

文章编号: 1006-3080(2025)04-0522-16

DOI: 10.14135/j.cnki.1006-3080.20241007001

# 基于网格拥挤度的自适应参考点多目标优化算法

王学武, 高永亮, 顾幸生

(华东理工大学能源化工过程智能制造教育部重点实验室, 上海 200237)

**摘要:**在多目标优化中, 对于搜索到的种群要兼顾收敛性和分布性。基于指标的参考点自适应多目标优化算法(AR-MOEA)算法强调IGD-NS指标的最优, 算法收敛过程加快, 容易陷入局部最优, 导致种群不能覆盖到完整的Pareto前沿。本文提出了一种基于网格拥挤度的自适应参考点多目标优化算法(AR-MOEA-GC), 该算法区分了种群中贡献个体与非贡献个体的适应度计算方法, 保证种群的分布性和收敛性; 同时, 为了加快种群在算法后期的收敛速度, 融入了参考点调整策略, 辅助种群向真实Pareto进化。将改进的算法与6个先进的多目标进化算法在3类测试函数上测试, 结果表明AR-MOEA-GC在三维的多目标优化问题上有着一定的竞争力。

**关键词:**进化算法; IGD-NS 指标; 多目标优化; 网格拥挤度; 进化计算

**中图分类号:** TP301

**文献标志码:** A

在化工生产<sup>[1]</sup>、数据挖掘<sup>[2]</sup>、生产调度<sup>[3]</sup>、AGV 机器人轨迹规划<sup>[4]</sup>中, 经常出现同时处理优化多个目标的多目标优化问题(Multi-Objective Problem, MOP)。与单目标优化问题相比较, 多目标优化问题不存在唯一的最优解, 而是一组折衷解, 这组折衷解在多个目标中均表现出色, 并不只关注一个目标而忽略其他目标。

在众多解决多目标优化问题的算法中, 常用的是基于进化的多目标进化优化算法(Multi-Objective Evolutionary Algorithm, MOEA)。Deb等<sup>[5-6]</sup>提出的非支配排序算法NSGA-II和NSGA-III, 是在多目标优化问题上应用广泛的两种算法, 在NSGA-II中临界层个体根据聚集距离进行选择, 而NSGA-III中临界层个体的选择需要参考点进行辅助以保证选择到个体在目标空间中的分布性。Zitzler等<sup>[7]</sup>将评价指标融入MOEA, 通过计算种群中个体独立支配的空间来计算个体的适应度, 从而对个体进行选择。

在国内, 对于多目标优化进化算法的研究也是一个热门的领域。Zhang等<sup>[8]</sup>提出了基于分解的多

目标优化算法(MOEA/D), 通过生成一系列均匀分布的参考点将多目标优化问题分解成若干个单目标优化子问题, 利用各个子问题之间的信息来优化多目标优化问题, 取得了较为不错的效果。李飞等<sup>[9]</sup>在基于分解的多目标优化算法中融合了差分进化的思想, 提出了dMOPSO-DE。周艳丽等<sup>[10]</sup>在多目标差分进化算法中使用精英镜像初始化策略, 引入参数自适应和多变异策略, 提升多目标差分进化(MODE)算法的性能。王学武等<sup>[11]</sup>等提出了一种基于统计信息反馈的分步多目标优化算法, 将算法分为单目标探索阶段、单目标到多目标的过渡阶段、群体划分局部优化阶段3个阶段, 根据每个阶段的性质设计任务和策略, 以增强算法的收敛性和分布性。杨雨婷等<sup>[12]</sup>使用参考向量再生的方式丰富解集以充分遍历不规则Pareto前沿面, 提出Dynamic-RVEA算法。Jain等<sup>[13]</sup>提出NSGA-III的改进算法A-NSGA-III, 该算法在原算法的基础上设计了自适应参考点策略, 提升了算法在不规则Pareto前沿问题上的性能。Chen等<sup>[14]</sup>提出了一种基于自适应切换的优化算法, 在该算法中

收稿日期: 2024-10-07

基金项目: 国家自然科学基金(62076095)

作者简介: 王学武(1972—), 男, 陕西合阳人, 博士, 副教授, 研究方向包括智能优化技术、工业机器人优化与调度、焊接专家系统、电弧增材制造、焊接机器人智能化技术。E-mail: wangxuewu@ecust.edu.cn

引用本文: 王学武, 高永亮, 顾幸生. 基于网格拥挤度的自适应参考点多目标优化算法[J]. 华东理工大学学报(自然科学版), 2025, 51(4): 522-537.

Citation: WANG Xuewu, GAO Yongliang, GU Xingsheng. An Adaptive Multi-Objective Optimization Algorithm with Reference Point Based on Grid Congestion Degree[J]. Journal of East China University of Science and Technology, 2025, 51(4): 522-537.

设计了一种基于学习机制的新型双种群协作搜索策略,增强开发能力并有效利用种群中的优化知识。王学武等<sup>[15]</sup>提出了一种基于超体积的多目标优化算法,该算法通过提前删除非支配个体再嵌入 NSGA-III 算法,大大降低了算法的复杂度,提升了算法的运行效率。耿焕同等<sup>[16]</sup>提出 AR-NSGA-III 算法,该算法设计了参考点自适应策略,根据种群的熵差删除一部分的参考点,提高了种群在不规则 Pareto 前沿上的分布性与收敛性。Tian等<sup>[17]</sup>提出了一种基于增强世代距离指标的算法 AR-MOEA,其中的自适应方法是根据外部存档中候选解的指标贡献来调整参考点集,从而更快地收敛于 Pareto 前沿。Jiang 等<sup>[18]</sup>提出了一种简单快速的、基于超体积指标的多目标优化算法(MOEA-FV),通过删除不相关的解来降低算法的运行时间。付世炜等<sup>[19]</sup>提出了 MOEA/IGD-NSE,该算法以 IGD-NSE 指标为基础,在 DTLZ 和 MaF 系列测试问题上有着较为不错的表现。叶灵芝等<sup>[20]</sup>使用 NGGA-II 与理想点法结合,并使用一种新的相似度度量方法以优化火电机组运行的最优参数。江灏等<sup>[21]</sup>在 FBG 传感网络中利用 NSGA-II 算法求解 Pareto 曲线,为网络中的每个 FBG 传感器合理地分配 Bragg 波长的工作范围,以节约光源带宽资源。张英贵等<sup>[22]</sup>以 NSGA-II 算法为基础,结合基于移动不可行解的自适应惩罚函数的约束处理技术,提出了一种基于多目标进化的铁路超限超重货物运输专列路径优化算法,克服了传统专列路径优化中单纯依靠经验或从备选方案中择优的弊端,可为铁路超限超重货物运输专列路径决策提供技术支持。

为进一步提高种群的收敛性与分布性,本文在 AR-MOEA 算法基础上进行改进,提出了基于网格拥挤度的自适应初始参考点集多目标优化算法(AR-MOEA-GC)。该算法针对三维多目标优化问题,首先,在 AR-MOEA 算法基础之上添加了网格拥挤度的思想,区分非贡献个体与贡献个体,以保证种群中个体的分布性,同时也可以降低算法陷入局部最优的几率;其次,根据种群的熵差将搜索过程分为探索阶段和收敛阶段,在收敛阶段删除一部分的参考点使搜索到的种群可以更完整地覆盖到真实 PF。

## 1 基本概念

MOP 是一类高度非线性的问题,有着较大的求解难度。对于一个连续的多目标优化问题,其数学描述<sup>[23]</sup>如下所示:

$$\begin{cases} \min \mathbf{F}(X) = (f_1(x), \dots, f_m(x))^T \in R^m \\ \text{s.t. } (x_1, \dots, x_m)^T \in \Omega \end{cases} \quad (1)$$

其中,  $R^m$  为目标空间,  $\Omega \in R^n$  为决策空间,  $\mathbf{F}(x): \Omega \rightarrow R^m$  是具有  $m$  个目标函数值的向量,  $\Omega \in R^n$  代表一个连通集, s.t. 代表该多目标优化问题的约束条件。多目标进化算法中的重要基本概念如下:

(1) 支配关系: 存在  $x_1$  与  $x_2$ , 当对所有的  $i = 1, 2, \dots, m$  都有  $f_i(x_1) \leq f_i(x_2)$ , 且至少存在一个  $j = 1, 2, \dots, m$ , 使  $f_j(x_1) < f_j(x_2)$ , 就称支配  $x_2$ , 记作:  $x_1 \leq x_2$ ;

(2) 非支配解:  $x^*$  是一个可行解, 当满足  $\nexists x \in \Omega: x \leq x^*$ ,  $x^*$  就被称为非支配解;

(3) Pareto 最优解集: 在决策空间中, 所有非支配解组成的集合为 Pareto 最优解集 (Pareto Set, PS);

(4) Pareto 前沿: 决策空间中的非支配解在目标空间的映射为 Pareto 前沿 (Pareto Front, PF)。

## 2 基于网格拥挤度的自适应多目标优化算法

### 2.1 AR-MOEA

基于指标的参考点自适应多目标优化算法(AR-MOEA)融合了基于支配的 MOEA 和基于指标的 MOEA 算法的特点, 首先对种群中的个体进行非支配排序, 临界层之前的个体全部选择进入下一代, 在临界层使用 IGD-NS 指标对个体进行选择, 使种群在加速收敛的同时也能够保持一定的分布性。此外, 为了增加算法的通用性, 让其能够适应不同形状的 Pareto 前沿, 算法还加入了自适应参考点机制, 根据算法初始化的参考点和外部存档中的候选解, 对 IGD-NS 指标的贡献自适应调整参考点。

AR-MOEA 中重要的概念是 IGD-NS 指标<sup>[17]</sup>, 即增强反向世代距离。要计算 IGD-NS 指标, 需要区分种群中的贡献个体与非贡献个体( $x'$ ), 计算如下:

$$\nexists y \in Y \text{ satisfying } \text{dis}(y, x') = \min_{x \in X} \text{dis}(y, x) \quad (2)$$

其中,  $Y$  是参考点,  $\text{dis}(y, x)$  表示目标空间中  $y$  和  $x$  之间的欧氏距离。依据非贡献个体的定义, IGD-NS 指标被定义为:

$$\text{IGD-NS}(X, Y) = \sum_{y \in Y} \min_{x \in X} \text{dis}(y, x) + \sum_{x' \in X'} \min_{y \in Y} \text{dis}(y, x') \quad (3)$$

其中,  $X$  为非支配解集。实际上, IGD-NS 计算的是非贡献个体与最近参考点的欧式距离以及每个参考点与最近的非支配解的欧式距离之和。在临界层使用

IGD-NS 指标选择个体,会导致算法比较容易陷入局部最优。分析其原因是在计算个体适应度时过于追求 IGD-NS 指标的最优,导致种群进化前期忽略某些关键个体,从而放弃关键个体所在区域的探索,使种

群在搜索前期就丧失良好分布性,导致算法在后期只会在某些区域进行搜索,算法陷入了局部最优,结果如图 1 所示。

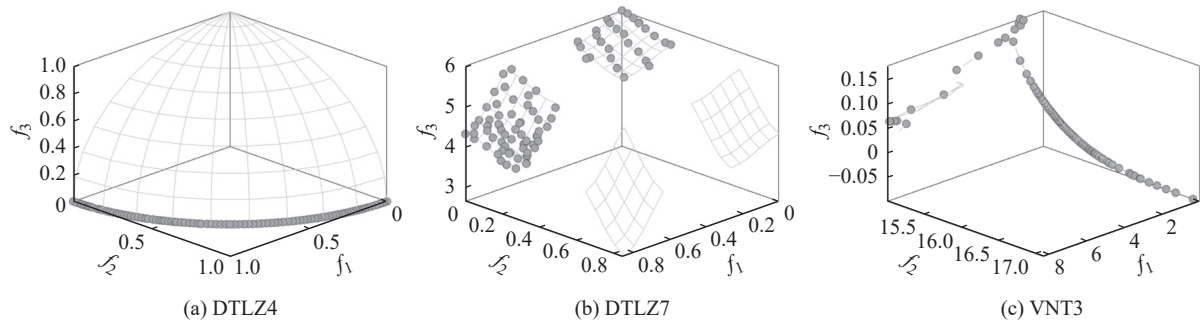


图 1 AR-MOEA 陷入局部最优示意图

Fig. 1 AR-MOEA local optimum trapping illustration

## 2.2 AR-MOEA-GC

前文提到 AR-MOEA 有时会陷入局部最优,从而无法搜索到整个 Pareto 前沿,导致在一些问题上算法的性能不佳。针对这个问题,在 AR-MOEA 的基础上加入了网格拥挤度(Grid congestion degree)的思想,该算法将整个目标空间划分为若干个网格。此外,在算法中加入参考点调整策略,减少种群在算法后期进行的无效搜索,尽可能地让种群接近真实的 Pareto 前沿。

**2.2.1 自适应参考点集方法** AR-MOEA-GC 中参考点调整方法是受到 AR-MOEA 和 AR-NSGA-III 中参考点调整的方法启发。该算法将整个搜索过程分为两个阶段:探索阶段与收敛阶段。在探索阶段,参考点集的大小是 1.1 倍种群的大小,引入参考点的作用是为了在环境选择中辅助选择非支配个体,让算法在前期进行足够的探索,保证种群分布性,避免算法过早收敛。同时,因为参考点的个数多于种群个体的数目,每一轮进化都会有参考点没有关联到个体,等算法进入收敛阶段,累积关联个体数目少的参考点与原点的连线不经过真实的 Pareto 前沿。如果删除参考点集  $R$  中关联个体数量后 10% 的参考点,减少无效计算,从而加快种群向真实 Pareto 前沿收敛的速度,再利用剩余的参考点辅助种群进化,这可以在保证种群分布性的同时加快种群的收敛速度。

对于如何判断是否进入收敛阶段,采用耿焕同等<sup>[16]</sup>提出的方法,随着种群的不断进化,会慢慢朝某个方向收敛,其熵的变化幅度也会越来越小,所以可以通过计算每代种群之间的熵差来判断种群的收敛程度,并与阈值  $u$  进行比较,来决定种群何时进入收敛阶段。

**2.2.2 个体适应度计算方法** AR-MOEA 算法为了追求 IGD-NS 指标的最优,会删除某些关键个体,从而放弃对该个体所在区域的探索。为了避免出现这类问题,改进算法中将目标空间分割成若干个网格,并计算每个网格中的个体数量,其代表该网格中每个个体的拥挤度。

种群中的贡献个体适应度的计算按照 AR-MOEA 的适应度计算方法<sup>[9]</sup>:

$$\text{fitness}_p = \text{IGD} - \text{NS}(P \setminus \{p\}, R') \quad (4)$$

其中,  $R'$  为自适应参考点集,  $P$  为算法搜索到的种群集合。对于非贡献个体,尽管它对 IGD 值的计算没有价值,但是可以增加种群朝着未知方向进化的可能性,使种群更有可能搜索到全部的 Pareto 前沿,增加种群向未知区域探索的几率,而为了找到可增加种群向未知方向进化可能性的个体,本文考虑在该类个体的适应度计算中加入网格拥挤度。另外,随着搜索过程的进行,种群在进化后期应该更加关注收敛性,放弃一定的分布性,减少向未知区域探索的概率,在进化的后期放弃选择非贡献个体,所以在计算非贡献个体适应度的时候也要考虑种群迭代的次数,综上,非贡献个体的适应度计算方法如下:

$$\text{fitness}_p = (\text{IGD} - \text{NS}(P \setminus \{p\}, R')) \times \left( \frac{1}{\text{Crowding}} \right)^T \quad (5)$$

其中,  $\text{Crowding}$  是网格拥挤度,即每个网格中存在的个体数,  $T$  是种群进化的代数。从公式(5)可以看出,网格拥挤度和迭代次数对与非贡献个体的影响比较大,  $\text{Crowding}$  值越大,随着迭代次数增加,非贡献个体的适应度减小,从而很难被选择进入下一代。

当重新调整个体适应度方法后可以更好选择分

布性更好的个体, 如图 2 所示, P2、P3 和 P6 距离最近的参考点的距离分别是 0.9、0.8 和 0.7, 倘若要在 P1 ~ P6 的非贡献个体中选择 5 个个体进入下一代, 如果按照原算法中个体适应度计算方法, 如图 2(a) 所示, 在选择个体时会首先淘汰 P6 个体, 然后选择 {P1、P2、P3、P4、P5} 进入下一代, 但显然非贡献个体 {P1、P2、P4、P5、P6} 分布性更好; 如果按照改进算法的适应度计算方法, 如图 2(b) 所示, 将目标空间所在个体分割成为 5×5 的网格, 因为 P2、P3 个体所在同一个网格, 这会导致这两个个体的适应度减小, 又因为 P3 相比于 P2 距离参考点更近, 所以淘汰掉 P3 个体, 将 {P1、P2、P4、P5、P6} 选择进入下一代, 寻找分布性更好的个体。

2.2.3 算法流程 AR-MOEA-GC 进一步优化初始参考点集并且调整了非贡献个体的计算方法, 过程如图 3 所示。

算法 1 是 AR-MOEA-GC 的伪代码, 具体如下:

Algorithm 1 The framework of the AR-MOEA-GC algorithm

Input:  $N$  (Population size, Size of reference points)

$P \leftarrow \text{PopulationInitialization}()$  ;

$R \leftarrow \text{RePointCreation}()$  ;

$A \leftarrow P$

$R', A \leftarrow \text{UpDateR}(R, P, A)$

While Termination conditions not met do

$P' \leftarrow \text{MatchingParent}(P, R')$  ;

$O \leftarrow \text{GenerateOffspring}(P')$  ;

$R', A \leftarrow \text{UpDateR}(R', [P; O], A)$  ;

$P \leftarrow \text{EliteChoiing}([P; O], R')$  ;

If  $\Delta e < u$

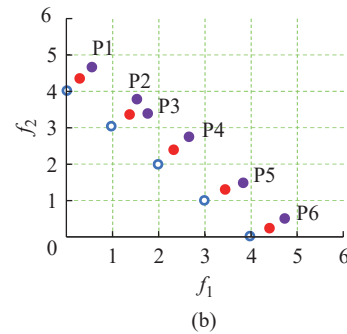
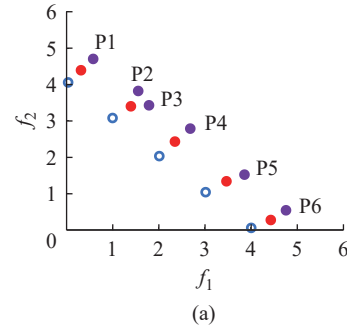
$R \leftarrow \text{deleteRePoint}(R)$

End

End

Return  $P$  ;

算法运行结束后会输出搜索到的种群。首先, 算法随机生成个体数量为  $N$  的种群, 然后生成  $1.1N$  个在目标空间中均匀分布的参考点集  $R$ , 并设置外部存档, 随后更新自适应参考点集  $R'$ 。进入主循环后, 根据每个个体的  $\text{fitness}_p$  采用二元锦标赛策略创建大小为  $N$  的交配池  $P'$ , 其中  $R'$  要参与到计算个体的适应度; 随后, 根据  $P'$  中的个体进行交叉、变异操作, 生成子代个体, 然后进行  $R'$  的调整, 调整的方法同 AR-MOEA。算法 2 详细介绍了算法的精英选择策略, 具体如下:



—○— Reference point; —●— Contributing individual; —●— Non-contributing individual

图 2 AR-MOEA 与 AR-MOEA-GC 适应度计算示意图

Fig. 2 Schematic diagram for fitness calculation of the AR-MOEA and AR-MOEA-GC

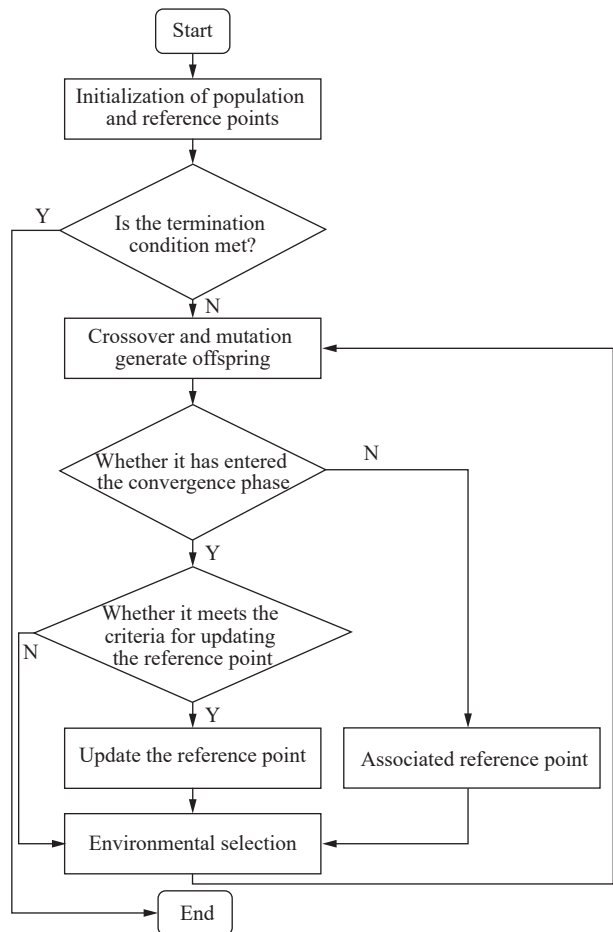


图 3 AR-MOEA-GC 算法流程

Fig. 3 Algorithm flow of the AR-MOEA-GC

## Algorithm 2 Elite selection

Input:  $P$ 、 $O$  (Current population and offspring population)、 $R$  (Adjusted reference points)

Output:  $P$  (Population output by elite selection)

- 1:  $P_s \leftarrow \text{NDSort}(P)$ ;
- 2:  $P_{\text{final}} \leftarrow \text{ChooseNonCriticalLayerIndividuals}(P_s)$ ;
- 3:  $\text{fitness}_p \leftarrow \text{CalculateFitnessoflast}(P_s)$ ;
- 4:  $P_{\text{final}} \leftarrow \text{CriticalIndividualSelection}()$

对于非贡献个体,为了增加种群向未知区域探索的概率,其适应度的大小与该个体对 IGD-NS 指标的贡献和在目标空间中的网格拥挤度有关;对于贡献个体,其适应度的大小就是该个体对 IGD-NS 指标的贡献值。在计算子代个体和父代个体的适应度之后,根据个体适应度大小淘汰个体,直到满足种群所要的个体数量。在精英选择之后,计算这一代种群的熵,然后计算本代种群与上一代种群的熵差,通过比较熵差与阈值  $u$  的大小,判断是否要调整初始参考点集  $R$ 。要注意的是,自适应参考点只参与计算个体的适应度,且每一代都会基于初始参考点集  $R$  进行调整。

### 2.3 算法复杂度分析

对于一个多目标优化问题,其优化问题和种群数量分别为  $M$  和  $N$ 。在 AR-MOEA-GC 中选择配对、生成后代、更新存档、参考点关联个体、删除参考点以及环境选择这几个步骤,只需要分析这几个步骤的时间复杂度即可。在选择配对中需要计算  $N$  个个体的适应度,所以其复杂度为  $O(N)$ ;生成后代的数目与种群大小一致,其复杂度可以近似为  $O(N)$ ;更新存档和参考点关联个体由于存档和参考点集的大小都是  $1.2N$ ,所以其复杂度可以认为  $O(1.2N)$ ;删除参考点个数为  $0.1N$ ,其复杂度可以认为是  $O(0.1N)$ ;环境选择是最复杂的阶段,在最坏的情况下选择一个个体后需要重新计算剩余个体的适应度,所以其复杂度为  $O(N^2)$ 。综上,AR-MOEA-GC 算法的复杂度为:  $O(N)+O(N)+2O(1.2N)+O(0.1N)+O(N^2)=O(N^2)$ 。

相比于 AR-MOEA-GC,AR-MOEA 缺少的步骤是参考点关联个体和删除参考点,并且其存档的大小为  $N$ ,故其复杂度为:  $O(N)+O(N)+O(N)+O(N^2)=O(N^2)$ 。综上所述,单次迭代中 AR-MOEA 的算法复杂度与 AR-MOEA-GC 的算法复杂度相同。

## 3 仿真测试与分析

### 3.1 测试环境

计算机型号:联想拯救者 R7000;CPU:AMD

Ryzen 5 4600H with Radeon Graphics 3.00 GHz;运行内存:16.0 GB;操作系统:Window 10 中文版;算法运行软件:Matlab R2022b;在 PlatEMO v3.7<sup>[24]</sup> 开源平台实现算法。

为了验证 AR-MOEA-GC 的有效性,将改进算法与 AR-MOEA、A-NSGA-III、MOEA/DD、NSGA-II、NSGA-III 和 AR-NSGA-III 这 6 个先进的算法进行对比。测试函数分别是 DTLZ 系列、WFG 系列和 MaF 系列。表 1 展示了各个测试函数的目标个数、决策变量个数以及真实 Pareto 前沿的形状。在比较算法中 MOEA/DD 的 delta 值设置为 0.9,其余算法参考点都采用两层方法<sup>[25]</sup>生成。AR-MOEA-GC 算法中将种群所在的目标空间划分成为  $5^M$  个网格, $M$  为测试函数的目标个数,初始参考点集的大小设置为 105。设置每个算法生成的种群大小  $N$  为 100,算法运行中最大评价次数 maxFE 为 10000,进化算子设置为多项式突变和模拟二进制交叉方法,个体之间交叉的概率为 0.9,个体变异的概率是  $\frac{1}{D}$ ,其中  $D$  是测试函数的决策变量的个数。

表 1 测试函数的目标数、决策变量以及真实 Pareto 前沿形状

Table 1 Numbers of objectives for the test function, decision variables and true Pareto front shape

Function	Number of the objectives	Number of the decision variables	True Pareto front shape
WFG1~3	3	12	Irregular
MaF2、MaF4、13	3	12	Irregular
DTLZ5~7	3	12	Irregular
WFG4~8	3	12	Regular
MaF1~15 (Excluding MaF2、4、13)	3	12	Regular
DTLZ1~4	3	12	Regular

### 3.2 测试结果

因为算法具有一定的随机性,为了保证测试的公平性并且更直观地判断算法性能的优劣,每种算法在每个测试函数上独立运行 30 次,采用可以综合反映种群分布性、收敛性的 IGD 指标和 HV(Hyper-Volume, HV)指标来评价算法的性能。其中 IGD 指标<sup>[26]</sup>的计算公式如下:

$$\text{IGD}(\text{PF}_t, \text{PF}^*) = \frac{\sum_{v \in \text{PF}^*} d(v, \text{PF}_t)}{|\text{PF}^*|} \quad (6)$$

其中,  $\text{PF}_t$  是真实的 Pareto 最优解,  $\text{PF}^*$  是算法搜索到的非支配解集,  $|\text{PF}^*|$  是搜索到的非支配解集的大

小,  $d(v, PF_t)$  表示个体  $v$  到  $PF_t$  的最小欧氏距离。IGD 指标越小, 说明种群的收敛性和分布性越好, 即算法的综合性能更好。HV 指标<sup>[2]</sup>的计算公式如下:

$$HV = \text{volume} \left( \bigcup_{i=1}^{|\text{PF}_t|} v_i \right) \quad (7)$$

其中,  $v_i$  是搜索到的个体  $i$  与参考点所形成的超体积。HV 的几何意义就是种群中的个体和参考点在目标空间所构成的体积, HV 指标越大, 说明种群在目标空间中所支配的空间就越大, 算法的性能也更好。

此外, 为了判断 AR-MOEA-GC 与 AR-MOEA 算法的性能, 也单独计算了种群的 Spacing 指标和 GD 指标。其中, GD 指标<sup>[8]</sup>的计算公式如下:

$$GD_t(\text{PF}_t, \text{PF}^*) = \frac{\sum_{v \in \text{PF}_t} d(v, \text{PF}^*)}{|\text{PF}_t|} \quad (8)$$

其中,  $\text{PF}^*$  是优化问题的真实 Pareto 前沿,  $\text{PF}_t$  是  $t$  时刻种群搜索到的 Pareto 前沿。  $d(v, \text{PF}^*)$  是  $\text{PF}_t$  中的个体  $v$  与真实 Pareto 前沿  $\text{PF}^*$  上最近采样点的欧式距离。GD 指标为搜索到的个体距离真实 Pareto 前沿距离的平均值, 所以 GD 指标越小, 种群的收敛性就越好。要注意的是, GD 指标小并不意味着种群的质量高, 因为有可能种群只找到一部分的真实 Pareto 前沿, IGD 指标与 GD 指标的示意图见图 4。

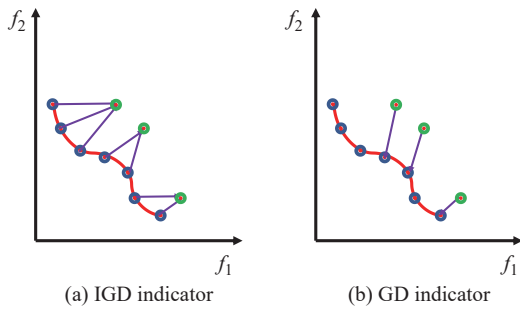


图 4 IGD 指标与 GD 指标示意图

Fig. 4 Diagram illustrating the IGD and GD indicators

Spacing 指标<sup>[27]</sup>的计算公式如下:

$$\text{Spacing}_t = \sqrt{\frac{1}{|\text{PF}_t| - 1} \sum_{i=1}^{|\text{PF}_t|} (d_i - \bar{d})^2} \quad (9)$$

$$d_i = \min_{k=1, \dots, |\text{PF}_t|, k \neq i} \left( \sum_{j=1}^M |\text{PF}_t(i, j) - \text{PF}_t(k, j)| \right) \quad (10)$$

其中,  $\text{PF}_t$  是  $t$  时刻搜索到的 Pareto 前沿,  $d_i$  是  $\text{PF}_t$  中第  $i$  个个体与离它最近的个体之间的欧式距离。Spacing 指标用于评价种群的分布性, 其值越小说明种群的分布性就越好。

图 5 所示为 7 种算法根据 IGD 指标的下降曲

线。AR-MOEA-GC 算法向真实 Pareto 前沿收敛的速度在 7 个算法中不是最快, 但是在迭代 1000 次的条件下, 其 IGD 曲线是在 7 种算法中最接近 0, 说明该算法搜索到的种群更加贴近真实的 Pareto 前沿。

表 2 和表 3 分别是 7 种算法在 3 类测试函数上运行 30 次后 IGD 值和 HV 值, 黑体表示最优结果。3 类测试函数共 28 个测试问题, 与其他 6 种算法相比较, AR-MOEA-GC 在 IGD 指标上有 17 个达到最优, 在 HV 指标上有 15 个达到最优。在表 2 和表 3 的最后一行统计了秩和检验后每种算法与 AR-MOEA-GC 比较的结果, 在 28 个测试问题上另外 6 种算法的表现远不如 AR-MOEA-GC。要注意的是, 在 28 个测试函数中, 7 种算法在最大评价次数为 10000 的条件下, 都未能搜索到 DTLZ1、DTLZ3、MaF3、MaF4 的真实 Pareto 前沿, 故 HV 指标的计算都为 0。

表 4 和表 5 分别是 7 种算法在 3 类测试函数二维测试用例上运行 30 次后的 IGD 值和 HV 值, 黑体表示最优结果。与其他 6 种算法相比较, AR-MOEA-GC 在 27 个测试案例中有 14 个达到最优, 在 HV 指标上有 14 个达到最优, 在表 4 和表 5 的最后一行统计了秩和检验后每种算法与 AR-MOEA-GC 比较的结果。根据秩和检验结果来看, 在 27 个测试案例上 A-NSGA-III、MOEA/DD、NSHA-III 和 AR-NSGA-III 的性能明显不及 AR-MOEA-GC, AR-MOEA 和 NSGA-II 的性能与 AR-MOEA-GC 相当。要注意的是, 在 27 个测试函数中, 7 种算法在最大评价次数为 10000 的条件下, 都未能搜索到 DTLZ1、DTLZ3、MaF3、MaF4 的真实 Pareto 前沿, 故 HV 指标的计算都为 0, 而且对于 MaF3 问题没有 3 维的优化问题。综上, 从 IGD 指标和 HV 指标的结果来看 AR-MOEA-GC 的在三维优化问题上的性能要优于在二维多目标优化问题上的性能。

### 3.3 测试结果分析

具体来看, 对于真实 Pareto 前沿不规则的问题, AR-MOEA-GC 在 IGD 值方面显著优于其他 6 种算法。在 9 个真实 Pareto 前沿不规则(表 1)的问题上, 改进的算法在 6 个测试问题都显示了最佳性能。图 6 展示了每个算法在 MaF2 上找到的种群分布, 红框所示为每种算法在真实 Pareto 前沿上分布的不足, 从 MOEA/DD、NSGA-II 和 NSGA-III 可以明显地看出种群在 Pareto 前沿上分布得极不均匀。其余 3 种算法 AR-MOEA、A-NSGA-III、AR-NSGA-III 和 AR-MOEA-GC 相比较, 改进的算法找到分布更均匀的种群, 但在真实 Pareto 前沿上小部分区域分布还是不均

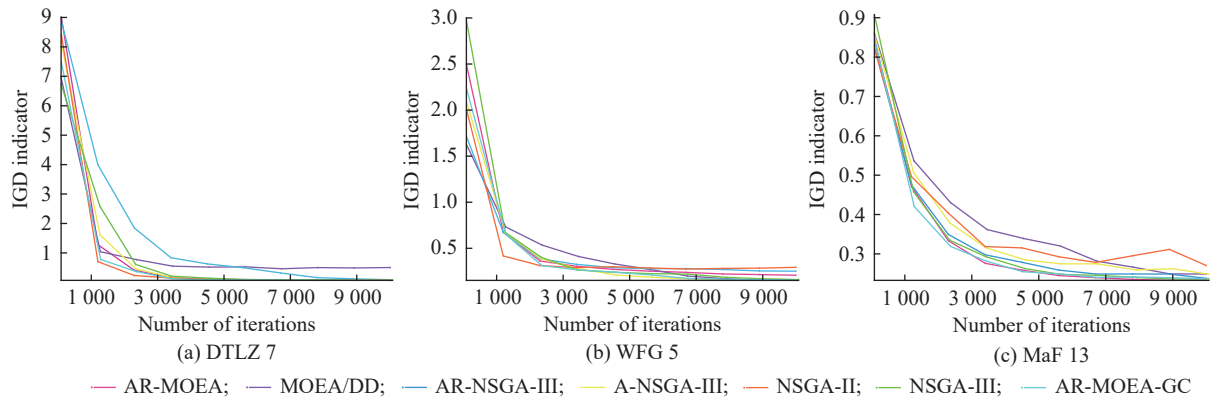


图 5 7 种算法在 DTLZ7、WFG8 和 MaF13 上的 IGD 下降曲线

Fig. 5 IGD reduction curves of seven algorithms on DTLZ7, WFG8, and MaF13

表 2 7 种算法在 3 类三维测试函数上平均 IGD 值

Table 2 Average IGD values of seven algorithms on three types of three-dimensional test functions

Function	IGD						
	AR-MOEA	A-NSGA-III	MOEA/DD	AR-NSGA-III	NSGA-II	NSGA-III	AR-MOEA-GC
WFG1	0.6180	0.6640	1.2600	0.7090	<b>0.5630</b>	0.6860	0.6130
WFG2	0.1670	0.1880	0.1940	0.1730	0.2400	0.1690	<b>0.1610</b>
WFG3	0.1620	0.1720	0.4190	0.1510	<b>0.1330</b>	0.1520	0.1690
WFG4	0.2290	0.2610	0.2480	0.2430	0.2810	0.2320	<b>0.2190</b>
WFG5	0.2330	0.2620	0.2540	0.2390	0.2850	0.2380	<b>0.2240</b>
WFG6	0.2690	0.3030	0.3130	0.2830	0.3360	0.2790	<b>0.2620</b>
WFG7	0.2320	0.2580	0.2760	0.2330	0.2820	0.2320	<b>0.2220</b>
WFG8	0.3110	0.3410	0.3410	0.3190	0.3770	0.3210	<b>0.3070</b>
DTLZ1	<b>2.5500</b>	4.0900	6.0000	4.0900	2.8800	3.8300	3.3200
DTLZ2	0.0550	0.0588	0.0553	0.0562	0.0696	0.0549	<b>0.0529</b>
DTLZ3	8.9700	11.1000	19.1000	13.2000	8.2400	10.2000	<b>7.9200</b>
DTLZ4	0.3790	0.2190	<b>0.0719</b>	0.2320	0.1330	0.1360	0.1350
DTLZ5	0.0058	0.0113	0.0313	0.0122	0.0066	0.0127	<b>0.0053</b>
DTLZ6	0.7320	<b>0.6820</b>	0.6860	0.7340	0.7320	0.7320	0.7320
DTLZ7	0.2960	0.0966	0.3060	0.1120	0.0961	0.0873	<b>0.0642</b>
MaF1	0.0441	0.0458	0.0800	0.0553	0.0611	0.0619	<b>0.0433</b>
MaF10	0.6030	0.6780	1.2300	0.7210	<b>0.5310</b>	0.6760	0.6030
MaF11	0.1680	0.1810	0.1940	0.1720	0.2250	0.1710	<b>0.1630</b>
MaF12	0.2350	0.2760	0.2650	0.2430	0.2900	0.2390	<b>0.2260</b>
MaF13	0.2130	0.3040	0.2150	0.2680	0.2500	0.3280	<b>0.1910</b>
MaF14	0.4640	0.4350	0.4270	0.4520	<b>0.3930</b>	0.4810	0.4070
MaF15	0.1600	0.2120	<b>0.1570</b>	0.1910	0.2760	0.1930	0.1610
MaF2	0.0331	0.0325	0.0580	0.0330	0.0483	0.0367	<b>0.0312</b>
MaF3	91.80	178.00	902.00	275.00	99.00	219.00	<b>197.00</b>
MaF4	27.60	34.50	39.80	41.20	30.50	34.60	<b>26.90</b>
MaF5	1.2200	0.6800	<b>0.4430</b>	0.7350	0.6470	0.8400	0.4910
MaF6	0.0066	0.0111	0.0328	0.0118	<b>0.0058</b>	0.0129	0.0063
MaF7	0.2100	0.0861	0.2990	0.1030	0.0788	0.0882	<b>0.0641</b>
+/-/=	1/18/9	0/24/4	3/23/2	1/24/3	6/17/5	1/26/1	

"+/-/=" represent significant better, worse, and statistically similar performance compared to AR-MOEA-GC after the rank-sum test

表3 7种算法在3类测试函数上平均HV值  
Table 3 Average HV values of seven algorithms on three types of three-dimensional test functions

Function	HV						
	AR-MOEA	A-NSGA-III	MOEA/DD	AR-NSGA-III	NSGA-II	NSGA-III	AR-MOEA-GC
WFG1	0.6630	0.6380	0.3580	0.6170	<b>0.6970</b>	0.6300	0.6650
WFG2	0.9130	0.9010	0.8920	0.9080	0.9030	0.9110	<b>0.9160</b>
WFG3	0.3510	0.3490	0.2410	0.3600	<b>0.3750</b>	0.3590	0.3480
WFG4	0.5310	0.5120	0.5200	0.5240	0.5040	0.5300	<b>0.5340</b>
WFG5	0.5050	0.4890	0.4940	0.4970	0.4820	0.5040	<b>0.5060</b>
WFG6	0.4770	0.4600	0.4500	0.4730	0.4480	0.4710	<b>0.4800</b>
WFG7	<b>0.5330</b>	0.5190	0.5050	0.5310	0.5100	0.5320	0.5330
WFG8	0.4480	0.4310	0.4280	0.4430	0.4260	0.4430	<b>0.4490</b>
DTLZ1	0	0	0	0	0	0	0
DTLZ2	0.5560	0.5490	0.5550	0.5570	0.5280	0.5560	<b>0.5580</b>
DTLZ3	0	0	0	0	0	0	0
DTLZ4	0.4000	0.4810	<b>0.5470</b>	0.4750	0.5060	0.5190	0.5230
DTLZ5	0.1980	0.1950	0.1820	0.1940	0.1980	0.1930	<b>0.1990</b>
DTLZ6	<b>0.3290</b>	0.2950	0.2790	0.3160	0.3290	0.3190	0.3290
DTLZ7	0.2480	0.2650	0.2310	0.2630	0.2630	0.2660	<b>0.2740</b>
MaF1	0.2160	0.2140	0.1880	0.2060	0.2000	0.2020	<b>0.2190</b>
MaF10	0.6690	0.6310	0.3710	0.6100	<b>0.7170</b>	0.6310	0.6720
MaF11	0.9120	0.9040	0.8900	0.9090	0.9060	0.9090	<b>0.9130</b>
MaF12	0.5030	0.4760	0.4880	0.4910	0.4800	0.5010	<b>0.5040</b>
MaF13	0.3460	0.3440	0.3280	0.3430	0.2960	0.3370	<b>0.3500</b>
MaF14	0.3660	0.3360	0.2840	0.3130	<b>0.4040</b>	0.2770	0.3990
MaF15	0.3760	0.3150	<b>0.4140</b>	0.3250	0.2120	0.3240	0.3560
MaF2	0.2380	0.2360	0.1970	0.2390	0.2310	0.2360	<b>0.2390</b>
MaF3	0	0	0	0	0	0	0
MaF4	0	0	0	0	0	0	0
MaF5	0.4390	0.4910	0.5190	0.4910	0.4930	0.4750	<b>0.5230</b>
MaF6	0.1970	0.1940	0.1720	0.1930	<b>0.1980</b>	0.1920	0.1970
MaF7	0.2560	0.2670	0.2300	0.2630	0.2660	0.2660	<b>0.2740</b>
+/-/=	2/10/16	0/21/7	2/22/4	1/19/8	4/19/5	1/21/6	

"+/-/=" represent significant better, worse, and statistically similar performance compared to AR-MOEA-GC after the rank-sum test

表 4 7 种算法在 3 类二维测试函数上平均 IGD 值  
Table 4 Average IGD values of seven algorithms on three types of two-dimensional test functions

Function	IGD						
	AR-MOEA	A-NSGA-III	MOEA/DD	AR-NSGA-III	NSGA-II	NSGA-III	AR-MOEA-GC
WFG1	0.4070	0.4690	0.5820	0.5160	<b>0.3270</b>	0.4910	0.3430
WFG2	0.0316	0.0271	0.0593	0.0351	<b>0.0205</b>	0.0271	0.0210
WFG3	0.0269	0.0352	0.0432	0.0360	0.0276	0.0316	<b>0.0252</b>
WFG4	0.0210	0.0274	0.0277	0.0287	0.0216	0.0255	<b>0.0200</b>
WFG5	0.0682	0.0721	0.0700	0.0729	0.0679	0.0682	<b>0.0672</b>
WFG6	0.0932	0.0891	0.0968	0.1040	<b>0.0888</b>	0.0971	0.0905
WFG7	0.0174	0.0220	0.0222	0.0251	0.0192	0.0193	<b>0.0170</b>
WFG8	0.1210	0.1290	0.1260	0.1280	0.1160	0.1250	<b>0.1150</b>
DTLZ1	4.0500	5.2100	12.9000	5.7000	3.3100	5.3500	<b>2.7100</b>
DTLZ2	<b>0.0042</b>	0.0052	0.0043	0.0057	0.0052	0.0043	0.0043
DTLZ3	11.50	14.40	30.30	15.30	12.00	14.40	<b>9.83</b>
DTLZ4	0.2990	0.1030	0.0043	0.0545	0.1530	0.1770	<b>0.0043</b>
DTLZ5	0.0042	0.0052	0.0043	0.0056	0.0052	0.0043	<b>0.0042</b>
DTLZ6	0.4110	0.4020	0.4190	0.4140	0.4140	<b>0.3970</b>	0.4140
DTLZ7	0.0928	0.0056	0.0187	0.0059	0.0199	0.0057	<b>0.0052</b>
MaF1	<b>0.0037</b>	0.0042	0.0037	0.0046	0.0048	0.0037	0.0041
MaF10	0.3930	0.4960	0.5670	0.5050	<b>0.3500</b>	0.4770	0.3840
MaF11	0.0368	0.0281	0.0612	0.0278	<b>0.0194</b>	0.0283	0.0349
MaF12	0.0392	0.0363	0.0546	0.0369	0.0376	0.0339	<b>0.0333</b>
MaF14	0.5220	0.6150	<b>0.5170</b>	0.5670	0.6230	0.5950	0.6160
MaF15	0.1040	0.0752	<b>0.0301</b>	0.0738	0.1040	0.0862	0.1120
MaF2	<b>0.0021</b>	0.0029	0.0044	0.0026	0.0028	0.0022	0.0022
MaF3	215.00	377.00	1680.00	289.00	126.00	361.00	<b>113.00</b>
MaF4	12.3000	21.3000	39.1000	20.4000	10.3000	18.7000	<b>8.9200</b>
MaF5	0.4790	0.0827	<b>0.0139</b>	0.2150	0.2150	0.3460	0.4120
MaF6	0.0076	0.0127	0.0216	0.0140	<b>0.0063</b>	0.0126	0.0067
MaF7	0.1080	0.0200	0.0182	0.0061	0.0053	0.0057	<b>0.0051</b>
+/-/=	4/5/18	4/20/3	4/19/4	4/20/3	7/11/19	6/14/7	

"+/-/=" represent significant better, worse, and statistically similar performance compared to AR-MOEA-GC after the rank-sum test

匀。这是因为 AR-MORA-GC 有初始参考点集调整和自适应参考点策略, 在 Pareto 前沿不规则的情况下, 自适应参考点策略会删除不在 Pareto 区域的参考

点同时补充在 Pareto 前沿附近的个体, 让种群尽可能向全局最优解进化; 同时, 在算法进入收敛阶段删除掉部分初始参考点, 进一步明确种群进化的方向, 放

表5 7种算法在3类二维测试函数上平均HV值  
Table 5 Average HV values of seven algorithms on three types of two-dimensional test functions

Function	HV						
	AR-MOEA	A-NSGA-III	MOEA/DD	AR-NSGA-III	NSGA-II	NSGA-III	AR-MOEA-GC
WFG1	0.4880	0.4440	0.3750	0.4170	<b>0.5290</b>	0.4320	0.5160
WFG2	0.6240	0.6220	0.6040	0.6200	<b>0.6250</b>	0.6220	0.6240
WFG3	0.5710	0.5660	0.5600	0.5660	0.5710	0.5680	<b>0.5710</b>
WFG4	<b>0.3420</b>	0.3390	0.3360	0.3390	0.3410	0.3390	0.3420
WFG5	0.3090	0.3070	0.3070	0.3090	<b>0.3100</b>	0.3090	0.3100
WFG6	0.2980	<b>0.3010</b>	0.2940	0.2940	0.3000	0.2960	0.3060
WFG7	0.3430	0.3410	0.3390	0.3420	0.3430	0.3420	<b>0.3440</b>
WFG8	0.2840	0.2800	0.2800	0.2810	<b>0.2860</b>	0.2810	0.2860
DTLZ1	0	0	0	0	0	0	0
DTLZ2	0.3470	0.3450	0.3460	0.3470	0.3460	0.3470	<b>0.3470</b>
DTLZ3	0	0	0	0	0	0	0
DTLZ4	0.2440	0.3120	0.3460	0.3300	0.2950	0.2870	0.3380
DTLZ5	0.3470	0.3450	0.3460	0.3470	0.3460	0.3470	<b>0.3470</b>
DTLZ6	0.3390	0.3340	0.3290	0.3450	0.3460	0.3310	<b>0.3470</b>
DTLZ7	0.2290	0.2420	0.2400	0.2420	0.2400	0.2420	<b>0.2420</b>
MaF1	<b>0.5820</b>	0.5810	0.5820	0.5810	0.5800	0.5820	0.5820
MaF10	0.4940	0.4310	0.3840	0.4270	<b>0.5170</b>	0.4380	0.5200
MaF11	0.6230	0.6210	0.6040	0.6210	<b>0.6260</b>	0.6210	0.6260
MaF12	0.3270	0.3280	0.3180	0.3290	0.3280	<b>0.3300</b>	0.3240
MaF13	0.3460	0.3440	0.3280	0.3430	0.2960	0.3370	<b>0.3500</b>
MaF14	<b>0.1360</b>	0.0931	0.1020	0.0945	0.1070	0.0958	0.0135
MaF15	0.6450	0.6990	<b>0.7860</b>	0.6990	0.6530	0.6830	0.6360
MaF2	<b>0.2070</b>	0.2060	0.2050	0.2070	0.2070	0.2070	0.2070
MaF3	0	0	0	0	0	0	0
MaF4	0	0	0	0	0	0	0
MaF6	0.3420	0.3380	0.3210	0.3370	<b>0.3440</b>	0.3370	0.3470
MaF7	0.2270	0.2400	0.2400	0.2420	<b>0.2430</b>	0.2420	0.2420
+/-/=	2/9/17	3/18/7	4/18/6	2/17/9	5/9/14	3/16/9	

"+/-/=" represent significant better, worse, and statistically similar performance compared to AR-MOEA-GC after the rank-sum test

弃探索关联个体少的参考点所在的区域, 加快种群向真实 Pareto 前沿收敛的速度。

表6和表7分别给出了不连续问题的平均IGD指标和HV指标, 在4个问题上, 改进算法在

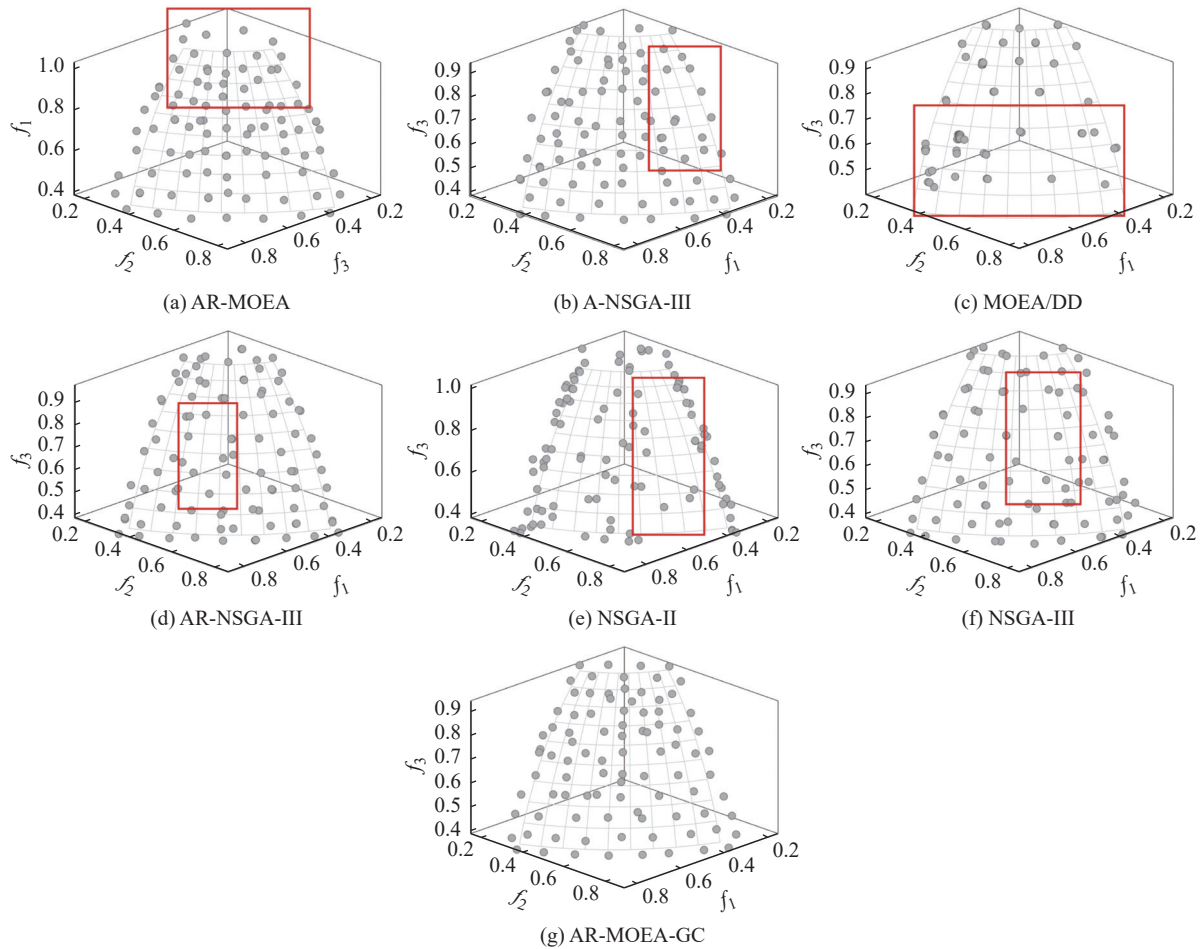


图 6 7 种算法在 MaF2 上种群分布结果

Fig. 6 Population distribution results of seven algorithms on MaF2

表 6 7 种算法在真实 Pareto 前沿不连续问题上的平均 IGD 值

Table 6 Average IGD values of seven algorithms on problems with discontinuous true Pareto fronts

Function	IGD						
	AR-MOEA	A-NSGA-III	MOEA/DD	NSGA-II	NSGA-III	AR-NSGA-III	AR-MOEA-GC
DTLZ7	0.2960	0.0966	0.3060	0.0961	0.0873	0.1120	<b>0.0642</b>
IMOP5	0.1430	0.1410	0.4030	<b>0.0757</b>	0.1170	0.1020	0.0998
IMOP8	0.1760	0.1780	0.5170	0.1600	0.1370	0.1400	<b>0.1340</b>
WFG2	0.1670	0.1880	0.1940	0.2400	0.1690	0.1730	<b>0.1610</b>

表 7 7 种算法在真实 Pareto 前沿不连续问题上的平均 HV 值

Table 7 Average HV values of seven algorithms on problems with discontinuous true Pareto fronts

Function	HV						
	AR-MOEA	A-NSGA-III	MOEA/DD	NSGA-II	NSGA-III	AR-NSGA-III	AR-MOEA-GC
DTLZ7	0.2480	0.2650	0.2310	0.2630	0.2660	0.2630	<b>0.2740</b>
IMOP5	0.5090	0.5010	0.4430	0.5070	0.5070	0.5140	<b>0.5180</b>
IMOP8	0.4840	0.4710	0.1630	0.4660	0.4880	0.4830	<b>0.5000</b>
WFG2	0.9130	0.9010	0.8920	0.9030	0.9110	0.9080	<b>0.9160</b>

IGD 有 3 个达到最优, HV 指标全部达到最优, 这说明针对真实 Pareto 前沿不连续的问题, 改进算法也能够找到较为不错的解集。图 7 展示了 7 种算法在 WFG8 问题上种群的分布, 红色箭头所示为另外 6 种算法搜索到种群在真实 Pareto 前沿上的分布。图 8 展示了 7 种算法在 IMOP5 问题上种群的分布, 改进算法在目标空间上 8 个不连续分布的 Pareto 前沿中找到 7 个, 相较于 AR-MOEA, AR-MOEA-GC 可以找到分布更均匀的解; 同时, 基于参考点调整的算法 A-NSGA-III 得到个体大都分布在各个 Pareto 前沿的边缘, 未能搜索到每个 Pareto 内部解; NSGA-II 算法的解分布到不连续的 Pareto 前沿上, 但在每个独立的 Pareto 前沿上个体分布得不均匀。图 9 所示为 7 种算法种群的 IGD 变化曲线, 可以看到 MOEA/DD 算法 IGD 指标在前期快速下降, 但根据最终种群分布(图 8)可以推测算法在前期陷入局部最优, 从而只探索这一小部分区域, 使种群快速地收敛到部分的 Pareto 前沿, 导致 IGD 指标快速下降。尽管 AR-MOEA-GC 种群的 IGD 指标下降不是最快的, 但是最终可以

找到大部分的解, 这说明算法在搜索前期适当地放弃一定的收敛性转而去多做一些探索是正确的。

### 3.4 划分网格数量的影响

目标空间中划分网格的数量也会对算法的性能产生一定的影响, 图 10 分别展示了每个维度划分成 5~50 网格(每个测试算法的网格数量间隔为 5)的 AR-MOEA-GC 算法在 WFG 系列和 MaF 系列测试问题上的 IGD 和 HV 值。可以看到, 随着网格数量的增加, 算法的 IGD 值并不是越来越小, 某些问题上网格数的增加还会导致算法的性能变得越来越差。从统计上来讲, 当算法取  $5^M \sim 10^M$  个网格时, IGD 值达到最优的问题数目最多, 因为划分网格数量越多, 算法的计算复杂度就越高, 为了提高算法运行的效率, 最终的改进算法选择了将种群所在的空间划分为  $5^M$  个网格。

### 3.5 AR-MOEA-GC 与 AR-MOEA 性能对比

AR-MOEA-GC 与 AR-MOEA 最大的不同点主要有两点: (1) 根据种群进化的阶段, 调整初始参考点集, 增加了初始点集调整策略; (2) 区分贡献个体

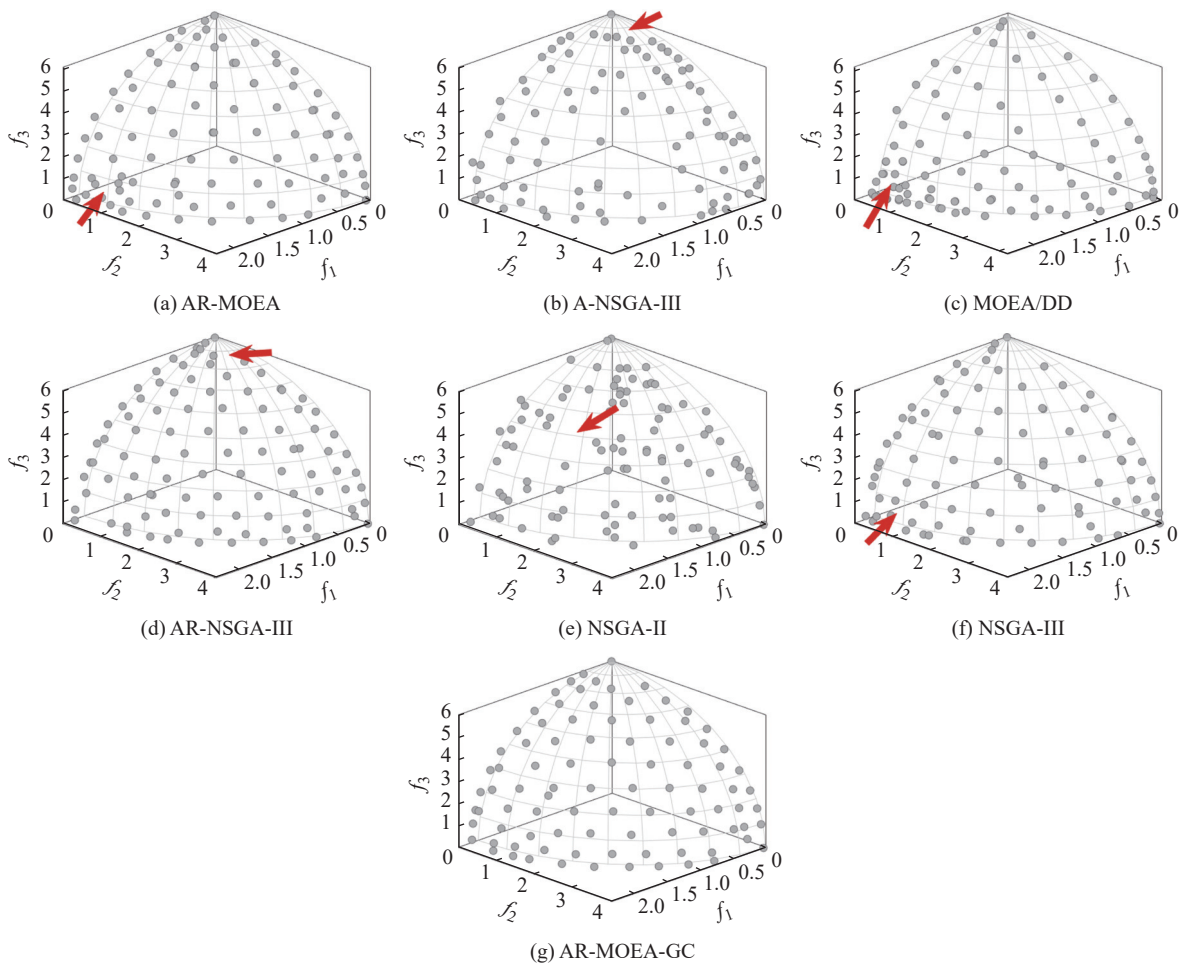


图 7 7 种算法在 WFG8 问题上找到的种群分布

Fig. 7 Population distributions found by seven algorithms on the WFG8 problem

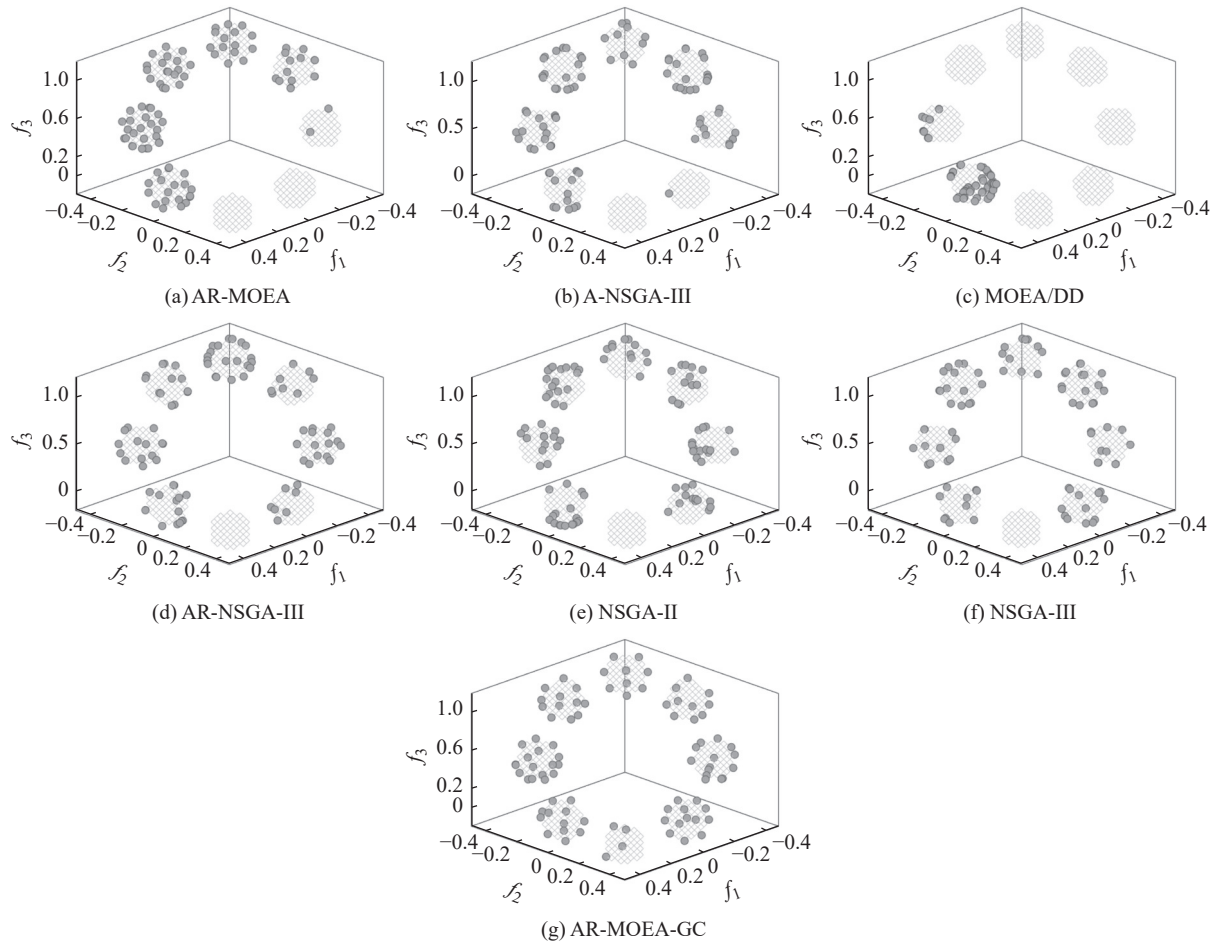


图 8 7 种算法在 IMOP5 找到的种群分布

Fig. 8 Population distributions found by seven algorithms on the IMOP5 problem

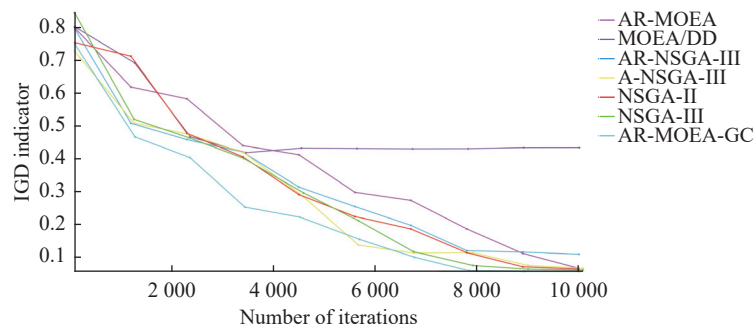


图 9 7 种算法在 IMOP5 上的 IGD 变化曲线

Fig. 9 IGD variation curves of seven algorithms on IMOP5

与非贡献个体,利用网格拥挤度计算非贡献个体的适应度。

这两点改进的目的是增强种群的分布性,避免算法在搜索前期就陷入局部最优。算法调整了非贡献个体适应度的计算方法,增大了种群搜索位置区域的概率,可以增强种群的分布性,但收敛性下降。表 8 展示了 AR-MOEA 与 AR-MOEA-GC 在各个三维测试问题上运行 30 次的 GD 和 Spacing 值的均值,

在 Spacing 指标上, AR-MOEA 算法在 28 个不同的测试问题上只有 1 个显著优于 AR-MOEA-GC 算法;在 GD 指标上, AR-MOEA 算法在 28 个不同的测试问题上有 9 个显著优于 AR-MOEA-GC 算法,这与之前的预测结果是一致的。尽管在适应度计算上放弃了一定的收敛性, AR-MOEA-GC 在 GD 指标上只有一个问题显著劣于 AR-MOEA,有 19 个在统计上相似,所以 AR-MOEA-GC 搜索到的种群的收敛性还是在可

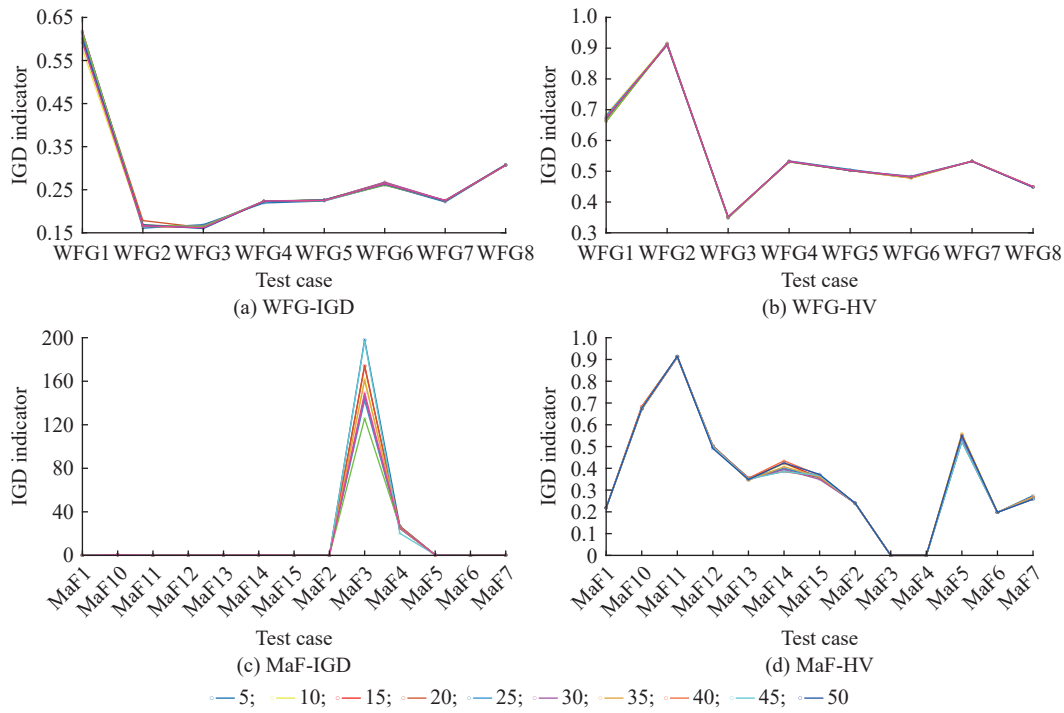


图 10 不同网格数量对 WFG 问题和 MaF 系列问题 IGD 值和 HV 值的影响

Fig. 10 Impact of different grid sizes on the IGD and HV values for WFG problems and MaF series problems

表 8 AR-MOEA 与 AR-MOEA-GC 在各个测试问题上的平均 GD 和 Spacing 值

Table 8 Average GD and Spacing values of AR-MOEA and AR-MOEA-GC on various test problems

Function	Spacing		GD	
	AR-MOEA	AR-MOEA-GC	AR-MOEA	AR-MOEA-GC
WFG1	0.4420	<b>0.3830</b>	0.0520	<b>0.0509</b>
WFG2	0.2030	<b>0.1740</b>	0.0065	<b>0.0063</b>
WFG3	0.1610	<b>0.1210</b>	0.0854	<b>0.0832</b>
WFG4	0.2960	<b>0.1890</b>	0.0058	<b>0.0060</b>
WFG5	0.2930	<b>0.1920</b>	0.0084	<b>0.0087</b>
WFG6	0.3000	<b>0.1920</b>	0.0149	<b>0.0152</b>
WFG7	0.3000	<b>0.2050</b>	0.0064	<b>0.0067</b>
WFG8	0.2990	<b>0.2060</b>	0.0224	<b>0.0228</b>
MaF1	0.0660	<b>0.0531</b>	0.0006	<b>0.0006</b>
MaF10	0.4160	<b>0.4090</b>	0.0502	<b>0.0492</b>
MaF11	0.2100	<b>0.1780</b>	0.0069	<b>0.0076</b>
MaF12	0.2860	<b>0.1920</b>	0.0088	<b>0.0090</b>
MaF13	0.1160	<b>0.1310</b>	0.0456	<b>0.0536</b>
MaF14	0.2220	<b>0.1880</b>	0.0797	<b>0.0738</b>
MaF15	0.9700	<b>1.9700</b>	0.2350	<b>0.5330</b>
MaF2	0.0445	<b>0.0304</b>	0.0018	<b>0.0018</b>
MaF3	1300.00	<b>5560.00</b>	2420.00	<b>8590.00</b>
MaF4	35.8000	<b>36.8000</b>	8.3200	<b>10.2000</b>
MaF5	0.2080	<b>0.1850</b>	0.0022	<b>0.0027</b>
MaF6	0.0129	<b>0.0120</b>	0.0002	<b>0.0004</b>
MaF7	0.0769	<b>0.0774</b>	0.0017	<b>0.0020</b>
DTLZ1	0.3640	<b>0.4290</b>	0.4230	<b>0.5340</b>
DTLZ2	0.0861	<b>0.0533</b>	0.0006	<b>0.0006</b>
DTLZ3	1.1700	<b>1.2700</b>	1.8700	<b>1.4700</b>
DTLZ4	0.0464	<b>0.0453</b>	0.0004	<b>0.0006</b>
DTLZ5	0.0133	<b>0.0105</b>	0.0002	<b>0.0002</b>
DTLZ6	0.0120	<b>0.0092</b>	0.0969	<b>0.0967</b>
DTLZ7	0.0708	<b>0.0761</b>	0.0016	<b>0.0019</b>
+/-/=	1/16/11		9/0/19	

"+/-/=" represent significant better, worse, and statistically similar performance compared to AR-MOEA-GC after the rank-sum test

接受范围的。

## 4 结 语

本文针对 AR-MOEA 算法在 3 维多目标优化问题上容易陷入局部最优的问题进行改进,提出了 AR-MOEA-GC 算法,改进算法在原算法的基础上,利用网格拥挤度,调整了非贡献个体的适应度计算方法并且添加了自适应初始参考点集调整策略,与原算法相对比,在收敛性和分布性上均有提升。将该算法与 6 个先进的多目标算法在 3 类标准测试函数上进行仿真测试,结果表明本文提出的算法有一定的竞争力。

### 参考文献:

- [1] 韩晓霞,程铭,刘树森,等.化工过程的多目标优化问题[J].现代化工,2017,37(11):194-197.
- [2] 陈志旺,白铎,杨七,等.区间多目标优化中决策空间约束、支配及同序解筛选策略[J].自动化学报,2015,41(12):2115-2124.
- [3] WANG Y, WANG X W, CHEN S Y, *et al.* Multi-station multi-robot welding system planning and scheduling based on STNSGA-D: An industrial case study[J]. *IEEE Transactions on Automation Science and Engineering*, 2023: 1-15.
- [4] 付建林,丁国富,张剑,等.基于响应面和 NSGA-II 的 AGV 系统多目标优化配置[J].系统仿真学报,2022,34(5):994.
- [5] DEB K, PRATAP A, AGARWAL S, *et al.* A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [6] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints[J]. *IEEE Transactions on Evolutionary Computation*, 2013, 18(4): 577-601.
- [7] ZITZLER E, KUNZLI S. Indicator-based selection in multiobjective search[C]//International Conference on Parallel Problem Solving from Nature. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004: 832-842.
- [8] ZHANG Q F, LI H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731.
- [9] 李飞,刘建昌,石怀涛,等.基于分解和差分进化的多目标粒子群优化算法[J].控制与决策,2017,32(3):403-410.
- [10] 周艳丽,徐硕,杨辉.基于改进多目标差分进化算法的城轨列车速度曲线优化[J].华东交通大学学报,2023,40(5):76-82.
- [11] 王学武,谢祖洪,周昕,等.基于统计信息反馈的分步多目标优化[J].华东理工大学学报(自然科学版),2022,48(5):665-676.
- [12] 杨雨婷,邱煜炎.针对不规则 Pareto 前沿的高维多目标优化算法应用[J].山西师范大学学报(自然科学版),2023,37(4):15-23.
- [13] 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[J]. *IEEE Transactions on Evolutionary Computation*, 2013, 18(4): 602-622.
- [14] CHEN S Y, WANG X W, WANG Y, *et al.* A modified adaptive switching-based many-objective evolutionary algorithm for distributed heterogeneous flowshop scheduling with lot-streaming[J]. *Swarm and Evolutionary Computation*, 2023, 81: 101353.
- [15] 王学武,魏建斌,周昕,等.一种基于超体积指标的多目标进化算法[J].华东理工大学学报(自然科学版),2020,46(6):780-791.
- [16] 耿焕同,戴中斌,王天雷,等.基于参考点选择策略的改进型 NSGA-III 算法[J].模式识别与人工智能,2020,33(3):191-201.
- [17] 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*, 2017, 22(4): 609-622.
- [18] JIANG S W, ZHANG J, ONG YEW-SOON, *et al.* A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm[J]. *IEEE Transactions on Cybernetics*, 2014, 45(10): 2202-2213.
- [19] 付世炜,苏毅娟,谢承旺. MOEA/IGD-NSE: 一种基于 IGD-NSE 指标的高维多目标进化算法[J].广西科学,2024(1):1-13.
- [20] 叶灵芝,贾立,宋鸣程.基于工况划分的火电机组运行多目标优化[J].自动化仪表,2019,40(5):25-30.
- [21] 江灏,刘墩东,陈静,等.基于 Pareto 多目标优化的光纤 Bragg 光栅传感网络的波长分配[J].光电子激光,2013,24(3):487-492.
- [22] 张英贵,刘家忱,雷定猷.基于多目标进化的超限超重货物运输专列路径优化方法[J].中国铁道科学,2024,45(3):138-148.
- [23] AZZOUZ R, BECHIKH S, BEN S L. Dynamic multi-objective optimization using evolutionary algorithms: A survey[J]. *Recent Advances in Evolutionary Multi-Objective Optimization*, 2017, 20: 31-70.
- [24] TIAN Y, CHENG R, ZHANG X Y, *et al.* PlatEMO: A MATLAB platform for evolutionary multi-objective optimization[J]. *IEEE Computational Intelligence Magazine*, 2017, 12(4): 73-87.

- [25] LI K, DEB K, ZHANG Q F, *et al.* An evolutionary many-objective optimization algorithm based on dominance and decomposition[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 19(5): 694-716.
- [26] SIERRAL M R, COELLO C A. Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance[C]//International Conference on Evolutionary Multi-Criterion Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 505-519.
- [27] SCOTT J R. Fault tolerant design using single and multi-criteria genetic algorithms[J]. Master's Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995, 37(5): 1-13.

## An Adaptive Multi-Objective Optimization Algorithm with Reference Point Based on Grid Congestion Degree

WANG Xuewu, GAO Yongliang, GU Xingsheng

(Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China)

**Abstract:** In multi-objective optimization, balancing both convergence and diversity in the searching population is paramount. However, the AR-MOEA algorithm, which prioritizes optimizing the IGD-NS indicator, and accelerates the convergence process, is prone to trapping in local optima, leading to incomplete coverage of the entire Pareto front by the population. To address this issue, this paper introduces the Adaptive Reference Point Multi-Objective Evolutionary Algorithm based on Grid Crowding (AR-MOEA-GC). This algorithm differentiates the fitness calculation methods for contributing and non-contributing individuals within the population, thereby ensuring both diversity and convergence. Furthermore, to expedite the convergence speed of the population during the later stages of the algorithm, a reference point adjustment strategy is integrated to guide the population to evolve towards the true Pareto front. The enhanced algorithm was rigorously tested against six advanced multi-objective evolutionary algorithms using three types of test functions. The results demonstrate that AR-MOEA-GC exhibits competitive performance in solving three-dimensional multi-objective optimization problems.

**Key words:** evolutionary algorithm; IGD-NS indicator; multi-objective optimization; grid congestion degree; evolutionary computation

(责任编辑: 王晓丽)