

基于邻域竞赛的多目标优化算法

刘元¹ 郑金华^{1,2} 邹娟¹ 喻果¹

摘要 传统多目标优化算法 (Multi-objective evolution algorithms, MOEAs) 的基本框架大致分为两部分: 首先是收敛性保持, 采用 Pareto 支配方法将种群分成若干非支配层; 其次是分布性保持, 在临界层中, 采用分布性保持机制维持种群的分布性. 然而在处理高维优化问题 (Many-objective optimization problems, MOPs) (目标维数大于 3) 时, 随着目标维数的增加, 种群的收敛性和分布性的冲突加剧, Pareto 支配关系比较个体优劣的能力也迅速下降, 此时传统的 MOEA 已不再适用于高维优化问题. 鉴于此, 本文提出了一种基于邻域竞赛的多目标优化算法 (Evolutionary algorithm based on neighborhood competition for multi-objective optimization, NCEA). NCEA 首先将个体的各个目标之和作为个体的收敛性估计; 然后, 计算当前个体向量与收敛性最好的个体向量之间的夹角, 并将其作为当前个体的邻域估计; 最后, 通过邻域竞赛方法将问题划分为若干个相互关联的子问题并逐步优化. 为了验证 NCEA 的有效性, 本文选取 5 个优秀的算法与 NCEA 进行对比实验. 通过对比实验验证, NCEA 具有较强的竞争力, 能同时保持良好的收敛性和分布性.

关键词 多目标优化算法, Pareto 支配关系, 邻域竞赛机制, 高维优化问题

引用格式 刘元, 郑金华, 邹娟, 喻果. 基于邻域竞赛的多目标优化算法. 自动化学报, 2018, 44(7): 1304–1320

DOI 10.16383/j.aas.2017.c160315

An Evolutionary Algorithm Through Neighborhood Competition for Multi-objective Optimization

LIU Yuan¹ ZHENG Jin-Hua^{1,2} ZOU Juan¹ YU Guo¹

Abstract The basic framework of traditional multi-objective evolutionary algorithms (MOEAs) can be classified into two parts: one is the convergence holding of the population, for which the fast nondominated sort approach is used to sort the population into certain nondomination layers; the other is the distribution maintenance of the population, for which diversity maintenance mechanisms are adopted to hold the distribution of the population. However, when dealing with many-objective optimization problems (MOPs) (The number of objective dimensions is greater than 3), with the increase of objective dimensions, the conflicts between convergence and distribution will intensify, and the Pareto dominance's ability of comparing the individuals will decline. In this case, traditional MOEAs are no longer apt. In this paper, a evolutionary algorithm is proposed based on neighborhood competition for multi-objective optimization (denoted as NCEA). Firstly, the convergence of each individual in the population is estimated by summing its objective values; then the angles between current selected solutions and the best converged solution are calculated and taken as the estimates of the distribution of the selected solutions; lastly, an MOP is divided into a number of mutually correlated sub-problems through neighborhood competition and optimizing, respectively. From the comparative experiments with other five representative MOEAs, NCEA is found to be competitive and successful in finding well-converged and well-distributed solution set.

Key words Multi-objective evolutionary algorithm, Pareto dominance, neighborhood competition mechanism, many-objective optimization

Citation Liu Yuan, Zheng Jin-Hua, Zou Juan, Yu Guo. An evolutionary algorithm through neighborhood competition for multi-objective optimization. *Acta Automatica Sinica*, 2018, 44(7): 1304–1320

收稿日期 2016-04-09 录用日期 2017-06-22
Manuscript received April 9, 2016; accepted June 22, 2017
国家自然科学基金 (61502408, 61673331, 61379062, 61403326),
湖南省自教育厅重点项目 (17A212), 赛尔网络创新项目
(NGII20150302), 湖南省科技计划项目 (2016TP1020), 湖南省自然
科学基金 (2017JJ4001, 14JJ2072) 资助

Supported by National Natural Science Foundation of China
(61502408, 61673331, 61379062, 61403326), The Education De-
partment Major Project (17A212), The CERNET Innovation
Project (NGII20150302), The Science and Technology Plan
Project of Hunan Province (2016TP1020), and the National
Natural Science Foundation of Hunan Province (2017JJ4001,
14JJ2072)

本文责任编辑 魏庆来
Recommended by Associate Editor WEI Qing-Lai

近几年, 因为多目标优化算法 (Multi-objective evolution algorithms, MOEAs) 能高效地解决 NP 难问题, 被广泛地运用到各个领域, 包括工程、经济、后勤等. 但大部分多目标进化算法仅在处理 2

1. 湘潭大学信息工程学院智能计算与信息处理教育部重点实验室 湘潭 411105 2. 衡阳师范学院智能信息处理与应用湖南省重点实验室 衡阳 421002

1. Key Laboratory of Intelligent Computing & Information Processing, The college of Information Engineering, Xiangtan University, Xiangtan 411105 2. Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang Normal University, Hengyang 421002

维和 3 维多目标优化问题 (Multi-objective optimization problems, MOPs) 时, 能得到收敛性和分布性均较好的非支配解集. 然而现实问题中, 大部分优化问题都是高维多目标优化问题, 通常需要同时优化多个相互冲突的目标, 目标维数甚至达到 10~15. 因此, 在处理高维多目标优化问题时, 我们需要找到一组均衡收敛性和分布性的解集.

高维多目标优化是指对目标维数大于 3 的 MOPs 进行优化. 对于维数较高的 MOPs, 想要获得一组 Pareto 最优解集非常困难, 一些处理低维数目标时不会遇到的问题将凸显出来, 同时随着目标维数的增加, Pareto 支配关系的优化效果会逐渐减弱^[1]. 故而对于强调非支配解的优化算法, 如果种群中解的选择压力还是如低维目标一样, 那么种群进化速度将会减慢, 甚至停滞. 因此在高维优化问题的研究中, 分布性保持机制将对算法的性能起决定性作用^[2].

然而, 分布性主导的选择机制可能会对算法的收敛性能造成负面影响^[3-4]. 目前大多数分布性保持机制, 如小生境 (Niche)、聚集距离 (Crowding Distance)、聚类 (Clustering) 等^[5], 由于它们过于强调分布性, 种群在优化中可能会偏好支配抵抗解 (Dominance resistant solutions, DRSs)^[6]. 这些 DRSs 具有较好的分布性能, 但由于某一维或者几维上的收敛性远逊色于其他个体, 这些 DRSs 不仅不能增加种群向真实 Pareto 面逼近的选择压力, 反而在某种程度上阻碍了种群的进化搜索, 以致于没有分布性保持机制的算法反而在高维情况下能够表现出更好的收敛性^[7]. 因此, 在处理高维优化问题中, 选择合适的分布性保持机制对基于 Pareto 支配关系的优化算法至关重要.

最近几年, 为了解决在处理高维优化问题时所面临的难题, 研究者们已提出了许多优秀的算法, 根据算法的收敛性和分布性的保持机制的不同, MOEAs 大致可以分为以下 5 类.

1) 宽松 Pareto 支配关系的算法: 此类算法的主要思想是将个体的 Pareto 支配区域进行适当的放大, 再与其他个体进行支配比较. 而这类算法一般都需要一组参数来控制个体的支配需求, 故而参数的设置决定了此类算法性能. 如 ϵ -MOEA^[8]、CDAS^[9]、模糊支配^[10] 等.

2) 基于评价指标的算法: 此类算法借鉴了评价指标的思想, 对评价指标稍加修改, 在个体之间两两比较, 从而帮助选择机制找到更好的个体. 如 IBEA^[11]、SMS-EMOA^[12] 等.

3) 基于聚合的算法: 此类算法通过设计一组均匀的权重系数将高维多目标优化问题转化为单目标优化问题, 再进行逐个优化, 而权重的均匀性决定

了种群的分布性. 如 MSOPS (Multiple single objective pareto sampling)^[13]、MOEA/D^[14]、NSGA (Non-dominated sorting genetic algorithm)-III^[15], 基于目标分解的高维多目标并行进化算法^[16] 等.

4) 基于排列的算法: 此类算法通过定义新的排列方法来区分个体之间的好坏, 使个体间形成全序关系. 因此基于排序的算法不会受到目标空间维数的影响. 如 AR + DMO^[17] 等.

5) 基于密度估计的算法: 此类算法通过改进分布性保持机制来提高收敛性. 其主要思想是为种群中的每个个体设置密度估计信息, 使收敛性差的个体获得更高的密度值, 使分布性保持机制能在维持种群均匀分布的同时提高收敛性. 如 SDE^[3] (计算平移后个体的聚集距离作为个体的密度估计, 值越小, 表示个体越密集), GrEA^[18] (GrEA 中引入了网格聚集距离 GCD 来表示个体的密度估计, 当 GCD 值越大表示个体的密度估计越大) 等.

以上介绍的 5 类算法利用不同的策略提高了算法在高维多目标优化问题上的性能, 同时为解决高维多目标优化问题提供了新的选择和思路. 本文提出一种基于邻域竞赛的多目标算法 (Neighborhood competition-based multi-objective evolutionary algorithms, NCEA) 求解高维多目标优化问题. NCEA 在邻域竞赛机制中同时为个体设计了收敛信息和分布信息, 首先, 根据收敛信息保留收敛性好的个体; 然后, 通过竞赛机制将分布性差的个体淘汰掉. 因此, 在临界层选择过程中, 不会因为分布性保持而破坏种群的收敛能力.

为了检验本文算法的性能, 将算法与其他 5 种经典算法 ϵ -MOEA^[8]、MSOPS^[15]、NSGA-III^[17]、GrEA (Grid-based evolutionary algorithm)^[18]、AR + DMO^[17] 分别在相同条件下优化 DTLZ 系列测试问题^[19] 并进行性能比较. 实验结果表明, 在收敛性和分布性上, NCEA 与其他 5 种算法相比能够获得更好的性能.

1 邻域竞赛机制

邻域竞赛机制综合考虑了种群的收敛信息和分布信息, 收敛信息作用于提高算法的收敛性, 而分布信息反映个体间的位置关系. 在优化过程中, 邻域竞赛机制能将任意多维目标问题转换成多个单目标问题, 首先采用收敛信息选择收敛性较好的个体, 将该个体作为参考点, 再通过分布信息淘汰离参考点近的个体. 如图 1 所示, 通过收敛信息, 选择收敛性最好的个体 p_1 , 阴影部分则为 p_1 的邻域. 位于邻域的两个个体 p_2 和 p_3 由于个体间分布信息的比较可能被淘汰.

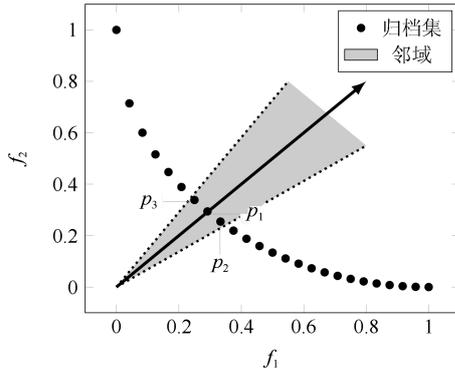


图 1 邻域竞赛机制示意图

Fig. 1 Sketch of neighborhood competition mechanism

1.1 收敛信息

以最小化问题为例, 邻域竞赛机制首先计算当前种群的理想点, 平移当前种群, 使得所有个体广泛的分布在第一象限, 然后通过式 (1) 将各个目标值相加求和 $f_{CI}(p)$, 并将它作为个体的收敛信息. 在分布性保持中, 首先选择收敛信息最小的个体, 然后通过分布信息保证该个体的分布性.

$$f_{CI}(p) = \sum_{j=1}^m f_j(p) \quad (1)$$

式 (1) 中, m 为目标维数, $f_j(p)$ 为第 j 个目标的值. 这个函数主要包含两个方面内容: 目标维数和个体在每一维上的数值大小. 当一个个体在大多数目标上有较好的性能, 那么该个体的收敛信息 $f_{CI}(p)$ 就会相对较小, 反之亦然.

为了证明收敛信息在算法中的有效性, 由于 NCEA 是基于 NSGA-II 框架, 所以本文采用 NCEA 算法和无分布性保持的 NSGA-II^[20] (Pareto 支配关系) 优化 DTLZ2 测试问题^[19], 并用收敛性评价指标 GD^[21] 对算法进行评价. 如图 2 所示, 目标维数的增加对 NCEA 的收敛性基本上没有影响. 在运行 50 代左右, 算法 NCEA 趋于收敛状态. 而 Pareto 支配的收敛能力明显减弱, 在 10 目标下, 算法基本上处于不收敛状态. 因此, 在处理高维问题时, 收敛信息起到了决定性作用, 当 Pareto 支配失效时, 算法主要靠收敛信息引导种群往最优 Pareto 面上收敛.

1.2 分布信息

算法仅有收敛信息往往是不够的, 虽然它能引导种群往最优 Pareto 面收敛, 但是无法保证种群广泛地分布到整个 Pareto 面. 如图 3 所示, 图 3(a) 为凸面问题, 当非支配集是凸面, 通过收敛信息选择个体, 处于中间的个体 p_1 的优先级会比边缘个体高,

将会导致种群往中间聚集; 如果非支配集是凹面, 如图 3(b), 处于边缘的个体的优先级高于中间的个体, 如 p_1 和 p_2 的收敛信息在种群中收敛信息最小, 因此收敛信息将会引导种群往边缘收敛; 而当非支配集是平面时, 如图 3(c), 个体的收敛信息相同, 如果没有分布信息维持种群的分布性, 算法会随机保留 n 个个体 (其中 n 为种群规模), 然而随机选择很难保证种群获得良好的分布性能. 因而, 除了收敛信息, 邻域竞赛机制还须引入分布信息.

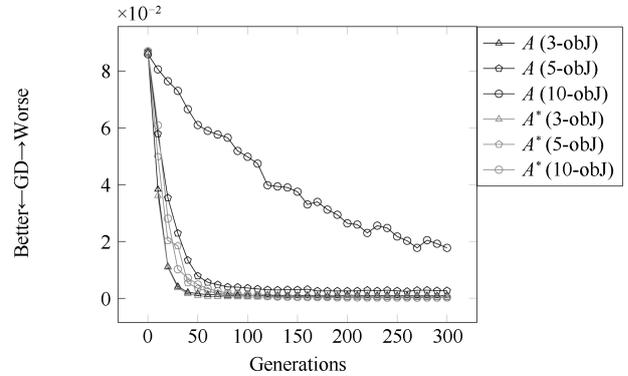


图 2 无分布性保持机制的 NSGA-II (A) 与 NCEA (A*) 的收敛性比较

Fig. 2 Evolutionary trajectories of the average GD for 30 runs of the modified without the diversity maintenance mechanism (denoted as A) and the NCEA (denoted as A*) on DTLZ2

分布信息的主要作用是保证种群拥有好的分布性能. 图 4 给出了分布信息示意图, 在图 4(a) 的非支配集中, 有 $p_1 = (0.3, 0.3)$, $p_2 = (0.5, 0.2)$ 以及 $p_3 = (0.2, 0.5)$ 三个个体. 因为 p_1 离 $y = x$ 这条直线最近, 通过收敛信息首先选择个体 p_1 , 假设 p_2 和 p_3 位于 p_1 的邻域内, 则邻域竞赛机制将分别为 p_2 和 p_3 分配一个分布信息. 本文采用计算两个个体向量夹角 (如式 (2)) 作为个体的分布信息. 如图 4(a) 所示, p_2 的分布信息为 θ_2 , p_3 的分布信息为 θ_3 .

$$\theta = \arccos \frac{op_1 op_2}{|op_1| |op_2|} \quad (2)$$

为了准确地描述种群中候选个体的位置关系, 分布信息需要实时更新. 如图 4(b), 有三个非支配个体, $p_1 = (0.4, 0.1)$, $p_2 = (0.3, 0.45)$ 以及 $p_3 = (0.4, 0.4)$. 假设 p_3 分别属于 p_1 和 p_2 邻域内, 根据收敛信息, 首先选择 p_1 , 则通过如式 (2) 得到 p_3 的分布信息 $\theta_{3(1)}$, 当 p_2 被选择时, 通过计算发现 $\theta_{3(2)}$ 小于 $\theta_{3(1)}$, 我们需要更新 p_3 的分布信息, 将较小的角度赋给 p_3 . 这样, 在竞赛过程中, 分布信息越小的个体越容易被淘汰, 种群的分布性能也会越好.

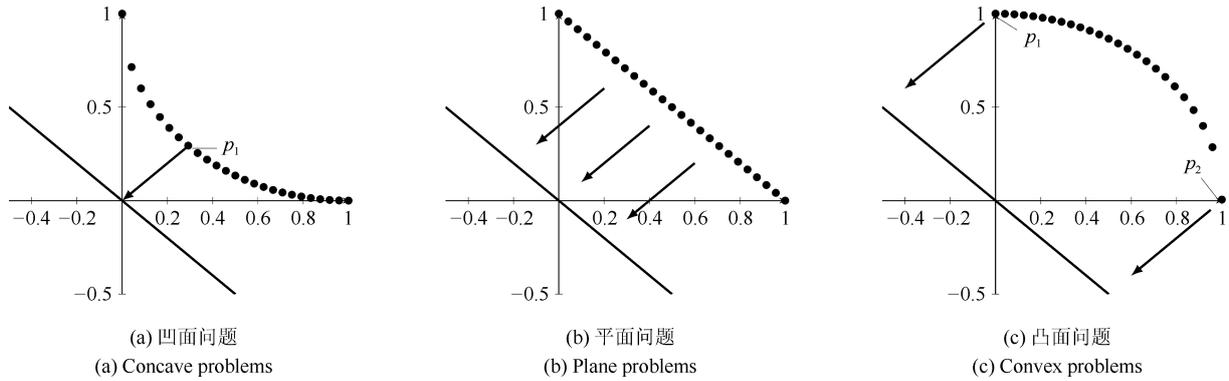


图 3 收敛信息对不同类问题的影响

Fig. 3 Impact of convergence information on different issues

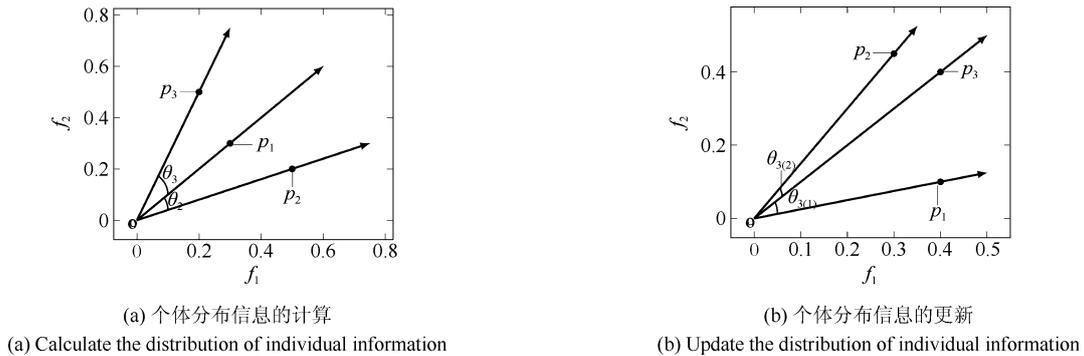


图 4 分布信息示意图

Fig. 4 Distribution diagram

1.3 邻域竞赛机制流程

本节将对邻域竞赛机制的流程进行分析. 该流程主要分为两部分, 首先是选择操作, 选择收敛性好的个体; 然后是竞赛操作, 淘汰分布性差的个体.

算法 1 介绍了邻域竞赛机制流程, 为了保证计算收敛信息的公平性, 需要对当前种群进行平移操作. 首先计算当前种群的理想点, 根据平移公式将当前种群中的各个目标平移, 使得种群位于第一象限, 而理想点作为当前坐标的原点. 然后通过式 (1) 计算个体的收敛信息. 当归档集 A 中的个体数量小于 x (x 为归档集的规模) 时, 将进行选择和竞赛操作.

算法 1. 邻域惩罚机制

Require: 种群: $P = \{p_1, p_2, \dots, p_n\}$, 邻域角度: α , 竞赛集规模: K

Ensure: 归档集: $A = \{p_1, p_2, \dots, p_x\}$ 其中 $x \leq n$

1: 归档集: $A = \phi$, 竞赛集: $E = \phi$;

2: 计算理想点: $z_j^{\min} = \min_{p \in P} f_j(p)$ 其中 $j \in 1, \dots, m$

3: 平移目标: $f'_i = f_j(p) - z_j^{\min}$ 其中 $p \in P, j \in 1, \dots, m$

4: 计算个体的收敛信息: $f_{CI}(p_i) = \sum_{j=1}^m f_j(p_i)$ 其中 $i \in 1, \dots, n, j \in 1, \dots, m$

5: **while** $|A| \leq x$ **do**

6: 选择收敛信息最小的个体: $p_{\text{elite}} = \min_{p \in P} f_{CI}(p)$

7: 将 p_{elite} 移入 A 中: $A = A \cup p_{\text{elite}}, P = P - p_{\text{elite}}$

8: 计算个体的分布信息: $f_{DI}(p_i) = \arccos \frac{op_1 op_2}{|op_1| |op_2|}$, 其中 $p_i \in P$

9: **for** $i = 1$ to $|P|$ **do**

10: **if** $f_{DI}(p_i) \leq \alpha$ **then**

11: **if** $|E| < K$ **then**

12: 将被竞赛个体移入竞赛集中: $E = E \cup p_i,$

$P = P - p_i$

13: **else**

14: 从竞赛集中选取分布信息最大的个体: $e = \max_{e_i \in E} f_{DI}(e_i)$

15: **if** $f_{DI}(e) > f_{DI}(p_i)$ **then**

16: $P = P \cup e, P = P - p_i, E = E - e,$

$E = E \cup p_i$

17: **else**

18: $n = n + 1$

19: **end if**

20: **end if**

21: **else**

22: $i = i + 1$

23: **end if**

24: **end for**

25: end while

在选择过程中, 首先通过收敛信息选择收敛性最好的个体 p_{elite} 并移入归档集. 然后通过式 (2) 计算种群中候选个体 p_i 的分布信息, 其中 $p_i \in P$. 最后判断种群 P 中的个体是否位于 p_{elite} 的邻域内. 如果不在, 则不进行竞赛操作; 反之, 则通过分布信息淘汰分布性差的个体.

这里, 我们需要引进一个竞赛集 E , 它的容量为 $s = n - x$, n 为种群规模, x 为归档集规模. 当竞赛集内的个体数目小于 s 时, 将 p_{elite} 邻域内个体全部移入竞赛集中, 而当竞赛集被填满时, 采用交换操作, 把竞赛集中分布信息最大的个体 e 与当前被竞赛的个体 p_i 进行比较, 如果 $f_{DI}(e) > f_{DI}(p_i)$, 则进行位置替换, 把个体 e 移入种群 P 中, 将个体 p_i 移入 E 中. 重复上述操作, 分布性差的个体都将落入竞赛集中, 而具有良好的分布性和收敛性的个体将被保留到下一代.

2 算法流程与时间复杂度

本节将详细分析 NCEA 的算法流程以及时间复杂度, 并选取了 5 个经典的 MOEAs (ϵ -MOEA、MSOPS、NSGA-III、GrEA、AR + DMO) 进行比较, 表 1 列出了各个算法的时间复杂度. 为了方便描述, 我们假设种群规模为 n , 目标维数为 m .

与传统的 MOEA 一样, NCEA 需要随机初始化种群, 在每一代运行过程中首先进行交叉和变异操作产生同样规模的子代种群, 并合并父代和子代种群构成一个规模为 $2n$ 的种群. 然后进行环境选择, 而环境选择又分为两部分, 首先是 Pareto 排序, 然后是临界层选择. 其算法框架如算法 2.

算法 2. 算法框架

Require: 种群规模: n , 目标数: m , 运行代数: g_{\max}

Ensure: 种群: $P_t + 1$

1: 随机初始化种群: $P_t = \{p_1, p_2, \dots, p_n\}, S_t = \phi, i = i + 1$

2: for $g = 1$ to g_{\max} do

3: 交叉变异操作: $Q_t = \text{Recombination} + \text{Mutation}(P_t)$

4: 合并操作: $R_t = P_t \cup Q_t$

5: 非支配排序操作: $\{F_1, F_2, \dots\} = \text{NondominateSort}(R_t)$

6: repeat

7: 优先级高的非支配层优先被选择: $S_t = S_t \cup F_i, i = i + 1$

8: until $|S_t| \geq n$

9: 临界层: $F_l = F_i$

10: if $|S_t| = n$ then

11: 将 S_t 中个体保留到下一代: $P_{t+1} = S_t$

12: break

13: else

14: 临界层选择: $P_{t+1} = S_t / F_l \cup \text{CriticalSelection}(F_l, \alpha, K)$

15: end if

16: end for

在算法 2 中, Recombination + Mutation 为交叉变异操作; 而 NondominateSort 操作本文选择了擂台赛法^[22] 对种群进行非支配排序; CriticalSelection 操作则为本文的邻域竞赛算法. NCEA 的时间消耗主要在环境选择中, 而环境选择的计算消耗主要取决于非支配排序和邻域竞赛机制. 非支配排序的时间复杂度为 $O(n \log^{m-2} n)$ ^[23]. 对于邻域竞赛机制, 计算理想点的时间复杂度为 $O(mn)$; 平移坐标操作的时间复杂度为 $O(mn)$; 而计算个体的收敛信息的时间复杂度为 $O(mn)$; 一次个体分布信息的计算的时间复杂度为 $O(mn)$, 总共为 $O(mn^2)$; 选择一个精英个体的时间复杂度为 $O(n)$; 一次竞赛操作的时间复杂度为 $O(n)$, 进行 n 次选择, 则有 n 次竞赛操作, 总共时间复杂度为 $O(n^2)$. 从而, 环境选择的时间复杂度为 $O(mn^2)$. 所以, NCEA 迭代一次的时间复杂度为 $O(mn^2)$.

由于 ϵ -MOEA 采用稳态进化模式, 每运行一代只产生一个个体, 故而此处以 n 次迭代为一代 (n 为种群规模). ϵ -MOEA 时间复杂度计算主要包含对种群和归档集的维护与修剪, 因此它的总体时间复杂度为 $O(mn(n+k))$, 其中 k 为归档集的规模. MSOPS 的时间复杂度主要是种群中个体在各个权重上的排序操作, 算法的时间复杂度为 $O(mnt \cdot \log(t))$ ($n > t$) 或 $O(mtn \cdot \log(n))$ ($n < t$), 其中 t 为权重规模. NSGA-III 的框架主要由非支配排序和环境选择两部分构成. 在最坏的情况下 (第一层非支配解集规模超过种群规模), 该算法迭代一次的时间复杂度为 $O(n^2 \log^{m-2} n)$ 或 $O(mn^2)$. GrEA 的时间开销主要在以下 3 个部分中: 网格构造, 适应度分配以及环境选择. 网格构造和适应度分配的时间复杂度为 $O(mn)$, 而环境选择的时间复杂度为 $O(mn^2)$. 因此, GrEA 迭代一次的时间复杂度为 $O(mn^2)$.

表 1 6 个算法的时间复杂度

Table 1 The time complexity of six algorithms

算法	NCEA	ϵ -MOEA	GrEA	AR + DMO	MSOPS	NSGA-III
时间复杂度	$O(mn^2)$	$O(mn(n+k))$	$O(mn^2)$	$O(mn^2)$	$O(mnt \cdot \log(t))$ ($n > t$)	$O(mn^2)$

时间复杂度为 $O(mn^2)$. AR + DMO 同样被分为 3 个部分: AR 排序, DMO 对种群收敛性评价, 分布性保持机制. 其中分布性保持机制采用的是 NSGA-II 的聚集距离策略. 这三个部分的时间复杂度都为 $O(mn^2)$, 因此 AR + DMO 运行一代的时间复杂度为 $O(mn^2)$.

为了更直观地反应各个算法在时间上的开销, 图 5 给出了算法参数 (种群大小和目标维数) 与时间复杂度的关系图. 图中的三个不同曲线代表了不同的时间复杂度的变化趋势, 其中下方曲线代表时间复杂度为 $O(mn^2)$ 的算法 (NCEA、NSGA-III、GrEA、AR + DMO) 的变化趋势; 中间曲线代表时间复杂度为 $O(mn(n+k))$ 的算法 (ϵ -MOEA) 的变化趋势, 为了方面描述, 这里假设 ϵ -MOEA 的归档集大小 $k = n$, 那么其时间复杂度就可以描述成 $O(2mn^2)$, 而上方曲线代表时间复杂度为 $O(mnt \cdot \log(t))$ ($n > t$) 的算法 (MSOPS) 的变化趋势, 这里假设权重数目 t 小于种群大小 n , 且 $t = n/2$. 从图 5 中的曲线变化来说, 各个算法的时间复杂度都与目标维数和种群大小呈正相关, 随着目标维数的增加或种群大小的增加而上升. 但是上升的趋势明显不同, 当种群大小和目标维数达到一定值时, MSOPS 的时间复杂度远高于其他几个算法. ϵ -MOEA 由于多了归档集操作, 所以其时间复杂度比 $O(mn^2)$ 稍高一些.

根据以上分析, 我们可以在时间复杂度上为 6 个算法进行排序: $NCEA = NSGA-III = GrEA = AR + DMO < \epsilon$ -MOEA < MSOPS.

3 实验分析

为了检验 NCEA 的性能, 本文选取了 5 个算法进行比较, 其中包括基于宽松 Pareto 支配关系的算法: ϵ -MOEA, 基于集聚合的算法: MSOPS 和 NSGA-III, 基于排列的算法: AR + DMO, 以及基于密度估计算法: GrEA, 而对于基于指标的算法, 由于需要对种群中的个体进行两两比较. 因此, 相对

其他类算法运行时间过高, 同时其性能也不够理想, 所以本文没有选择基于指标的算法进行比较. 在测试问题上, 本文采用 DTLZ 系列测试函数^[19]. 评价方法选择了收敛性评价指标 GD^[21], 分布性评价指标 DM^[24] 以及综合评价指标 IGD^[25].

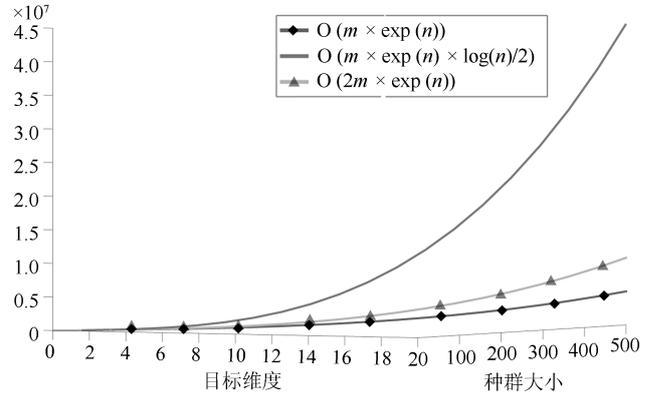


图 5 参数与时间复杂度的关系
Fig. 5 The relationship between parameters and time complexity

3.1 测试问题

DTLZ 系列测试函数^[19] 是由 Kalyanmoy 在 2005 年提出的. 它的目标个数可以任意设置, 被广泛地用来检验算法的性能. DTLZ 测试问题包含多种特性, 包括: 线性问题、凸面问题、凹面问题、多峰问题、连续问题、非连续问题以及退化问题等. 表 2 给出了 DTLZ 系列测试问题的参数设置及问题特性.

3.2 评价指标

本小节将介绍三种评价指标: Generational distance (GD)、Diversity metric (DM) 和 Inverted generational distance (IGD). 其中 GD 是用于检验算法在优化过程中种群收敛的能力, DM 反映评价种群的分布性能, 而 IGD 是综合评价指标, 同时反映了种群的收敛性和分布性.

表 2 DTLZ 系列测试问题及相关算法参数设置
Table 2 DTLZ series test and related algorithm parameters

问题	目标个数	特性	ϵ 参数	GrEA 参数
DTLZ1	3, 4, 5, 6, 8, 10	Linear, multimodal	0.033, 0.052, 0.059, 0.0554, 0.0549, 0.0565	10, 10, 10, 10, 10, 11
DTLZ2	3, 4, 5, 6, 8, 10	Concave	0.06, 0.1312, 0.1927, 0.234, 0.29, 0.308	10, 10, 9, 8, 7, 8
DTLZ3	3, 4, 5, 6, 8, 10	Concave, multimodal	0.06, 0.1385, 0.2, 0.227, 0.1567, 0.85	11, 11, 11, 11, 10, 11
DTLZ4	3, 4, 5, 6, 8, 10	Concave, biased	0.06, 0.1312, 0.1927, 0.234, 0.29, 0.308	10, 10, 9, 8, 7, 8
DTLZ5	3, 4, 5, 6, 8, 10	Concave, degenerate	0.0052, 0.042, 0.0785, 0.11, 0.1272, 1.15, 1.45	35, 35, 29, 14, 11, 11
DTLZ6	3, 4, 5, 6, 8, 10	Concave, degenerate, biased	0.0227, 0.12, 0.3552, 0.75, 1.15, 1.45	36, 36, 24, 50, 50, 50
DTLZ7	3, 4, 5, 6, 8, 10	Mixed, disconnected, biased	0.048, 0.105, 0.158, 0.15, 0.225, 0.46	9, 9, 8, 6, 5, 4

GD 通过式 (3) 计算种群中的个体到最优 Pareto 面上最近点的距离, 当 GD 值越小表明算法的收敛性能越好, 在理想情况下, GD 值等于 0.

$$GD_p(S, P^*) = \frac{\sum_{i=1}^{|S|} d_i(a_i, P^*)}{|S|} \quad (3)$$

这里, d_i 表示 S 中的个体 a_i 到最优 Pareto 面上的最小距离, 如式 (4) 所示.

$$d_i(a_i, P^*) = \min\{\|a_i - p\|_2 : a_i \in S\} \quad (4)$$

DM 评价种群的分布均匀性, 它的取值范围为 (0, 1], DM 值越接近 1, 说明种群的分布越均匀. DM 需要参数, 其参数设置如表 3.

IGD 为 GD 的逆向映射, 即 $IGD = GD(P^*, S)$. 其中 S 为当前种群, P^* 为最优解集. 与 GD 相同, 当 IGD 值越小, 说明算法的综合性能越好.

表 3 Metric 的网格划分数设置

Table 3 Settings of division for diversity metric

目标数	3	4	5	6	8	10
网格划分数	10	6	4	3	3	3

3.3 实验参数设置

在本文中, 为了保证算法的公平性, 都采用实数编码, 交叉算子采用模拟二进制交叉, 其中 $\eta_c = 20$, 交叉率 $p_c = 1$; 变异算子采用多项式变异 $\eta_m = 20$, 变异率 $p_m = 1/n$, 其中 n 为决策空间的维数. 对每个算法每次实验独立重复进行 30 次, 种群规模均为 100. 目标空间维数如表 2, 分别为 3, 4, 5, 6, 8, 10. 根据每个问题的难易程度, 我们需要为不同问题设置不同的运行代数, 本文参照了文献 [8] 和文献 [26] 等中的设置, 如表 4 所示.

表 4 终止条件, 以代数为单位

Table 4 Terminate condition, in generation

问题	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
运行代数	1000	300	1000	300	300	1000	300

由文献 [8] 可知, ϵ -MOEA 需要为不同的问题设置合适的 ϵ 参数, 为了使 ϵ -MOEA 得到更好的分布性, 本文采用表 2 中的数据, 在 GrEA 需要给出每一维坐标的分割数, 参照表 2 中的 GrEA 参数, 以上设置是根据算法在各个测试问题上独立重复 30 次实验获得的最好结果时所对应的参数.

在 NCEA 中, 同样为了在不同问题上能取得最好的效果, 需要根据不同问题的特性为个体设置不

同的邻域范围, 这里用 α 表示. 为了分析不同 α 对 NCEA 性能的影响, 本文在一定范围内将 α 等分成 100 份, 并统计 NCEA 在不同 α 参数下的 GD 值, DM 值和 IGD 值变化规律, 其中实验独立重复 30 次. 如图 6, 根据不同问题的特性, 本文给出了线性问题 DTLZ1, 球面问题 DTLZ2, 退化问题 DTLZ5 以及非连续问题 DTLZ7 在 3 维目标和 6 维目标上面的结果.

从图 6 可知, 除了 DTLZ1 的 GD 值波动较大, 其他数据基本上保持平稳. 并且当 α 等于 0 时, 在不同问题上 GD 值最小, 但是 DM 值也是最小, IGD 值最大. 这说明算法的收敛性较好, 而分布性和综合性能都不理想, 证明了邻域的引入是有必要的. 而当 α 大于 0 时, 算法的收敛能力稍微减弱, 但是分布性和综合指标会明显好转, 并且都保持在一个平稳的状态. 因此为了平衡算法的收敛性和分布性, 针对不同的优化问题, 本文给出了一组参数 α 来控制个体的邻域范围, 如表 5 所示.

表 5 NCEA 的参数设置

Table 5 Settings of α parameter for NCEA, in degree

目标数	问题						
	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
3	0.128	0.192	0.128	0.256	0.128	0.192	0.064
4	0.256	0.320	0.256	0.512	0.064	0.064	0.128
5	0.385	0.385	0.385	0.385	0.064	0.064	0.064
6	0.385	0.769	0.513	0.833	0.064	0.064	0.128
8	0.385	0.577	0.577	0.577	0.577	0.064	0.192
10	0.513	0.769	0.641	0.641	0.641	0.064	0.256

3.4 实验结果

本节给出了 6 个算法在 DTLZ 系列测试问题上的实验结果, 表 6 中列出了所有实验的均值和方差 (括号内为方差). 其中表现最好的结果用粗体标识, 第二好的结果用下划线标识. 本文还对样本数据进行了显著性比较, 其中方差分析的显著性水平 β 设置为 0.05. 当 Sig 越大, 表明算法间的显著性差异越大. † 表示 Sig 大于 0.05 时的情况.

本节将从两个角度对 6 种算法的性能进行分析: 首先从单个角度, 将算法的收敛性 (GD) 和分布性 (DM) 单独拿出来进行评价; 然后从整体角度, 将收敛性和分布性结合起来进行综合分评价 (IGD). 这样, 通过三个思想完全不同的指标来反映算法的性能, 使得实验结果更加严谨, 更具有说服力.

表 6 给出了 6 种算法在不同维数上的 DTLZ 系列测试问题中的 GD 值. 如表所示, 被粗体和下划线标记最多的算法是 NCEA (粗体标记的有 25 个, 下划线标记的有 9 个, 占总测试问题的 80%), 其次是 ϵ -MOEA、NCEA 特别在 DTLZ1、DTLZ3

和 DTLZ6 这类难收敛的问题上都有较好的收敛性,但是在 DTLZ2、DTLZ4 以及 DTLZ7 的低维情况下表现欠佳,不如 GrEA. 而 ϵ -MOEA 仅在 DTLZ1 和 DTLZ7 问题上表现较好,在其他问题上收敛性远没有 NCEA 算法好. 对于 GrEA, 在 3 维和 4 维问题上能获得较好的收敛性,在优化 DTLZ6 问题时,收敛性能与 NCEA 相当,比其他 4 种算法好. MSOPS 在 DTLZ2 和 DTLZ4 上拥有较好的收敛性,在其他问题上收敛效果表现不佳. 最后, NSGA-III 在收敛性上表现不佳,其效果远差于 NCEA,特别地,在维数较高的情况下, NSGA-III 完全不收敛. 因此,根据 GD 值对 6 种算法的收敛能力进行排序有: NCEA、 ϵ -MOEA、GrEA、MSOPS、AR + DMO、NSGA-III.

表 7 反映的是算法的分布性能. 根据粗体标记,明显可知, NCEA 整体的分布性最好(粗体标记的有 20 个,下划线标记的有 11 个,占总测试问题的 73%),其次是 ϵ -MOEA,但是 ϵ -MOEA 在处理 DTLZ4 问题上分布性能不太理想. 虽然 NSGA-III 的收敛性表现不佳,但是在处理 DTLZ1、DTLZ2 以及 DTLZ4 问题上具有较好的分布性能. 除了 DTLZ2 和 DTLZ3 问题外, GrEA 的分布性能也较为理想. 6 种算法中,分布性最差的要属 AR + DMO,从图 7 和图 8 可以看出,种群被收敛到很小

的区域内. 值得一提的是, MSOPS 虽然拥有均匀权重,但是它的权重数量只是种群的一半,存在个体与个体重叠的情况.

以上两个指标将分布性和收敛性进行了单独的分析,从单方面来考虑算法的性能. 对于 GD 指标,通过式 (3) 可以知道,它能够充分地反映出算法的收敛性能,但是收敛性好不代表算法能够获得好的分布性. 而 DM 作为分布性指标,虽然它采用了基于网格划分的方法来分析算法的分布性,但是它也存在一些缺陷,导致在对算法分布性能分析上产生误差,例如: 1) DM 需要一组均匀覆盖整个 Pareto 面的参考集,同时保证这些参考集的数目与种群大小相等,从而保证理想状态下的分析性能 DM 等于 1,如何确定一组这样的参考集对高维优化问题非常困难. 2) 随着优化问题维度的不断增加,DM 描述算法的分布性能时产生的误差也会越来越大,主要原因是 DM 采用的是基于空间划分网格思想,对于某网格中的个体,它的邻域是通过计算个体的网格坐标间的曼哈顿距离来确定的,而不是通过计算个体间的欧氏距离确定. 因此,在确定邻域个体时会产生无法估计的误差,从而影响算法性能的评估.

为了更严谨地分析算法的性能,使得算法的比较更具说服力,本文还做了综合性指标分析 (IGD),将收敛性和分析性综合考虑分析算法. 如表 8 所

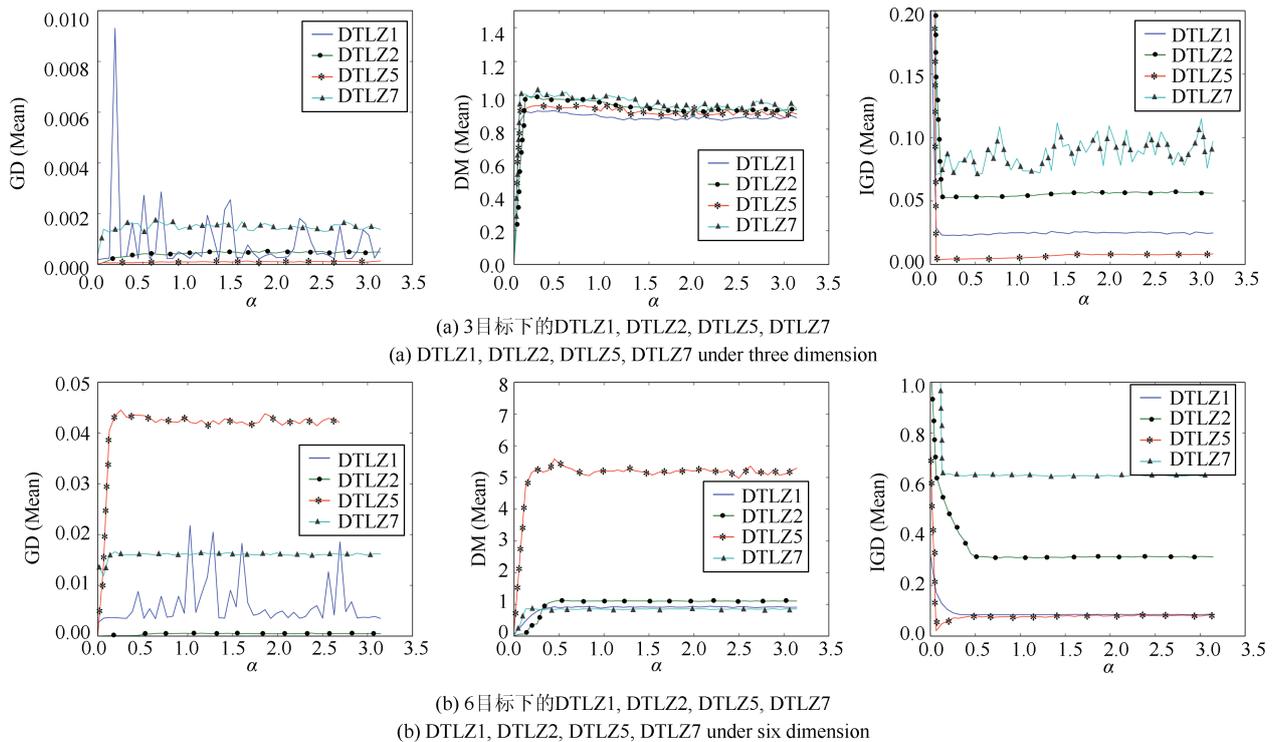


图 6 在 $(0, \pi]$ 内, 将 α 等分成 100 份, 分析 NCEA 在不同指标下的均值

Fig. 6 Splitting the parameter α into 100 parts in certain ranges, and analyzing the different indicators' mean values NCEA

表 6 收敛性指标 GD 的统计数据 (均值和方差)
Table 6 Statistical results of the convergence indicator GD (mean and SD)

问题	目标数	均值与方差					
		NCEA	e-MOEA	ε-GrEA	AR + DMO	MSOPS	NSGA-III
DTLZ1	3	<u>6.5173E-04 (2.16738E-03)</u>	2.4240E-04 (3.24306E-05)	1.7059E-02 (8.29783E-02) [†]	1.6188E-02 (4.54827E-02)	6.9466E-03 (3.67556E-02)	8.3968E-02 (3.03360E-01)
	4	1.3333E-03 (5.87960E-04)	<u>1.5342E-03 (1.01728E-04)</u>	5.0108E-02 (1.48793E-01)	7.1744E-03 (1.27516E-02)	8.0661E-03 (3.25514E-02)	2.7705E-02 (7.46720E-02)
	5	2.3973E-03 (3.29058E-04)	<u>2.8019E-03 (5.10129E-04)</u>	6.5782E-02 (3.16118E-01)	5.9357E-02 (1.54993E-01)	2.8872E-02 (9.41475E-02)	5.3457E-02 (1.27134E-01) [†]
	6	3.7016E-03 (8.14840E-04)	3.5723E-03 (4.48798E-04)	4.1469E-02 (1.31608E-01)	5.1421E-02 (1.02472E-01)	3.8983E-02 (1.02165E-01) [†]	1.6179E-01 (2.49415E-01) [†]
	8	5.5829E-03 (1.27751E-04)	<u>6.1082E-03 (9.60869E-04)</u>	8.6450E-02 (3.30601E-01) [†]	3.6173E-02 (9.55063E-02) [†]	9.9818E-02 (1.48950E-01) [†]	9.0905E-01 (1.19337E+00) [†]
	10	8.2959E-03 (6.14113E-03)	<u>3.4608E-02 (3.75171E-02)[†]</u>	4.1050E-02 (2.78262E-02) [†]	7.8265E-02 (1.99814E-01)	1.2987E-01 (1.80125E-01) [†]	2.0312E-01 (4.43355E-01) [†]
	3	2.4938E-04 (7.71848E-05)	7.5429E-04 (5.67439E-05) [†]	4.4901E-05 (4.52089E-05)[†]	4.9012E-04 (1.50824E-04) [†]	<u>1.1257E-04 (1.33222E-04)[†]</u>	3.0712E-04 (2.34867E-04)
	4	3.4325E-04 (1.07068E-04)	2.1259E-03 (1.25929E-04) [†]	<u>2.4815E-04 (3.10381E-04)</u>	1.1270E-03 (3.29167E-04) [†]	2.0637E-04 (1.21438E-04)[†]	7.0224E-04 (1.24904E-04) [†]
	5	2.1616E-04 (4.46158E-05)	4.1994E-03 (6.61445E-04) [†]	4.6204E-04 (1.75780E-04) [†]	4.1831E-03 (1.24812E-03) [†]	<u>3.7035E-04 (2.29673E-04)[†]</u>	1.9392E-03 (2.82516E-04) [†]
	6	<u>5.7860E-04 (1.84086E-04)</u>	5.6277E-03 (1.97491E-03) [†]	6.3318E-04 (1.86383E-04)	9.1966E-03 (2.34274E-03) [†]	5.2284E-04 (1.85337E-04)[†]	4.3691E-03 (6.82254E-04) [†]
8	2.8758E-04 (1.06852E-04)	6.8790E-03 (8.32033E-04) [†]	2.2182E-03 (8.86939E-04) [†]	1.9660E-02 (3.92533E-03) [†]	<u>1.0396E-03 (2.79993E-04)[†]</u>	1.1352E-02 (3.21989E-03) [†]	
10	3.4444E-04 (1.40595E-04)	5.5698E-03 (4.47660E-04) [†]	<u>1.7998E-03 (3.52157E-04)[†]</u>	3.1090E-02 (3.81003E-03) [†]	1.6458E-03 (3.34714E-04) [†]	4.4401E-03 (2.74564E-03) [†]	
DTLZ3	3	<u>2.8456E-04 (2.41286E-04)</u>	13291E-03 (4.28045E-04) [†]	1.3041E-01 (5.43435E-01) [†]	4.0661E-03 (8.97266E-03) [†]	1.5370E-04 (1.11865E-04) [†]	8.0032E-03 (3.01346E-02)
	4	3.9361E-04 (2.52161E-04)	<u>4.8620E-03 (2.40060E-03)[†]</u>	9.6669E-02 (4.69495E-01)	1.2016E-01 (4.21198E-01)	6.6798E-03 (2.22771E-02)	6.3762E-02 (1.68358E-01) [†]
	5	5.6584E-04 (3.77953E-04)	<u>8.9207E-03 (4.14879E-03)</u>	1.5762E+00 (2.65502E+00)	1.7752E-02 (4.77234E-02) [†]	9.4459E-02 (3.57033E-01) [†]	6.0880E-01 (1.38366E+00)
	6	5.6684E-04 (3.12144E-04)	<u>1.7783E-02 (1.10904E-02)[†]</u>	2.9341E+00 (4.02971E+00) [†]	8.4404E-02 (2.22184E-01) [†]	2.5392E-01 (6.65749E-01) [†]	2.7222E+00 (2.01272E+00) [†]
	8	6.8971E-04 (3.84881E-04)	1.5738E+00 (2.37967E+00) [†]	2.1981E+00 (2.32611E+00) [†]	<u>1.9263E-01 (6.19185E-01)</u>	1.3256E+00 (1.21451E+00) [†]	1.8191E+01 (5.70052E+00) [†]
	10	7.6931E-04 (4.06091E-04)	3.1649E+00 (2.87679E+00) [†]	2.4362E-01 (6.95779E-01)	<u>1.4932E-01 (2.98389E-01)[†]</u>	1.4714E+00 (9.55260E-01) [†]	1.7709E+01 (1.09209E+01) [†]
	3	2.1106E-04 (1.30383E-04)	9.8535E-04 (4.16771E-04) [†]	<u>1.2073E-04 (2.58715E-04)</u>	2.5681E-04 (2.91554E-04)	6.5585E-05 (1.65279E-04)[†]	2.4774E-04 (1.30491E-04)
	4	5.7399E-04 (1.77486E-04)	2.4360E-03 (5.77431E-04) [†]	<u>2.0198E-04 (3.13694E-04)[†]</u>	1.5170E-03 (3.12700E-03)	1.7653E-04 (9.46011E-05)[†]	6.7933E-04 (2.68225E-04)
	5	2.8600E-04 (1.56925E-04)	5.3361E-03 (1.76792E-03) [†]	4.3361E-04 (1.76189E-04) [†]	2.4495E-03 (2.17001E-03) [†]	<u>3.5318E-04 (9.82127E-05)</u>	1.6399E-03 (3.96018E-04) [†]
	6	5.9198E-04 (2.68476E-04)	1.0150E-02 (8.10313E-03) [†]	8.3766E-04 (3.33206E-04) [†]	4.8072E-03 (3.04760E-03) [†]	<u>8.3145E-04 (5.98874E-04)[†]</u>	3.4386E-03 (1.12700E-03) [†]
8	4.3000E-04 (2.03320E-04)	1.0441E-02 (5.13729E-03) [†]	2.4109E-03 (1.05389E-03) [†]	1.4767E-02 (2.59938E-03) [†]	<u>1.5980E-03 (5.54551E-04)[†]</u>	5.2812E-03 (4.42282E-03) [†]	
10	3.3791E-04 (1.63041E-04)	1.5882E-02 (1.24883E-02) [†]	1.6270E-03 (2.48190E-04) [†]	2.8114E-02 (4.28606E-03) [†]	<u>2.8111E-03 (9.12415E-04)[†]</u>	1.5193E-02 (5.35355E-03) [†]	

DTLZ5	3	8.2379E-05 (4.28364E-05)	6.0527E-05 (6.42860E-06) [†]	5.9233E-05 (5.66109E-05)	8.3765E-04 (1.13766E-03) [†]	1.0821E-01 (2.80056E-03) [†]	2.0193E-04 (4.83885E-05) [†]
	4	3.4898E-02 (2.52789E-03)	5.0231E-02 (3.20057E-03) [†]	1.9988E-03 (1.06200E-03) [†]	1.6314E-02 (7.87450E-03) [†]	1.5362E-01 (3.29054E-03) [†]	3.3321E-02 (1.53968E-02)
	5	1.7277E-02 (9.20726E-04)	5.1506E-02 (1.82190E-03) [†]	1.8679E-02 (1.93218E-02)	2.4014E-02 (6.34334E-03) [†]	1.8936E-01 (3.08728E-03) [†]	5.4787E-02 (9.97725E-03) [†]
	6	1.4070E-02 (6.11126E-04)	5.7760E-02 (6.23864E-03) [†]	5.6970E-02 (3.73031E-03) [†]	3.4634E-02 (8.44098E-03) [†]	2.0364E-01 (2.87587E-03) [†]	7.0914E-02 (1.34050E-02) [†]
	8	4.8683E-02 (4.12569E-03)	5.3952E-02 (4.32586E-03) [†]	1.0139E-01 (5.95976E-03) [†]	1.3747E-01 (3.12555E-02) [†]	2.2976E-01 (2.27175E-03) [†]	1.0419E-01 (1.477761E-02) [†]
	10	5.4205E-02 (5.08366E-03)	6.0848E-02 (6.67483E-03) [†]	1.1183E-01 (7.60976E-03) [†]	1.7080E-01 (2.86372E-02) [†]	2.3387E-01 (1.92825E-03) [†]	1.5273E-01 (1.37459E-02) [†]
	3	3.1665E-03 (2.76698E-03)	5.2588E-03 (4.90865E-04) [†]	3.3264E-03 (1.61298E-03)	5.1275E-03 (3.55167E-03) [†]	2.0425E-01 (4.49689E-03) [†]	9.8696E-03 (7.13856E-03) [†]
	4	4.2131E-02 (5.62475E-03)	1.1064E-01 (9.59272E-03) [†]	2.5454E-02 (9.72852E-03) [†]	1.4171E-01 (2.38262E-02) [†]	3.3404E-01 (7.50779E-03) [†]	1.6955E-01 (2.03052E-02) [†]
	5	1.5250E-02 (1.92197E-03)	1.5048E-01 (5.64830E-03) [†]	9.3128E-02 (5.99215E-02) [†]	2.9541E-01 (2.07511E-02) [†]	4.9366E-01 (1.37175E-02) [†]	6.8704E-01 (2.20882E-02) [†]
	6	1.3164E-02 (1.52962E-03)	2.5553E-01 (2.02566E-02) [†]	3.3090E-02 (1.30923E-01) [†]	3.3130E-01 (3.18902E-02) [†]	5.5517E-01 (1.51144E-02) [†]	8.7899E-01 (1.26348E-03) [†]
8	1.3972E-02 (1.16462E-03)	2.9568E-01 (1.46263E-01) [†]	1.3262E-01 (2.81758E-01) [†]	6.3637E-01 (3.31022E-02) [†]	7.6863E-01 (1.65607E-02) [†]	8.8443E-01 (6.83775E-02) [†]	
10	1.6102E-02 (9.86460E-04)	3.4632E-01 (2.74054E-01) [†]	4.0319E-01 (3.82174E-01) [†]	7.4899E-01 (5.44463E-02) [†]	8.1216E-01 (1.81888E-02) [†]	7.6559E-01 (3.71012E-02) [†]	
DTLZ6	3	1.4269E-03 (6.27153E-04)	6.9496E-04 (3.51603E-05) [†]	9.5260E-04 (7.14402E-04) [†]	2.2605E-03 (1.09999E-03) [†]	4.4358E-03 (6.88598E-04) [†]	2.6307E-03 (9.13444E-04) [†]
	4	5.6080E-03 (4.49695E-04)	2.3024E-03 (5.5552E-04) [†]	4.7689E-03 (2.24499E-04) [†]	1.4251E-02 (5.16088E-03) [†]	6.9670E-03 (1.08085E-03) [†]	5.4934E-03 (1.15755E-03)
	5	7.6494E-03 (3.39809E-04)	3.3534E-03 (1.06828E-03) [†]	1.0717E-02 (6.69624E-04) [†]	8.3798E-02 (2.79588E-02) [†]	1.0825E-02 (3.33610E-03) [†]	2.0268E-02 (5.51018E-03) [†]
	6	1.6013E-02 (6.62155E-04)	4.6055E-03 (1.70038E-03) [†]	1.1947E-02 (4.76109E-04) [†]	1.7956E-01 (4.96227E-02) [†]	2.4724E-02 (1.41937E-02) [†]	9.1749E-02 (3.63369E-02) [†]
	8	2.1716E-02 (3.75157E-04)	1.4274E-02 (1.09412E-02) [†]	2.6989E-02 (1.56032E-03) [†]	3.3977E-01 (7.68451E-02) [†]	2.0888E-01 (1.13986E-01) [†]	9.5229E-01 (2.31546E-01) [†]
	10	4.8018E-02 (2.56958E-03)	1.5505E-02 (1.25040E-02) [†]	4.9859E-02 (2.27362E-03) [†]	6.6739E-01 (1.39620E-01) [†]	6.5667E-01 (2.37870E-01) [†]	2.6579E+00 (4.98006E-01) [†]

表 7 收敛性指标 DM 的统计数据 (均值和方差)

问题	目标数	均值与方差					
		NCEA	ε-MOEA	ε-GrEA	AR + DMO	MSOPS	NSGA-III
DTLZ1	3	9.0819E-01 (2.30021E-02)	1.0026E+00 (9.73518E-03) [†]	5.9153E-01 (1.56974E-01) [†]	4.8806E-01 (1.41348E-01) [†]	7.2868E-01 (8.60635E-03) [†]	6.5816E-01 (1.22945E-01) [†]
	4	9.3824E-01 (3.92344E-02)	9.0282E-01 (9.30006E-02)	8.0118E-01 (1.90615E-01) [†]	2.7235E-01 (1.05381E-01) [†]	7.5392E-01 (1.49390E-02) [†]	1.0242E+00 (2.41363E-01) [†]
	5	8.9325E-01 (5.14120E-02)	7.3063E-01 (1.26330E-01) [†]	7.7569E-01 (1.75880E-01) [†]	2.4440E-01 (8.58044E-02) [†]	8.5011E-01 (1.34821E-02) [†]	1.1297E+00 (2.79471E-01) [†]
	6	9.0806E-01 (7.17513E-02)	7.7961E-01 (3.00338E-02) [†]	8.9749E-01 (2.66680E-01) [†]	3.3819E-01 (8.43546E-02) [†]	9.9849E-01 (1.02165E-01) [†]	1.0617E+00 (4.52105E-01) [†]
	8	5.1983E-01 (6.96936E-02)	5.5871E+00 (1.19030E+01) [†]	7.7478E-01 (1.97824E-01) [†]	2.1625E-01 (6.57912E-02) [†]	9.9818E-02 (3.21026E-02) [†]	1.3082E-01 (3.92564E-01) [†]
10	3.7010E-01 (6.41427E-02)	6.7953E+00 (1.34618E+01) [†]	7.3715E-01 (3.20770E-01) [†]	1.5727E-01 (4.25823E-02) [†]	6.7204E-01 (2.40584E-02) [†]	3.1103E-02 (4.48038E-02) [†]	

	3	9.8798E-01 (1.24042E-02)	8.8260E-01 (2.40016E-02) [†]	6.8468E-01 (2.64578E-02) [†]	2.7246E-01 (6.91272E-02) [†]	4.4127E-01 (2.73807E-02) [†]	6.0373E-01 (2.89793E-03) [†]
	4	<u>1.0198E+00 (6.74614E-03)</u>	9.1986E-01 (4.11008E-02) [†]	8.7220E-01 (2.73173E-02) [†]	2.3443E-01 (7.24373E-02) [†]	5.8514E-01 (1.84067E-02) [†]	1.1324E+00 (7.99399E-03)[†]
DTLZ2	5	<u>1.0296E+00 (7.83495E-03)</u>	9.3372E-01 (6.11612E-02) [†]	9.5388E-01 (3.10375E-02) [†]	3.5310E-01 (9.60867E-02) [†]	6.4514E-01 (2.30645E-02) [†]	1.3012E+00 (5.25382E-03)[†]
	6	<u>1.1161E+00 (2.33482E-02)</u>	8.3860E-01 (2.37732E-02) [†]	9.5260E-01 (3.97041E-02) [†]	2.9994E-01 (5.69157E-02) [†]	6.7853E-01 (3.23148E-02) [†]	1.3499E+00 (2.26676E-02)[†]
	8	9.8795E-01 (6.45634E-03)	<u>1.0236E+00 (1.25375E-01)</u>	9.1789E-01 (2.48723E-02) [†]	2.8558E-01 (7.24260E-02) [†]	6.4368E-01 (2.45093E-02) [†]	1.0534E+00 (2.47743E-01)
	10	<u>9.7058E-01 (9.77224E-03)</u>	1.1440E+00 (1.84157E-01)[†]	9.7039E-01 (1.69468E-02)	2.6161E-01 (5.65284E-02) [†]	6.6149E-01 (2.40152E-02) [†]	2.1640E-01 (2.45770E-01) [†]
	3	9.9727E-01 (1.18592E-02)	8.7523E-01 (3.60858E-02) [†]	5.6777E-01 (1.60735E-01) [†]	2.5400E-01 (1.34175E-01) [†]	5.8864E-01 (9.58502E-03) [†]	5.9554E-01 (3.14145E-02) [†]
	4	9.7707E-01 (8.21852E-03)	8.7997E-01 (1.05353E-01) [†]	6.3245E-01 (2.74485E-01) [†]	2.6224E-01 (8.97735E-02) [†]	6.0668E-01 (1.93795E-02) [†]	9.2399E-01 (2.43623E-01)
	5	1.0168E+00 (6.54401E-03)	7.4548E-01 (1.93596E-01) [†]	4.1279E-01 (3.28220E-01) [†]	1.9796E-01 (1.09482E-01) [†]	6.4776E-01 (1.83865E-02) [†]	2.2725E-01 (3.28181E-01) [†]
DTLZ3	6	1.1173E+00 (1.13113E-02)	<u>8.7064E-01 (4.34695E-01)[†]</u>	2.8565E-01 (3.17094E-01) [†]	1.4374E-01 (8.15058E-02) [†]	6.5772E-01 (1.12801E-01) [†]	0.0000E+00 (0.00000E+00) [†]
	8	9.7677E-01 (9.65714E-03)	6.2783E-01 (1.34328E+00)	1.9315E-01 (2.84029E-01) [†]	1.2406E-01 (4.04796E-02) [†]	3.9382E-01 (2.50102E-01) [†]	0.0000E+00 (0.00000E+00) [†]
	10	9.3645E-01 (5.16393E-03)	5.7879E-03 (1.54484E-02) [†]	6.0733E-01 (2.74915E-01) [†]	1.3581E-01 (5.93626E-02) [†]	2.6957E-01 (2.31341E-01) [†]	0.0000E+00 (0.00000E+00) [†]
	3	6.6197E-01 (4.27371E-01)	4.6188E-01 (3.95122E-01)	6.1770E-01 (1.83316E-01)	2.1352E-01 (1.48506E-01) [†]	5.7741E-01 (5.28796E-03)	4.1289E-01 (2.54687E-01) [†]
	4	<u>9.1824E-01 (2.05013E-01)</u>	4.4573E-01 (3.64057E-01) [†]	7.4143E-01 (2.48123E-01) [†]	2.5653E-01 (1.87011E-01) [†]	5.7539E-01 (1.76688E-02) [†]	1.0616E+00 (2.37711E-01) [†]
DTLZ4	5	<u>9.5754E-01 (1.68432E-01)</u>	3.9015E-01 (2.93544E-01) [†]	8.6074E-01 (1.58894E-01) [†]	2.6176E-01 (1.86893E-01) [†]	6.3258E-01 (3.13921E-02) [†]	1.1518E+00 (3.40926E-01)[†]
	6	1.0725E+00 (6.38296E-02)	5.1194E-01 (2.85566E-01) [†]	9.5005E-01 (3.82742E-02) [†]	3.2241E-01 (2.01053E-01) [†]	6.8764E-01 (2.43298E-02)	9.6736E-01 (5.37545E-01)
	8	9.4195E-01 (1.74099E-02)	7.2219E-01 (3.03730E-01) [†]	<u>9.2790E-01 (2.26546E-02)[†]</u>	3.5804E-01 (7.20134E-02) [†]	6.1996E-01 (2.37531E-02) [†]	<u>5.2325E-01 (5.52567E-01)[†]</u>
	10	9.1121E-01 (2.45957E-02)	9.6665E-01 (3.79451E-01)	9.6638E-01 (1.14391E-02) [†]	3.2112E-01 (8.25448E-02) [†]	6.9372E-01 (3.57074E-02) [†]	1.0501E-01 (2.40114E-01) [†]
	3	9.3890E-01 (4.68561E-02)	9.4044E-01 (1.02859E-02) [†]	9.2575E-01 (3.86653E-02) [†]	<u>9.5395E-01 (6.11423E-02)[†]</u>	<u>1.3685E+00 (3.30802E-02)[†]</u>	9.2722E-01 (6.44623E-02) [†]
	4	2.1707E+00 (1.27853E-01)	<u>1.9068E+00 (1.34963E-01)[†]</u>	9.9110E-01 (1.54379E-01) [†]	1.2684E+00 (4.40710E-01) [†]	1.3788E+00 (8.11361E-02) [†]	9.5825E-01 (2.91647E-01) [†]
DTLZ5	5	1.7299E+00 (1.43731E-01)	<u>1.6524E+00 (1.18251E-01)[†]</u>	1.1591E+00 (1.75680E-01) [†]	1.3607E+00 (3.95131E-01) [†]	1.2771E+00 (1.41335E-01) [†]	1.1022E+00 (4.99464E-01)
	6	2.4580E+00 (2.24064E-01)	2.7376E+00 (3.75631E-01)	<u>2.6813E+00 (2.33371E-01)</u>	1.4756E+00 (5.42065E-01)	1.5616E+00 (2.20953E-01)	2.1622E+00 (9.78533E-01)
	8	7.0107E+00 (5.94377E-01)	<u>2.4153E+00 (3.37337E-01)[†]</u>	2.1679E+00 (6.84087E-01) [†]	5.2367E-02 (9.60686E-02) [†]	7.2613E-01 (2.18209E-01) [†]	7.9548E-01 (6.32929E-01) [†]
	10	8.5619E+00 (6.27846E-01)	2.3997E+00 (3.22199E-01) [†]	<u>2.4172E+00 (7.50029E-01)[†]</u>	4.9863E-03 (2.73110E-02) [†]	4.0498E-01 (1.09439E-01) [†]	7.6460E-02 (1.46121E-01) [†]
	3	1.3435E+00 (1.34596E-01)	1.3889E+00 (8.52688E-02)	1.3396E+00 (1.72888E-01)	1.1501E+00 (3.41395E-01) [†]	1.7816E+00 (4.67236E-02)[†]	<u>1.4259E+00 (1.18837E-01)[†]</u>
	4	<u>2.4410E+00 (1.26378E-01)</u>	3.0357E+00 (2.93393E-01)[†]	1.5270E+00 (4.90222E-01) [†]	0.0000E+00 (0.00000E+00) [†]	1.5699E+00 (1.05247E-01) [†]	4.5635E-03 (1.74166E-02) [†]
DTLZ6	5	1.7951E+00 (1.42279E-01)	3.0859E-03 (1.69023E-02) [†]	<u>1.4194E+00 (4.09709E-01)[†]</u>	0.0000E+00 (0.00000E+00) [†]	1.1870E+00 (1.78062E-01) [†]	0.0000E+00 (0.00000E+00) [†]
	6	2.8895E+00 (3.80171E-01)	0.0000E+00 (0.00000E+00) [†]	<u>2.5593E-01 (8.02479E-02)[†]</u>	0.0000E+00 (0.00000E+00) [†]	<u>9.4320E-01 (4.62978E-01)[†]</u>	0.0000E+00 (0.00000E+00) [†]
	8	1.9559E+00 (2.16270E-01)	6.4114E-02 (9.25841E-02) [†]	2.6740E-01 (1.18849E-01) [†]	0.0000E+00 (0.00000E+00) [†]	0.0000E+00 (0.00000E+00) [†]	0.0000E+00 (0.00000E+00) [†]
	10	1.8986E+00 (2.20235E-01)	3.7771E-02 (7.68760E-02) [†]	<u>1.4075E-01 (1.32006E-01)[†]</u>	0.0000E+00 (0.00000E+00) [†]	0.0000E+00 (0.00000E+00) [†]	0.0000E+00 (0.00000E+00) [†]

3	9.7062E-01 (1.33304E-01)	9.9912E-01 (1.44470E-01) †	7.1857E-01 (4.28577E-02)†	3.7137E-01 (1.72866E-01)†	7.3749E-01 (2.16690E-02)†	5.6630E-01 (5.12711E-02)†
4	6.6394E-01 (5.01756E-02)	3.2074E-01 (1.09399E-01)†	<u>5.0842E-01 (7.85438E-02)†</u>	2.3610E-01 (7.93598E-02)†	4.7106E-01 (2.01419E-02)†	4.0124E-01 (1.83707E-01)†
5	7.1072E-01 (6.29801E-02)	1.4714E+00 (6.34844E-01)†	<u>8.2760E-01 (3.84418E-02)†</u>	3.5503E-01 (1.60258E-01)†	4.6132E-01 (1.90079E-02)†	2.4188E-01 (9.36043E-02)†
6	8.7237E-01 (3.46582E-02)	<u>6.5235E-01 (4.31959E-01)†</u>	5.4844E-01 (4.80601E-02)†	3.3734E-01 (1.55325E-01)†	2.9776E-01 (2.34195E-02)†	1.0036E-01 (6.84524E-02)†
8	5.9887E-01 (1.52663E-02)	2.2230E+00 (2.02132E+00)†	<u>8.7016E-01 (6.50380E-02)†</u>	6.1437E-02 (6.25518E-02)†	1.5046E-01 (5.11722E-02)†	0.0000E+00 (0.00000E+00)†
10	4.4481E-02 (1.46695E-02)	3.2024E+00 (2.16088E+00)†	9.6667E-01 (6.04123E-02)†	3.4990E-03 (7.53848E-03)†	1.8454E-02 (1.15707E-02)†	0.0000E+00 (0.00000E+00)†

表 8 综合性指标 IGD 的统计数据 (均值和方差)
Table 8 Statistical results of the integrated indicator IGD (mean and SD)

问题	目标数	均值与方差					
		NCEA	ϵ -MOEA	ϵ -GrEA	AR + DMO	MSOPS	NSGA-III
DTLZ1	3	2.2436E-02 (1.63962E-03)	2.4240E-04 (3.24306E-05)†	1.7059E-02 (8.29783E-02)†	1.6188E-02 (4.54827E-02)†	<u>6.9466E-03 (3.67556E-02)†</u>	8.3968E-02 (3.03360E-01)
	4	4.5368E-02 (1.22833E-03)	1.5342E-03 (1.01728E-04)†	5.0108E-02 (1.48793E-01)	7.1744E-03 (1.27516E-02)†	8.0661E-03 (3.25514E-02)†	2.7705E-02 (7.46720E-02)
	5	6.6923E-02 (1.51909E-03)	2.8019E-03 (5.10129E-04)†	6.5782E-02 (3.16118E-01)†	5.9357E-02 (1.54993E-01)	<u>2.8872E-02 (9.41475E-02)</u>	5.3457E-02 (1.27134E-01)†
	6	8.5221E-02 (2.19744E-03)	3.5723E-03 (4.48798E-04)†	4.1469E-02 (1.31608E-01)	5.1421E-02 (1.02472E-01)†	<u>3.8983E-02 (1.02165E-01)†</u>	1.6179E-01 (2.49415E-01)†
	8	5.5829E-03 (1.27751E-04)	<u>6.1082E-03 (9.60869E-04)†</u>	8.6450E-02 (3.30601E-01)	3.6173E-02 (9.55063E-02)†	9.9818E-02 (1.48950E-01)†	9.0905E-01 (1.19337E+00)†
DTLZ2	10	8.2959E-03 (6.14113E-03)	<u>3.4608E-02 (3.75171E-02)†</u>	4.1050E-02 (2.78262E-02)†	7.8265E-02 (1.99814E-01)†	1.2987E-01 (1.80125E-01)†	2.0312E-01 (4.43355E-01)†
	3	2.4938E-04 (7.71848E-05)	7.5429E-04 (5.67439E-05)†	4.4901E-05 (4.52089E-05)†	4.9012E-04 (1.50824E-04)†	<u>1.1257E-04 (1.33222E-04)†</u>	3.0712E-04 (2.34867E-04)†
	4	3.4325E-04 (1.07068E-04)	2.1259E-03 (1.25929E-04)†	<u>2.4815E-04 (3.10381E-04)†</u>	1.1270E-03 (3.29167E-04)†	2.0637E-04 (1.21438E-04)†	7.0224E-04 (1.24904E-04)†
	5	2.1616E-04 (4.46158E-05)	4.1994E-03 (6.61445E-04)†	4.6204E-04 (1.75780E-04)†	4.1831E-03 (1.24812E-03)†	<u>3.7035E-04 (2.29673E-04)†</u>	1.9392E-03 (2.82516E-04)†
	6	<u>5.7860E-04 (1.84086E-04)</u>	5.6277E-03 (1.97491E-03)†	6.3318E-04 (1.86383E-04)†	9.1966E-03 (2.34274E-03)†	5.2284E-04 (1.85337E-04)†	4.3691E-03 (6.82254E-04)†
DTLZ3	8	2.8758E-04 (1.06852E-04)	6.8790E-03 (8.32033E-04)†	2.2182E-03 (8.86939E-04)†	1.9660E-02 (3.92533E-03)†	<u>1.0396E-03 (2.79993E-04)†</u>	1.1352E-02 (3.21989E-03)†
	10	3.4444E-04 (1.40595E-04)	5.5698E-03 (4.47660E-04)†	1.7998E-03 (3.52157E-04)†	3.1090E-02 (3.81003E-03)†	<u>1.6458E-03 (3.34714E-04)†</u>	4.4401E-03 (2.74564E-03)†
	3	<u>2.8456E-04 (2.41286E-04)</u>	1.3291E-03 (4.28045E-04)†	1.3041E-01 (5.43435E-01)†	4.0661E-03 (8.97266E-03)†	1.5370E-04 (1.11865E-04)†	8.0032E-03 (3.01346E-02)†
	4	3.9361E-04 (2.52161E-04)	<u>4.8620E-03 (2.40060E-03)†</u>	9.6669E-02 (4.69495E-01)†	1.2016E-01 (4.21198E-01)†	6.6798E-03 (2.22771E-02)†	6.3762E-02 (1.68358E-01)
	5	5.6584E-04 (3.77953E-04)	<u>8.9207E-03 (4.14879E-03)†</u>	1.5762E+00 (2.65502E+00)†	1.7752E-02 (4.77234E-02)†	9.4459E-02 (3.57033E-01)†	6.0880E-01 (1.38366E+00)†
DTLZ3	6	5.6684E-04 (3.12144E-04)	<u>1.7783E-02 (1.10904E-02)†</u>	2.9341E+00 (4.02971E+00)†	8.4404E-02 (2.22184E-01)†	2.5392E-01 (6.65749E-01)	2.7222E+00 (2.01272E+00)
	8	6.8971E-04 (3.84881E-04)	1.5738E+00 (2.37967E+00)†	2.1981E+00 (2.32611E+00)†	<u>1.9263E-01 (6.19185E-01)†</u>	1.3256E+00 (1.21451E+00)†	1.8191E+01 (5.70052E+00)†
	10	7.6931E-04 (4.06091E-04)	3.1649E+00 (2.87679E+00)†	2.4362E-01 (6.95779E-01)†	<u>1.4932E-01 (2.98389E-01)†</u>	1.4714E+00 (9.55260E-01)†	1.7709E+01 (1.09209E+01)†

3	2.1106E-04 (1.30383E-04)	9.8535E-04 (4.16771E-04)	<u>1.2073E-04 (2.58715E-04)</u> †	2.5681E-04 (2.91554E-04) †	6.5585E-05 (1.65279E-04) †	2.4774E-04 (1.30491E-04)
4	5.7399E-04 (1.77486E-04)	2.4360E-03 (5.77431E-04) †	<u>2.0198E-04 (3.13694E-04)</u>	1.5170E-03 (3.12700E-03) †	1.7653E-04 (9.46011E-05)	6.7933E-04 (2.68225E-04)
5	2.8600E-04 (1.56925E-04)	5.3361E-03 (1.76792E-03) †	4.3361E-04 (1.76189E-04)	2.4495E-03 (2.17001E-03) †	<u>3.5318E-04 (9.82127E-05)</u>	1.6399E-03 (3.96018E-04)
6	5.9198E-04 (2.68476E-04)	1.0150E-02 (8.10313E-03) †	8.3766E-04 (3.33206E-04) †	4.8072E-03 (3.04760E-03) †	<u>8.3145E-04 (5.98874E-04)</u> †	3.4386E-03 (1.12700E-03) †
8	4.3000E-04 (2.03320E-04)	1.0441E-02 (5.13729E-03) †	2.4109E-03 (1.05389E-03) †	1.4767E-02 (2.59938E-03) †	<u>1.5980E-03 (5.54551E-04)</u> †	5.2812E-03 (4.42282E-03) †
10	3.3791E-04 (1.63041E-04)	1.5882E-02 (1.24883E-02) †	1.6270E-03 (2.48190E-04) †	2.8114E-02 (4.28606E-03) †	<u>2.8111E-03 (9.12415E-04)</u> †	1.5193E-02 (5.35355E-03) †
3	8.2379E-05 (4.28364E-05)	6.0527E-05 (6.42860E-06) †	5.9233E-05 (5.66109E-05) †	8.3765E-04 (1.13766E-03)	1.0821E-01 (2.80056E-03) †	2.0193E-04 (4.83885E-05) †
4	3.4898E-02 (2.52789E-03)	5.0231E-02 (3.20057E-03) †	1.9988E-03 (1.06200E-03)	<u>1.6314E-02 (7.87450E-03)</u> †	1.5362E-01 (3.29054E-03) †	3.3321E-02 (1.53968E-02) †
5	1.7277E-02 (9.20726E-04)	5.1506E-02 (1.82190E-03) †	<u>1.8679E-02 (1.93218E-02)</u> †	2.4014E-02 (6.34334E-03) †	1.8936E-01 (3.08728E-03) †	5.4787E-02 (9.97725E-03) †
6	1.4070E-02 (6.1126E-04)	5.7760E-02 (6.23864E-03) †	5.6970E-02 (3.73031E-03) †	<u>3.4634E-02 (8.44098E-03)</u> †	2.0364E-01 (2.87387E-03) †	7.0914E-02 (1.34050E-02) †
8	4.8683E-02 (4.12569E-03)	5.3952E-02 (4.32586E-03) †	1.0139E-01 (5.95976E-03) †	1.3747E-01 (3.12555E-02) †	2.2976E-01 (2.27175E-03) †	1.0419E-01 (1.47761E-02) †
10	5.4205E-02 (5.08366E-03)	6.0848E-02 (6.67483E-03) †	1.1183E-01 (7.60976E-03) †	1.7080E-01 (2.86372E-02) †	2.3387E-01 (1.92825E-03) †	1.5273E-01 (1.37459E-02) †
3	3.1665E-03 (2.76698E-03)	5.2588E-03 (4.90865E-04) †	<u>3.3264E-03 (1.61298E-03)</u> †	5.1275E-03 (3.55167E-03) †	2.0425E-01 (4.49689E-03) †	9.8696E-03 (7.13856E-03) †
4	<u>4.2131E-02 (5.62475E-03)</u>	1.1064E-01 (9.59272E-03) †	2.5454E-02 (9.72852E-03) †	1.4171E-01 (2.38262E-02) †	3.3404E-01 (7.50779E-03) †	1.6955E-01 (2.03052E-02) †
5	1.5250E-02 (1.92197E-03)	1.5048E-01 (5.64830E-03) †	<u>9.3128E-02 (5.99215E-02)</u> †	2.9541E-01 (2.07511E-02) †	4.9366E-01 (1.37175E-02) †	6.8704E-01 (2.20882E-02) †
6	1.3164E-02 (1.52962E-03)	<u>2.5553E-01 (2.02566E-02)</u> †	3.3090E-02 (1.30923E-01) †	3.3130E-01 (3.18902E-02) †	5.5517E-01 (1.51144E-02) †	8.7899E-01 (1.26348E-03) †
8	1.3972E-02 (1.16462E-03)	<u>2.9568E-01 (1.46263E-01)</u> †	7.6863E-01 (1.65607E-02) †	8.8443E-01 (6.83775E-02) †	3.2865E+00 (4.21908E-01) †	9.6881E+00 (9.73209E-01) †
10	1.6102E-02 (9.86460E-04)	<u>3.4632E-01 (2.74054E-01)</u> †	4.0319E-01 (3.82174E-01) †	7.4899E-01 (5.44463E-02) †	8.1216E-01 (1.81888E-02) †	7.6559E-01 (3.71012E-02) †
3	1.4269E-03 (6.27153E-04)	6.9496E-04 (3.51603E-05)	<u>9.5260E-04 (7.14402E-04)</u>	2.2605E-03 (1.09999E-03) †	4.4358E-03 (6.88598E-04) †	2.6307E-03 (9.13444E-04) †
4	5.6080E-03 (4.49695E-04)	2.3024E-03 (5.55552E-04) †	<u>4.7689E-03 (2.24499E-04)</u>	1.4251E-02 (5.16088E-03) †	6.9670E-03 (1.08085E-03) †	5.4934E-03 (1.15755E-03) †
5	<u>7.6494E-03 (3.39809E-04)</u>	3.3534E-03 (1.06828E-03) †	1.0717E-02 (6.69624E-04) †	8.3798E-02 (2.79588E-02) †	1.0825E-02 (3.33610E-03) †	2.0268E-02 (5.51018E-03) †
6	<u>1.6013E-02 (6.62155E-04)</u>	4.6055E-03 (1.70038E-03)	1.1947E-02 (4.76109E-04) †	1.7956E-01 (4.96227E-02) †	2.4724E-02 (1.41937E-02) †	9.1749E-02 (3.63369E-02) †
8	<u>2.1716E-02 (3.75157E-04)</u>	1.4274E-02 (1.09412E-02) †	2.6989E-02 (1.56032E-03) †	3.3977E-01 (7.68451E-02) †	2.0888E-01 (1.13986E-01) †	9.5229E-01 (2.31546E-01) †
10	<u>4.8018E-02 (2.56958E-03)</u>	1.5505E-02 (1.25040E-02) †	4.9859E-02 (2.27362E-03) †	6.6739E-01 (1.39620E-01) †	6.5667E-01 (2.37870E-01) †	2.6579E+00 (4.98006E-01) †

DTLZ4

DTLZ5

DTLZ6

DTLZ7

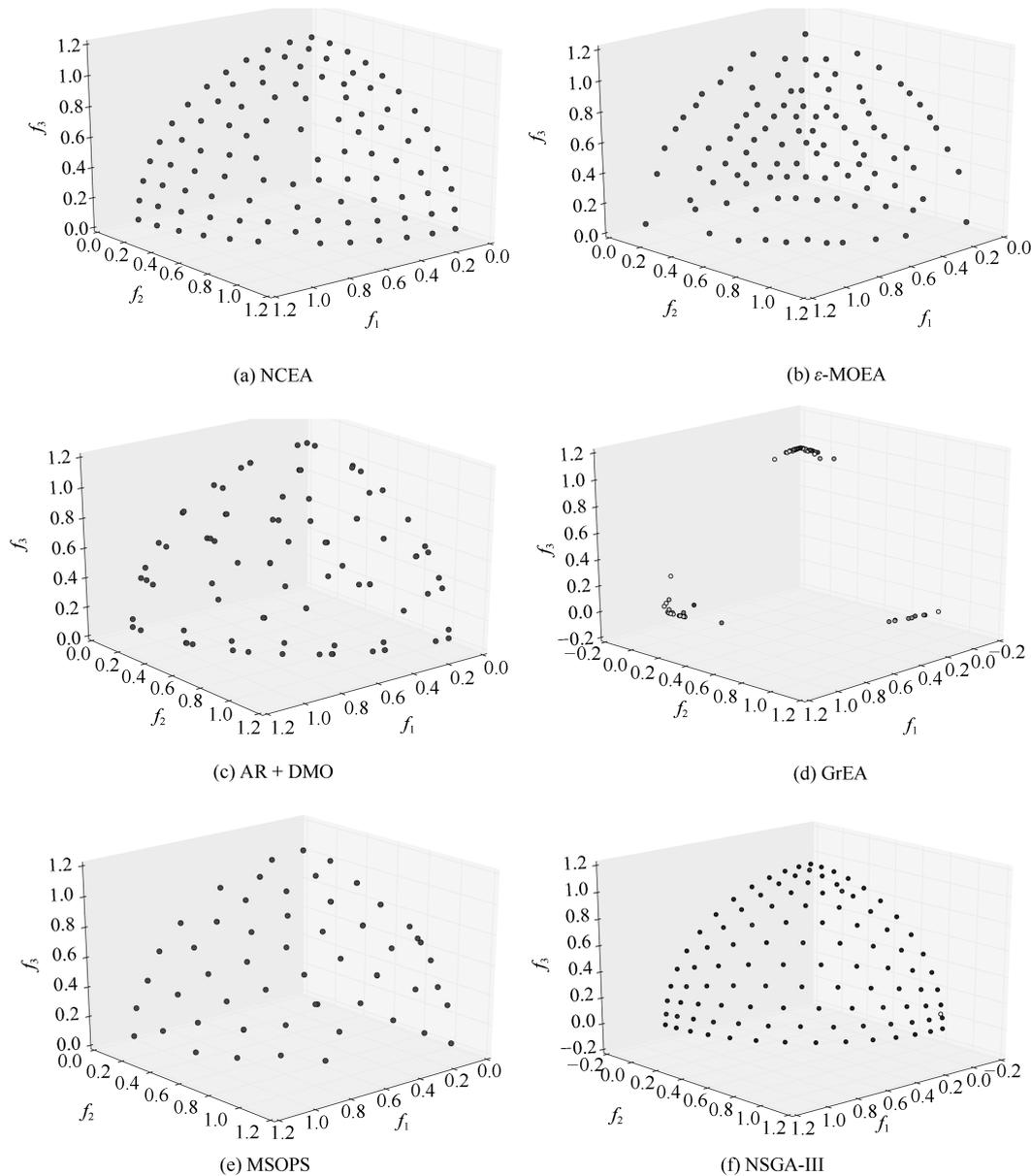


图7 各算法在3目标DTLZ3测试问题上的最终解集

Fig. 7 The final solution set of different algorithms on 3-objective DTLZ3 test problem

示, NCEA 拥有最好的综合性能 (粗体标记的有 23 个, 下划线标记的有 7 个, 占总测试问题的 71%), 但是在 DTLZ1 问题上表现欠佳. 而排名第二的是 ϵ -MOEA, 它在 DTLZ1 和 DTLZ7 问题上获得了最好的综合性能, 但是在其他问题上, 其综合性能不如 NCEA. MSOPS 的综合性能仅次与 ϵ -MOEA, 在 DTLZ1、DTLZ2 和 DTLZ4 问题上取得较好的结果. 虽然 GrEA 的效果不如前面几个算法, 但是在低维问题上的效果也比较理想. 而 AR + DMO 因为分布性能差, NSGA-III 因为收敛性能不理想, 所以这两个算法在综合指标 IGD 上显示的结果也不好.

此外, 为了直观地反映所有算法所得到的最终解集在目标空间中的分布情况, 本文还给出了各个算法的最终解集在目标空间的图形. 在低维情况下, 本文列出了各个算法在 3 维 DTLZ3 的最终解集, 而在目标维数大于 3 的情况下, 本文将采用平行坐标^[27-28]实现 6 维 DTLZ3 数据的可视化, 在平行坐标中, 横坐标表示目标的各个维度, 纵坐标表示个体在每一维上的数值, 而坐标轴中每条折线代表种群中的每个个体.

图 7 描述了各算法在 3 维 DTLZ3 问题下的分布情况. 虽然这 6 种算法基本收敛, 但就分布性而言, 表现最好的是 NSGA-III, 其次是 NCEA, 这两

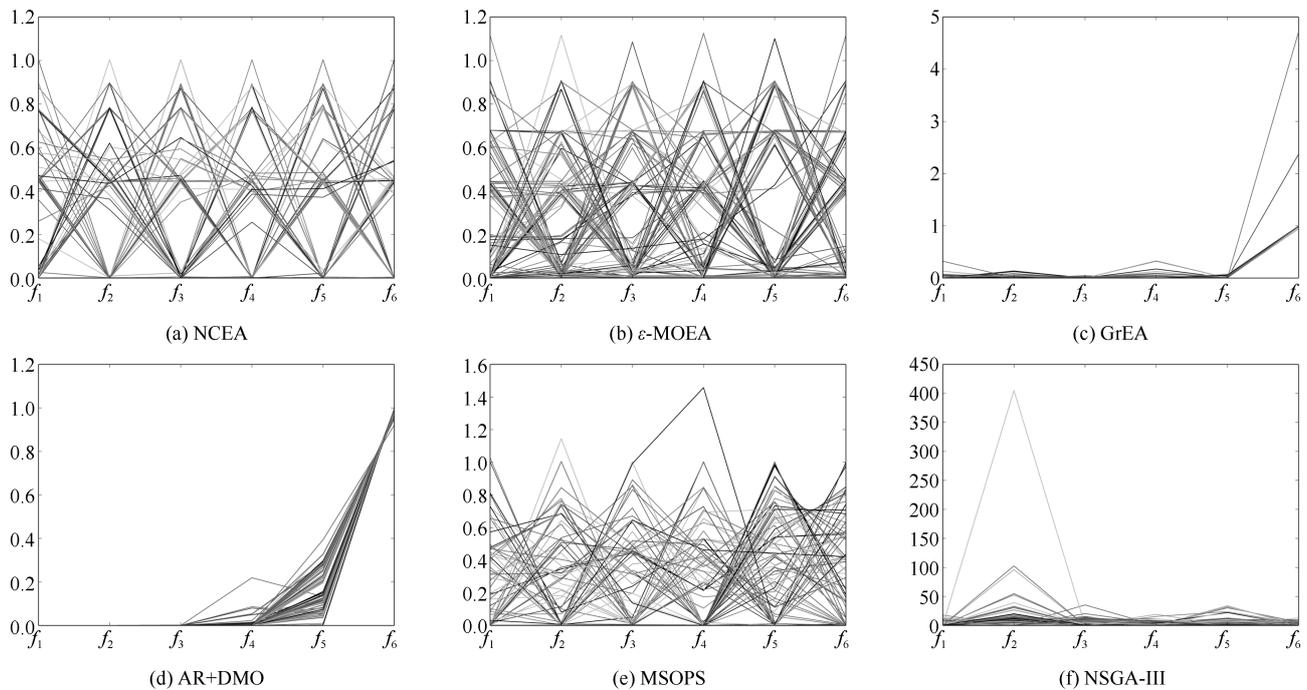


图 8 各算法在 6 目标 DTLZ3 测试问题上的最终解集

Fig. 8 The final solution set of different algorithms on 6-objective DTLZ3 test problem

种算法基本上均匀地分布在最优 Pareto 面上, 其主要原因是 NSGA-III 采用的基于权重的方法来引导种群搜索整个目标空间. 在低维问题中, 权重的设计相对来说比较简单, 而且能够准确地获得一组均匀的权重覆盖到整个 Pareto 面上. 而 ϵ -MOEA, GrEA 和 MSOPS 虽然分布广泛性较好, 但是分布均匀性较差, GrEA 和 MSOPS 都存在点与点重叠的情况, ϵ -MOEA 则不能很好地使个体布满整个最优 Pareto 面上. 特别地, 在 AR + DMO 中, 种群基本上集中在最优 Pareto 面上的三个角落, 分布性能很差.

图 8 给出了各算法在 6 维 DTLZ3 问题下的分布情况. 平行坐标图中的每条折线代表高维空间中的一个点. 就收敛性而言, 处于收敛状态的算法有 NCEA 和 AR + DMO, 其他算法都有一个或多个个体处于未收敛状态. 值得一提的是, 在 NSGA-III 中, 个体在第二维的数值达到了 400. 而就分布性而言, 分布性较好的是 NCEA、 ϵ -MOEA 和 MSOPS, 其他 3 种算法 (GrEA、AR + DMO、NSGA-III) 的分布性都不理想, 这也与 DM 指标所给出的数值相对应.

通过以上实验结果与分析, 可以得出如下结论:

- 1) 在处理高维多目标优化问题时, NCEA 获得了一组收敛性和分布性都较好的非支配解集;
- 2) 收敛信息和分布信息对 NCEA 的性能起了促进作用;

3) 根据以上三个对比实验可以验证, 与其他 5 种算法相比, NCEA 的整体性能表现最好.

4 结论

本文提出了一种基于邻域竞赛机制的高维多目标算法 NCEA: 首先, 通过收敛信息选择优秀个体; 然后通过分布信息以竞赛模式淘汰分布性差的个体. NCEA 一方面提高了种群的收敛性能, 另一方面确保了种群的分布性. 同时, 为了证实算法在处理高维问题的有效性, 本文选择 5 种优秀的算法进行了大量的对比实验. 实验结果表明, NCEA 在 7 个测试问题的不同维度上 (3, 4, 5, 6, 8, 10) 整体性能 (收敛性以及分布性) 最佳. 它在保证种群能快速地收敛到最优 Pareto 面上的同时还兼顾了种群的分布性能, 使得种群能铺满整个最优 Pareto 面上.

本文已经通过大量的分析验证了 NCEA 能够很好地求解高维优化问题, 但是需要指出的是, NCEA 的性能在一定程度上依赖于角度参数 α 的设置. 因此, 在后续的工作中, 需要根据不同问题自适应的产生合适的角度参数, 这样可以大大提高 NCEA 算法在实际优化问题上的适用性, 减少不必要的时间开销. 另外, 为了检验算法的有效性, 引进更多的实际问题检测 NECA 的性能也是今后的研究重点.

References

- 1 Guo Guan-Qi, Yin Cheng, Zeng Wen-Jing, Li Wu, Yan Tai-Shan. Prediction of Pareto dominance by cross similarity of equivalent components. *Acta Automatica Sinica*, 2014, **40**(1): 33–40
(郭观七, 尹呈, 曾文静, 李武, 严太山. 基于等价分量交叉相似性的 Pareto 支配性预测. *自动化学报*, 2014, **40**(1): 33–40)
- 2 Purshouse R C, Fleming P J. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 2007, **11**(6): 770–784
- 3 Li M Q, Yang S X, Liu X H. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2014, **18**(3): 348–365
- 4 Chen Zhen-Xing, Yan Xuan-Hui, Wu Kun-An, Bai Meng. Many-objective optimization integrating open angle based congestion control strategy. *Acta Automatica Sinica*, 2015, **41**(6): 1145–1158
(陈振兴, 严宣辉, 吴坤安, 白猛. 融合张角拥挤控制策略的高维多目标优化. *自动化学报*, 2015, **41**(6): 1145–1158)
- 5 Li M Q, Yang S X, Zheng J H, Liu X H. ETEA: a euclidean minimum spanning tree-based evolutionary algorithm for multiobjective optimization. *Evolutionary Computation*, 2014, **22**(2): 189–230
- 6 Ikeda K, Kita H, Kobayashi S. Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal? In: Proceedings of the 2001 IEEE Congress on Evolutionary Computation. Seoul, South Korea: IEEE, 2001. 957–962
- 7 Li M Q, Yang S X, Liu X H. Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, 2015, **228**: 45–65
- 8 Deb K, Mohan M, Mishra S. Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary Computation*, 2005, **13**(4): 501–525
- 9 Sato H, Aguirre H E, Tanaka K. Controlling dominance area of solutions and its impact on the performance of MOEAs. In: Proceedings of the 2007 International Conference on Evolutionary Multi-Criterion Optimization. Berlin: Springer, 2007. 5–20
- 10 Wang G P, Jiang H W. Fuzzy-dominance and its application in evolutionary many objective optimization. In: Proceedings of the International Conference on Computational Intelligence and Security Workshops (CISW 2007). Washington, D. C., USA: IEEE, 2007. 195–198
- 11 Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: Proceedings of the 8th International Conference on Parallel Problem Solving from Nature. Birmingham, UK: Springer, 2004. 832–842
- 12 Beume N, Naujoks B, Emmerich M. SMS-EMOA: multi-objective selection based on dominated hypervolume. *European Journal of Operational Research*, 2007, **181**(3): 1653–1669
- 13 Hughes E J. Multiple single objective Pareto sampling. In: Proceedings of the 2003 IEEE Congress on Evolutionary Computation. Canberra, Australia: IEEE, 2003. 2678–2684
- 14 Zhang Q F, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, **11**(6): 712–731
- 15 Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 2014, **18**(4): 577–601
- 16 Gong Dun-Wei, Liu Yi-Ping, Sun Xiao-Yan, Han Yu-Yan. Parallel Many-objective evolutionary optimization using objectives decomposition. *Acta Automatica Sinica*, 2015, **41**(8): 1438–1451
(巩敦卫, 刘益萍, 孙晓燕, 韩玉艳. 基于目标分解的高维多目标并行进化优化方法. *自动化学报*, 2015, **41**(8): 1438–1451)
- 17 Adra S F, Fleming P J. A diversity management operator for evolutionary many-objective optimisation. In: Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization. Nantes, France: Springer, 2009. 81–94
- 18 Yang S X, Li M Q, Liu X H, Zheng J H. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2013, **17**(5): 721–736
- 19 Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multiobjective optimization. *Evolutionary Multiobjective Optimization*. In: Proceedings of the Advanced Information and Knowledge Processing. Berlin, Germany: Springer, 2005. 105–145
- 20 Deb A, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(2): 182–197
- 21 van Veldhuizen D A, Lamont G B. Evolutionary computation and convergence to a pareto front. In: Proceedings of the Late Breaking Papers at the Genetic Programming 1998 Conference. Stanford University, California, USA: Citeseer, 1998. 221–228
- 22 Zheng J H, Li M Q. A diversity metric for MOEAs. In: Proceedings of 7th International Conference on Optimization: Techniques and Applications. Kobe, Japan, 2007. 451–452
- 23 Kung H T, Luccio F, Preparata F P. On finding the maxima of a set of vectors. *Journal of the ACM (JACM)*, 1975, **22**(4): 469–476
- 24 Deb K, Jain S. Running performance metrics for evolutionary multi-objective optimization. Technical Report Kangal Report No. 2002004, Indian Institute of Technology, 2002
- 25 Bosman P A N, Thierens D. The balance between proximity and diversity in multi-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2003, **7**(2): 174–188
- 26 Shen R M, Zheng J H, Li M Q, Zou J. Many-objective optimization based on information separation and neighbor punishment selection. *Soft Computing*, 2015, **21**(5): 1109–1128
- 27 Inselberg A. The plane with parallel coordinates. *The Visual Computer*, 1985, **1**(2): 69–91
- 28 Inselberg A, Dimsdale B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: Proceedings of the 1st IEEE Conference on Visualization. San Francisco, California, USA: IEEE, 1990. 361–378



刘元 湘潭大学信息工程学院硕士研究生. 主要研究方向为高维多目标进化算法. E-mail: liu3yuan@gmail.com
(**LIU Yuan** Master student at the College of Information Engineering, Xiangtan University. His research interest covers multi-objective evolutionary algorithms.)



郑金华 湘潭大学信息工程学院教授. 2000 年获得中南大学控制理论与控制工程专业博士学位. 主要研究方向为进化计算, 多目标遗传算法, 机器学习.
E-mail: jhzheng@xtu.edu.com

(**ZHENG Jin-Hua** Professor at the College of Information Engineering, Xiangtan University. He received his

Ph.D. degree in control theory and control engineering from Central South University in 2000. His research interest covers evolutionary computation, multi-objective genetic algorithm, machine learning.)



邹娟 湘潭大学信息工程学院副教授. 2014 年获得湘潭大学应用与数学专业博士学位. 主要研究方向为人工智能, 优化算法设计, 进化计算. 本文通信作者.

E-mail: zoujuan@xtu.edu.com

(**ZOU Juan** Associate professor at the College of Information Engineering, Xiangtan University. She received her

Ph. D. degree in application and mathematics from Xiangtan University in 2014. Her research interest covers artificial intelligence, optimization design, evolutionary computation. Corresponding author of this paper.)



喻果 湘潭大学信息工程学院硕士研究生. 主要研究方向为偏好多目标优化算法. E-mail: yuguo0801@126.com

(**YU Guo** Master student at the College of Information Engineering, Xiangtan University. His research interest covers preference for multi-objective optimization algorithms.)