于MOEA/D和MOEA/D-AWA. WFG2上表现最好的是参考向量引导多目标优化进化算法(reference vector guided evolutionary algorithm, RVEA), MOEA/D-GUAW表现一般,但仍优于MOEA/D.

表4 IGD性能指标的平均值及方差统计表

Table 4 Average value and variance of IGD performance indicators

测试函数	算法				
	MOEA/D	MOEA/D-AWA	RVEA	NSGA-III	MOEA/D-GUAW
ZDT1	5.924e-3(4.742e-3)	5.057e-3(1.411e-3)	3.235e-2(3.609e-3)	4.677e-3(4.892e-4)	4.1e-3(1.207e-3)
ZDT2	5.916e-3(6.773e-3)	7.294e-3(9.787e-3)	4.993e-2(5.171e-3)	1.519e-2(2.482e-2)	3.112e-3(2.036e-3)
ZDT3	1.654e-2(2e-2)	1.271e-2(1.6e-2)	4.367e-2(6.36e-3)	5.029e-3(2.311e-4)	8.634e-3(7.765e-3)
ZDT4	1.153e-2(3.456e-3)	1.302e-2(5.762e-3)	2.49e-1(1.357e-1)	1.047e-1(6.983e-2)	9.86e-3(2.677e-3)
ZDT6	2.175e-3(8.006e-5)	2.137e-3(7.07e-5)	1.78e-1(2.214e-2)	6.767e-2(7.754e-3)	2.174e-3(1.645e-3)
Kursawe	3.49e-2(1.334e-3)	3.047e-2(1.75e-3)	8.351e-2(1.186e-2)	3.016e-2(1.688e-3)	2.754e-2(1.854e-3)
SymPart	1.836e-1(5.78e-3)	3.609e-2(2.469e-3)	4.096e-2(4.29e-3)	3.544e-2(6.02e-4)	3.539e-2(1.003e-3)
DTLZ1	2.135e-2(3.643e-4)	1.957e-2(7.764e-4)	2.952e-2(3.002e-2)	1.414e-2(5.692e-4)	1.393e-2(9.257e-4)
DTLZ2	5.0432e-2(3.413e-4)	4.556e-2(3.196e-4)	3.541e-2(6.834e-5)	3.533e-2(2.88e-5)	4.362e-2(1.047e-4)
DTLZ3	5.497e-2(4.581e-3)	5.933e-2(2.78e-2)	2.149e-1(2.705e-1)	5.287e-1(4.68e-1)	5.173e-2(5.747e-3)
DTLZ5	6.628e-3(1.376e-4)	5.099e-3(2.731e-4)	5.458e-2(4.6e-3)	3.092e-2(5.866e-3)	4.663e-3(5.19e-4)
WFG1	9.92e-1(3.183e-2)	9.818e-1(4.175e-2)	1.198(7.34e-2)	1.1(9.245e-2)	9.729e-1(5.159e-2)
WFG2	6.396e-1(1.577e-1)	4.903e-1(1.287e-1)	1.925e-1(7.752e-3)	2.612e-1(1.138e-1)	5.043e-1(1.116e-1)
WFG3	4.48e-2(3.603e-3)	3.249e-2(3.004e-3)	1.819e-1(1.534e-2)	9.793e-2(5.467e-3)	2.819e-2(6.228e-3)

表5 HV性能指标的平均值及方差统计表

Table 5 Average value and variance of HV performance indicators

测试函数	算法				
	MOEA/D	MOEA/D-AWA	RVEA	NSGA-III	MOEA/D-GUAW
ZDT1	3.645(2.336e-2)	3.649(8.457e-3)	3.587(9.518e-3)	3.654(1.739e-3)	3.655(9.009e-3)
ZDT2	3.306(3.631e-2)	3.298(6.969e-2)	3.175(2.029e-2)	3.227(1.667e-1)	3.324(1.344e-2)
ZDT3	4.751(1.235e-1)	4.777(9.097e-2)	4.645(2.428e-2)	4.802(1.823e-3)	4.804(7.184e-3)
ZDT4	3.626(1.274e-2)	3.621(2.228e-2)	3.084(2.519e-1)	3.379(1.403e-1)	3.632(1.526e-2)
ZDT6	3.038(2.221e-3)	3.038(2.588e-3)	2.513(6.491e-2)	2.826(2.412e-2)	3.042(1.846e-3)
Kursawe	260.652(3.073e-1)	260.689(2.115e-1)	260.193(2.352e-1)	260.945(1.536e-1)	260.949(2.148e-1)
SymPart	2.16e-3(7.564e-5)	1.737(8.106e-4)	1.719(7.701e-3)	1.738(6.103e-4)	1.738(5.287e-4)
DTLZ1	7.955(1.104e-2)	7.971(9.368e-4)	7.969(9.991e-3)	7.974(5.919e-4)	7.974(4.393e-4)
DTLZ2	7.398(2.29e-3)	7.417(1.091e-3)	7.431(5.307e-4)	7.432(2.049e-4)	7.411(3.532e-3)
DTLZ3	7.379(1.745e-2)	7.376(5.91e-2)	6.714(1.45)	5.515(2.463)	7.384(1.983e-2)
DTLZ5	6.097(3.424e-4)	6.1(5.618e-4)	5.971(2.22e-2)	6.053(2.723e-3)	6.197(1.701e-2)
WFG1	8.463e-1(6.8e-2)	8.817e-1(1.026e-1)	5.78e-1(7.575e-2)	8.447e-1(1.183e-1)	8.976e-1(1.327e-1)
WFG2	2.892(1.168)	3.351(1.285)	5.134(4.062e-2)	4.729(1.11)	3.337(1.266e-1)
WFG3	9.11e-1(5.717e-3)	9.143e-1(6.394e-3)	7.856e-1(2.14e-2)	8.342e-1(7.703e-3)	9.151e-1(5.149e-3)

4.3.2 HV性能指标结果分析

表5给出了各算法独立运行30次后HV指标的平均值以及方差,并对各算法中较优的数据进行了加黑操作。

从表5各算法的HV性能指标的平均值可以看出:在二目标问题的7个测试问题中,MOEA/D-GUAW算法的表现都是最好的,其中在SymPart上,HV的平均值上NSGA-III与MOEA/D-GUAW的表现并列第一,但是MOEA/D-GUAW在方差上更优一些。

在三目标问题的7个测试函数当中,MOEA/D-

GUAW算法在DTLZ1、DTLZ2、DTLZ5、WFG1以及WFG3这5个测试函数上表现都是最好的。在其他的两个测试函数中,NSGA-III与MOEA/D-GUAW在DTLZ2上的表现并列第一,但是MOEA/D-GUAW在方差上更优一些。DTLZ2表现最好的NSGA-III,WFG1上表现最好的RVEA算的,MOEA/D-GUAW虽然略差,但也优于MOEA/D。

4.3.3 收敛速度结果与分析

图7给出了MOEA/D,MOEA/D-AWA和MOEA/D-GUAW这3个基于分解类型的算法,在二目标问题上

单次运行时,迭代过程中的IGD值变化.从图中,可以看出,除了Kursawe, MOEA/D-GUAW在其他的5个测试函数上收敛速度都有所提升,在ZDT1, ZDT2以及ZDT4上的提升较为明显.

图8则是这三个算法在三目标问题上单次运行时,迭代过程中的IGD值变化.可以看出在DTLZ2, DTLZ5, WFG1和WFG3上, MOEA/D-GUAW的收敛速度较MOEA/D和MOEA/D-AWA也有所提升.

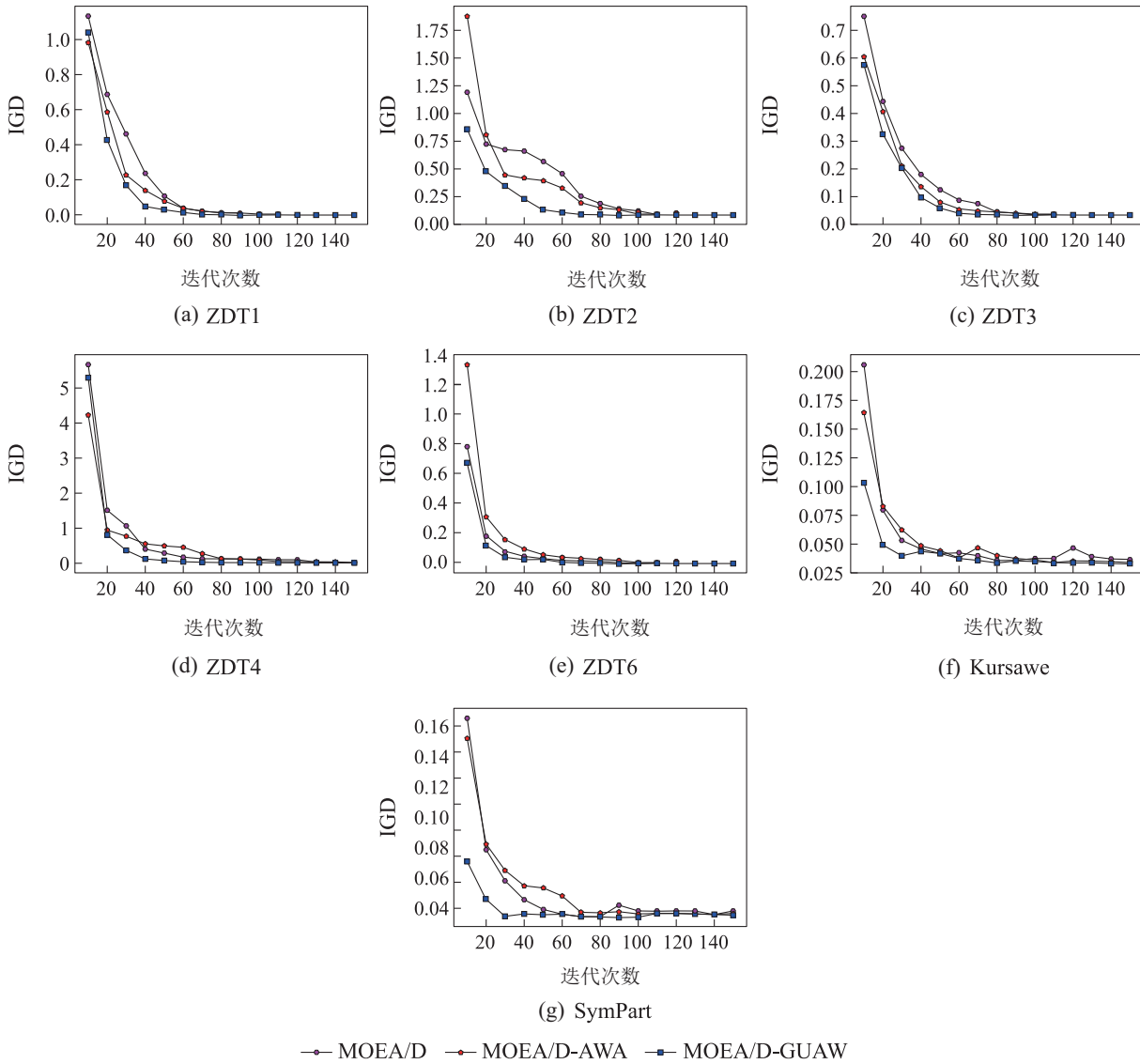


图7 各算法在二目标问题上迭代时IGD值的变化

Fig. 7 The change of IGD value of each algorithm when iterating on the two-objective problem

为了进一步验证全局替换策略在促进种群加快收敛的效果,以本文使用到的种群活跃度及收敛状态来对3个算法进行判断,得到各算法在测试问题集上达到一定收敛状态的迭代次数的平均值以及各算法达到收敛的迭代次数的平均值,二目标问题结果如图9所示,3目标结果如图10.图中深色部分代表达到一定收敛的迭代次数,浅色部分代表收敛的迭代次数.

从图9可以看出,在7个二目标问题测试函数中, MOEA/D-GUAW算法都是最快达到一定收敛状态和收敛的.在ZDT3, ZDT4, ZDT6和SymPart上达到一定收敛状态的提升幅度较大.在ZDT1, ZDT2, ZDT3,

Kursawe和SymPart上达到收敛的时间都提升比较多.

从图10可以看出,除WFG3和DTLZ5外, MOEA/D-GUAW算法都是最快达到一定收敛状态的.在所有问题中, MOEA/D-GUAW算法都是最快达到收敛状态的,在WFG1, WFG2和WFG3上的提升效果较大.可见全局替换策略在促进种群收敛上起到了一定的作用.无论是二目标还是三目标的优化问题, MOEA/D-GUAW都能加快一定的收敛速度.

综上,本文提出的MOEA/D-GUAW算法能够较好的解决不同的多目标优化问题,既保证了收敛,也很好的平衡了多样性.

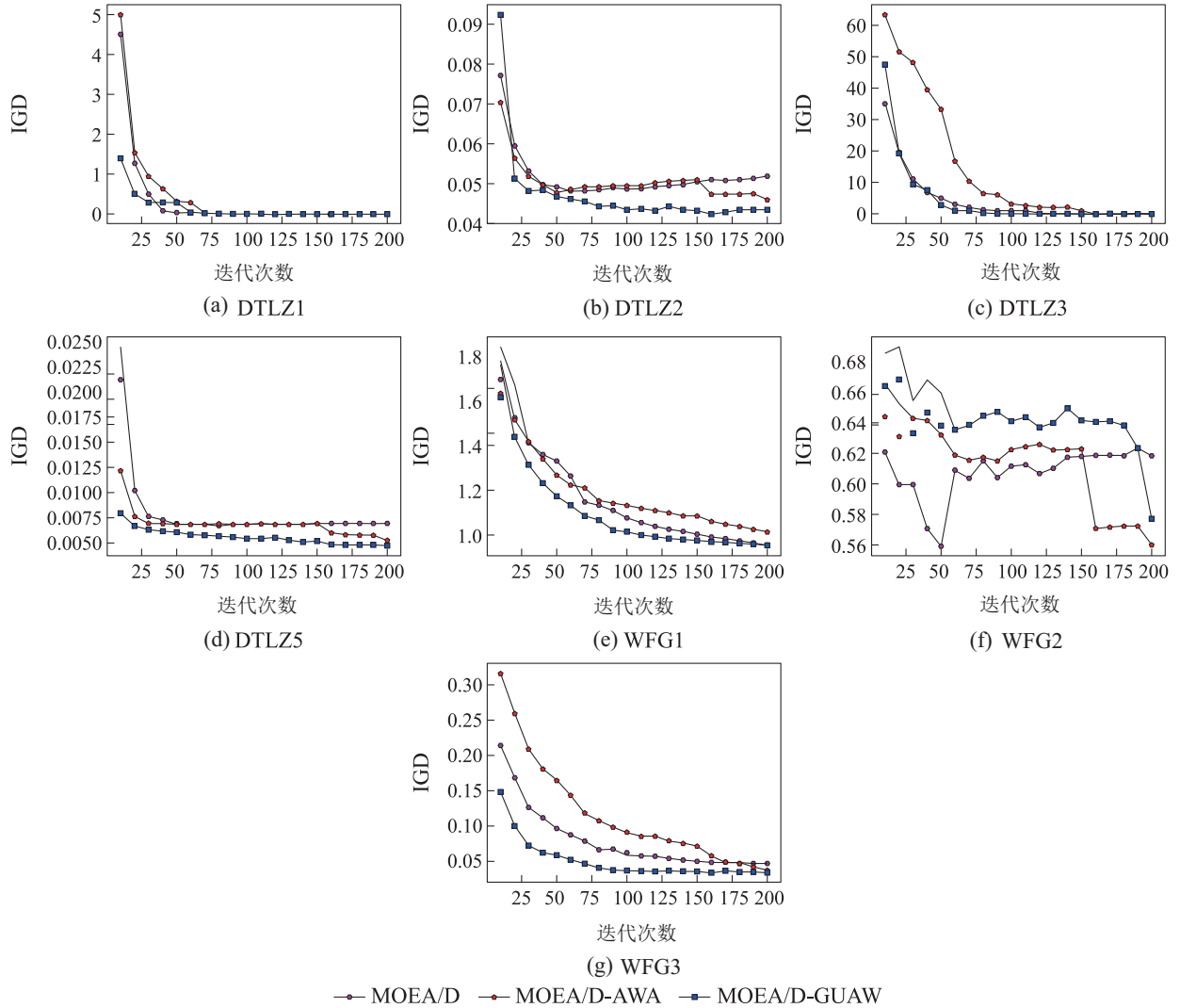


图 8 各算法在三目标问题上迭代时IGD值的变化

Fig. 8 The change of IGD value of each algorithm when iterating on the three-objective problem

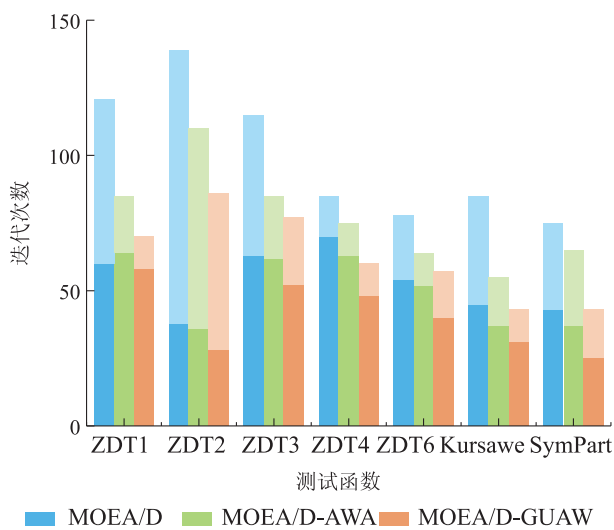


图 9 各算法在二目标问题上达到一定收敛和收敛状态的迭代次数

Fig. 9 The numbers of iterations for each algorithm to reach a certain convergence state and convergence state on the 2-objective problems

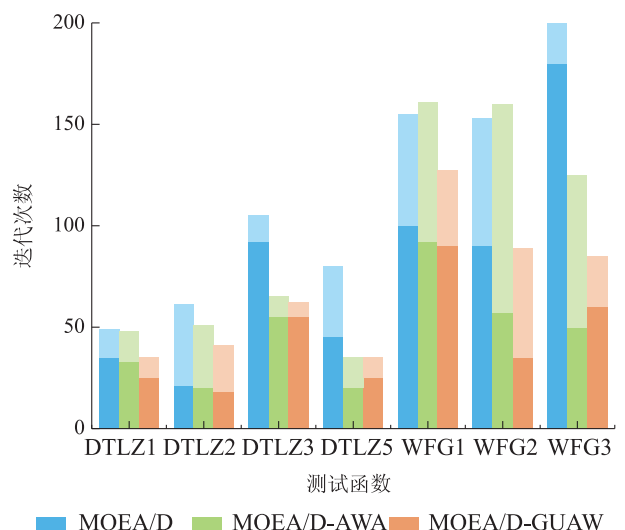


图 10 各算法在三目标问题上达到一定收敛和收敛状态的迭代次数

Fig. 10 The numbers of iterations for each algorithm to reach a certain convergence state and convergence state on the 3-objective problems

5 结论

为了解决MOEA/D算法不能很好解决拥有不同形状PF的MOPs以及前期邻域更新策略对解信息利用不充分的问题, 本文提出充分挖掘解信息, 合理运用精英个体进行全局替换的策略. 同时利用种群活跃度判断种群的状态, 在种群达到一定收敛程度后, 利用种群的分布信息, 自适应的调整权重向量, 最终形成了一种全局更新的自适应权重调整MOEA/D算法. 实验结果表明, 本文所提出的算法具有更优的分布性和收敛性, 在二目标问题上的提升比较明显, 在三目标问题上也有一定的提升.

参考文献:

- [1] GASPAR-CUNHA A, COVAS J A. Multi-objective optimization using evolutionary algorithms. *Computational Optimization and Applications*, 2008, 39(1): 4 – 19.
- [2] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182 – 197.
- [3] DONG Z, WANG X, TANG L. MOEA/D with a self-adaptive weight vector adjustment strategy based on chain segmentation. *Information Sciences*, 2020, 521: 209 – 230.
- [4] JAIN H, DEB K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 602 – 622.
- [5] BBADER J, ZITZLER E. HypE: An algorithm for fast hypervolume-based many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2011, 19(1): 45 – 76.
- [6] PAMULAPATI T, MALLIPEDDI R, SUGANTHAN P N. I_{SDE}^+ - An indicator for multi and many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2018, 23(2): 346 – 352.
- [7] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577 – 601.
- [8] ZHANG Q, LI H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2008, 11(6): 712 – 731.
- [9] KE L, ZHANG Q, KWONG S, et al. Stable matching-based selection in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2014, 18(6): 909 – 923.
- [10] CAI X, YANG Z, FAN Z, et al. Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization. *IEEE Transactions on Cybernetics*, 2016, 47(9): 2824 – 2837.
- [11] PENG W, BO L, WEN Z, et al. Adaptive region adjustment to improve the balance of convergence and diversity in MOEA/D. *Applied Soft Computing*, 2018, 70: 797 – 813.
- [12] ISHIBUCHI H, SAKANE Y, TSUKAMOTO N, et al. Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm. *The 5th International Conference on Evolutionary Multi-Criterion Optimization*. Univ Nantes, Fac Sci, Nantes, France: Springer-Verlag, 2009: 438 – 452.
- [13] ISHIBUCHI H, SAKANE Y, TSUKAMOTO N, et al. Simultaneous use of different scalarizing functions in MOEA/D. *The 12th Annual Genetic and Evolutionary Computation Conference*. Portland, OR, USA: IEEE, 2010: 519 – 526.
- [14] SCHEFFÉ H. The simplex-centroid designs for experiments with mixtures. *Journal of Royal Statistical Society Series B: Methodological*, 1963, 25(2): 235 – 251.
- [15] ZHANG Q, LEUNG Y W. An orthogonal genetic algorithm for multimedia multicast routing. *IEEE Transactions on Evolutionary Computation*, 1999, 3(1): 53 – 62.
- [16] TAN Y Y, JIAO Y C, HONG L, 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.
- [17] QI Y, MA X, FANG L, et al. MOEA/D with adaptive weight adjustment. *Evolutionary Computation*, 2014, 22(2): 231 – 264.
- [18] LIU H, GU F, ZHANG Q. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 2014, 18(3): 450 – 455.
- [19] LI M Q, YAO X. What weights work for you? Adapting weights for any pareto front shape in decomposition-based evolutionary multiobjective optimisation. *Evolutionary Computation*, 2020, 28(2): 227 – 253.
- [20] LIU H L, GU F Q, CHEUNG Y M. T-MOEA/D: MOEA/D with objective transform in multi-objective problems. *International Conference of Information Science and Management Engineering*. Xi'an, China: IEEE, 2010, 2: 282 – 285.
- [21] KUKKONEN S, DEB K. A fast and effective method for pruning of non-dominated solutions in many-objective problems. *Lecture Notes in Computer Science*, 2006, 4193: 553 – 562.
- [22] BOSMAN, AN P, THIERENS, et al. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 174 – 188.

作者简介:

袁田 硕士研究生, 目前研究方向为进化多目标优化, E-mail: yauntian7@qq.com;

尹云飞 副教授, 研究生导师, 目前研究方向为数据挖掘、物联网与嵌入式开发、自动控制与工业自动化, E-mail: yinyunfei@cqu.edu.cn;

黄发良 教授, 研究生导师, 目前研究方向为数据挖掘和人工智能, E-mail: faliang.huang@gmail.com;

陈乙雄 研究生导师, 目前研究方向为最优化理论及其应用、语义网、移动计算、云计算, E-mail: chenyx@cqu.edu.cn.