

第3章

在基于聚合的多目标演化算法中 平衡收敛性和多样性

3.1 前言

本章旨在提高基于聚合的 MOEAs 在高维多目标优化中多样性保持方面的能力,从而更好地平衡收敛性和多样性。本章的贡献概括如下。

(1) 本章提出了一个一般性思想以在基于聚合的 MOEAs 中实现收敛性和多样性的平衡,该思想利用了目标空间中解到权向量的垂直距离。

(2) 本章分别在两个典型的基于聚合的 MOEAs(MOEA/D 和 EFR)中实现了该思想以增强它们在高维多目标优化中的性能,并形成了两个新的算法,即基于距离更新策略的 MOEA/D,简称 MOEA/D-DU;带有排序限制模式的 EFR,简称 EFR-RR。

(3) 本章给出了一个可选的在线的归一化过程,它能嵌入 MOEA/D-DU 和 EFR-RR 中以有效解决非归一化问题。这是一个相当新颖的做法,截至本项研究工作发表为止,已有文献还未对此进行广泛的研究^[45,75]。

(4) 本章所提出的算法基本包含所有在基于分解的 MOEAs 中曾经提出的技术,并将它们扩展到解决 MaOPs 方面。尽管有些概念是类似的,但是本章实际上将它们有机地组合在一个算法中。例如,MOEA/D-DU 继承了系统采样(与文献[45]相同)、邻域定义(与文献[58]相同)、自适应归一化(与文献[45,75]类似)、首次唯一性替代(与文献[75]相同)和以优先顺序替代(本章予以介绍)的优点。

(5) 本章讨论了所提出的算法与其他已有的基于分解的算法的相似性和不同之处。基于大量的实验结果,对为什么这些算法在高维多目标优化中劣于所提算法给出了可能的解释。

(6) 在实验研究中,本章建议了一种可以在标准测试集上更加合理公平地比较 MOEAs(带有或不带有复杂归一化过程)性能的做法。

本章后续部分组织如下:3.2 节简要介绍了一些最近提出的基于分解的算法,它们在某种程度上与本章所提算法类似;3.3 节在分析了 MOEA/D 和 EFR 的缺陷后,给出了本章的基本思想;3.4 节详细介绍了如何利用该基本思想增强 MOEA/D 和 EFR 在高维多



目标优化中的性能;3.5节阐述了本章所采用的实验设计;3.6节从两个方面实验研究了改进算法的性能;3.7节将所提改进算法与若干先进算法进行了大量的实验比较;3.8节对本章工作进行了小结。

3.2 类似算法简介

本节将简要回顾最近提出的与本章所介绍算法类似的几个算法。

(1) Qi 等^[119]提出了一个改进的 MOEA/D 算法,称为 MOEA/D-AWA。MOEA/D-AWA 从两个方面增强了 MOEA/D 的性能。首先,MOEA/D-AWA 基于对原始切比雪夫函数几何性质的分析提出了一种新的权向量初始化方法。其次,MOEA/D-AWA 采用了一种自适应的权向量调整策略以处理具有复杂 Pareto 前沿面的问题。这种策略周期性地调整权值以使子问题的权向量能够自适应地重新分布,从而得到更加均匀分布的解。

(2) Li 等^[73]提出使用稳定匹配模型来协调 MOEA/D 中的选择过程,形成的新的 MOEA/D 变体称作 MOEA/D-STM。在 MOEA/D-STM 中,子问题和解被认为是两组智能体。子问题偏好那些能使它的聚合函数值更低的解,而解则偏好那些对应权向量与之相近的子问题。为了平衡子问题和解之间的偏好关系,MOEA/D-SMT 使用匹配算法将每个解分配到每个子问题上以平衡进化搜索中的收敛性和多样性。与 MOEA/D 不同,MOEA/D-STM 使用生成模式。

(3) Deb 等^[45]提出了一个基于参考点的高维多目标 NSGA-II,称作 NSGA-III,细节请参见 2.1.3 节。需要特别指出的是,NSGA-III 中结合了一种复杂的归一化技术以有效处理非归一化问题。

(4) Asafuddoula 等^[75]针对高维多目标优化提出了一个改进的基于分解的演化算法,称作 I-DBEA。在 I-DBEA 中,子代解只有在不被当前种群中任何解所支配的情况下才尝试通过替代已有解的方式进入种群。子代解以随机的顺序与种群中所有解逐个竞争,直到能够完成一次成功的替代或所有解都已比较完整。竞争的依据是两个距离,即 d_1 和 d_2 ,其中, d_2 就是上文提及的垂直距离。 d_2 较小的解胜出。只有当两个所比较解对应的 d_2 值相等时,才进一步考虑 d_1 ,该情形下 d_1 较小的解胜出。另外,I-DBEA 中也包含一个与 NSGA-III 类似的在线归一化过程。

(5) Wang 等^[120]提出了一个使用全局替换策略的 MOEA/D 算法,称作 MOEA/D-GR。在 MOEA/D-GR 中,一旦产生一个新解,它就被关联到能获得最小聚合函数值的那个问题上。然后,与这个子问题最接近的若干子问题被选择作为该子问题的替换邻域,新的解会尝试替换这些子问题的当前解。

注意,MOEA/D-STM 和 MOEA/D-GR 只在 2 和 3 目标问题上进行了研究和验证。



尽管 MOEA/D-AWA 在 MaOPs 问题上进行了测试,但是问题只限于退化问题。NSGA-III 和 I-DBEA 是专门为高维多目标优化而设计的。另外,NSGA-III 和 I-DBEA 虽然采用了分解的思想,但是它们仍然依赖 Pareto 支配来控制收敛性,而不是通过聚合函数,这与本节所提及的其他方法不同。

3.3 基本思想

在基于聚合的 MOEAs 中,切比雪夫函数可能是最常使用的聚合函数类型。本章采用切比雪夫函数的改进版本。设 $\lambda_1, \lambda_2, \dots, \lambda_N$ 是一组在目标空间中均匀分布的权向量, z^* 是理想点,那么对于第 j 个子问题,该函数可以定义为

$$F_j(\mathbf{x}) = \max_{k=1}^m \left\{ \frac{1}{\lambda_{j,k}} | f_k(\mathbf{x}) - z_k^* | \right\} \quad (3.1)$$

其中,对于任意 $k \in \{1, 2, \dots, m\}$, $\lambda_{j,k} \geq 0$ 且 $\sum_{k=1}^m \lambda_{j,k} = 1$ 。若 $\lambda_{j,k} = 0$, $\lambda_{j,k}$ 设置为 10^{-6} 。

这种形式的切比雪夫函数较原始形式^[7]有两个优势。首先,均匀分布的权向量将会使搜索方向在目标空间中也是均匀分布的。其次,每个权向量唯一一对应 Pareto 前沿面上的一个 Pareto 最优解。证明可参见文献[119]。这两个优势在一定程度上缓解了算法在多样性保持方面的难度。

然而在实际中,即使公式(3.1)也是有缺陷的。理想情况下,如果每个如公式(3.1)所定义的函数 F_j 都能够获得最优解,那么就能够同时实现最理想的多样性分布。但是,实际上对于 MOEA/D 和 EFR,这种情况并不成立。一方面,它们所得到的最终解一般都只是近优解,不足以隐式地确定最终种群的多样性。例如,图 3.1 阐释了,在二维目标空间中,在 5 个权向量 $(\lambda_1, \lambda_2, \dots, \lambda_5)$ 的辅助下,所得到最终解(A, B, C, D 和 E)的分布,其中,虚线表示按公式(3.1)分解所得子问题的等高线。可以看出,解的分布并不如权向量的分布那么均匀。这是因为尽管 B 和 D 能够在函数 F_2 和 F_4 上取得较好的值,但是二者均与它们各自所对应的搜索方向偏离较远。另一方面,更加重要的问题出现在进化过程中,在其中,如果只依赖于聚合函数值,解的选择将会被误导。例如,在图 3.1 中,存在另一个解 F,它在 F_2 上的值略差于 B。在 MOEA/D 中, F 有很大的可能在更新过程中替换 B。而在 EFR

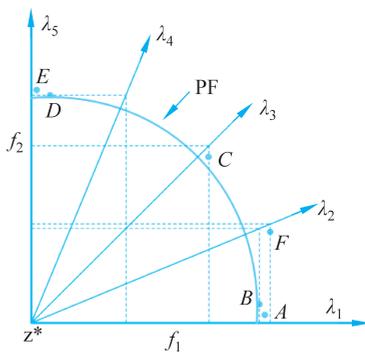


图 3.1 二维目标空间中解的分布示意图



中,因为在函数 F_2 上, B 比 F 获得了更好的排序序号,所以 F 非常可能在环境选择过程中被淘汰。然而直觉上,对于权向量 λ_2 , F 实际上是更佳的选择。

值得提及的是,在进化的早期,解通常是远离 Pareto 前沿面的,这时这种误导性的选择更容易发生,从而可能导致搜索偏向 Pareto 前沿面的局部区域。

上述提及的两种情形可以都归因于一个事实,那就是,在目标空间中远离 λ_j 的解也能得到相对较好的 F_j 值。公式(3.1)的等高线可以很好地解释这一问题,且由于稀疏分布的解以及指数级增长的超体积,这个问题在高维多目标空间中将会变得更加严重。

考虑到所有这些因素,本章的研究动机是,在 MOEA/D 和 EFR 中,不仅要考虑一个解的聚合函数值,而且要考虑它到相应权向量的垂直距离。这种做法期望能够使解接近其所对应的权向量,从而显式地在进化过程中保持解的理想分布,最终实现在高维多目标优化中收敛性和多样性的平衡。值得一提的是,基于惩罚的边界交叉函数(PBI)^[7]在某种程度上隐式地考虑了解与权向量的接近程度,但是它仍然存在上述提到的问题。不失一般性,本章只考虑公式(3.1)中定义的改进的切比雪夫函数。

假设 $f(x) = (f_1(x), f_2(x), \dots, f_m(x))^T$ 是解 x 的目标向量, L 是以方向 λ_j 通过 z^* 的直线,且 u 是 $f(x)$ 在 L 上的投影。设 $d_{j,2}(x)$ 是在目标空间中从解 x 到权值向量 λ_j 的垂直距离,那么它可以按下式计算:

$$d_{j,2}(x) = \| f(x) - z^* - d_{j,1}(x) (\lambda_j / \|\lambda_j\|) \| \quad (3.2)$$

其中, $d_{j,1}(x)$ 是 z^* 和 u 之间的距离,且它能够通过下式得到:

$$d_{j,1}(x) = \| (f(x) - z^*)^T \lambda_j \| / \|\lambda_j\| \quad (3.3)$$

图 3.2 在二维目标空间中阐释了垂直距离 $d_{j,2}(x)$ 。接下来,将详细描述如何利用 $d_{j,2}(x)$ 分别增强 MOEA/D 和 EFR 的性能。

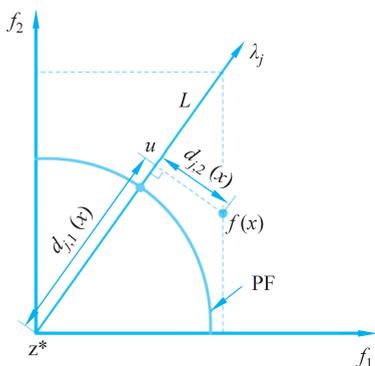


图 3.2 二维目标空间中解到权向量的垂直距离的示意图

3.4 算法详解

在本节中,两个改进的算法 MOEA/D-DU 和 EFR-RR 将分别在 3.4.1 节和 3.4.2 节中详细介绍。3.4.3 节提供了一个可选的归一化过程,它可以嵌入所提算法之中。3.4.4 节分别简要分析了 MOEA/D-DU 和 EFR-RR 在每代中的计算复杂度。3.4.5 节将讨论所提算法与文献中一些已有算法的相似点和不同点。





3.4.1 增强 MOEA/D

MOEA/D-DU 的算法框架如算法 3-1 所示。算法首先产生一组均匀分布的权向量 $\mathbf{A} = \lambda_1, \lambda_2, \dots, \lambda_N$, 其中 λ_j 确定了第 j 个子问题, 即 F_j 。与 NSGA-III^[45] 类似, MOEA/D-DU 也采用了 Das 和 Dennis 的系统方法^[121] 来产生这些结构化的权向量。在权向量生成之后, 随机初始化一个包含 N 个解 x_1, x_2, \dots, x_N 的种群, 其中 x_j 表示第 j 个子问题的当前解。在步骤 3 中, 初始化理想点 z^* 。因为精确地计算 z_i^* 是非常耗时的, 它实际上是由当前所找到的目标 f_i 的最小值来估计的, 且在搜索过程中不断更新。MOEA/D-DU 仍然采用与它的前身算法^[58] 相同的交配限制模式来产生子代解, 因此在步骤 4~步骤 6 中, 需要确定每个子问题 F_i 的邻域 $B(i)$ 。步骤 7~步骤 8 迭代执行直至终止条件满足。在每一迭代中, 解 x_i 的交配解 x_k 以概率 δ 从它的邻域 $B(i)$ 中选择, 以 $1-\delta$ 的概率从整个种群中选取。然后对 x_i 和 x_k 执行遗传算子, 即模拟两点交叉和多项式变异^[92], 以得到一个新的解 y , 最后使用 y 来更新理想点和当前种群。

算法 3-1 MOEA/D-DU 的算法框架

```

1: 生成一组权向量  $\mathbf{A} \leftarrow \{\lambda_1, \lambda_2, \dots, \lambda_N\}$ 
2: 初始化种群  $P \leftarrow \{x_1, x_2, \dots, x_N\}$ 
3: 初始化理想点  $z^* \leftarrow (z_1^*, z_2^*, \dots, z_m^*)^T$ 
4: for  $i \leftarrow 1$  to  $N$  do
5:      $B(i) \leftarrow \{i_1, i_2, \dots, i_T\}$ , 其中  $\lambda_{i_1}, \lambda_{i_2}, \dots, \lambda_{i_T}$  是离  $\lambda_i$  最近的  $T$  个权向量
6: end for
7: while 终止条件不满足 do
8:     for  $i \leftarrow 1$  to  $N$  do
9:         if  $\text{rand}() < \delta$  then
10:             $E \leftarrow B(i)$ 
11:        else
12:             $E \leftarrow \{1, 2, \dots, N\}$ 
13:        end if
14:        随机选择一个下标  $k \in E$  且  $k \neq i$ 
15:         $y \leftarrow \text{GeneticOperators}(x_i, x_k)$ 
16:         $\text{UpdateIdealPoint}(y, z^*)$ 
17:         $\text{UpdateCurrentPopulation}(y, z^*, \mathbf{A}, P, K)$ 
18:    end if
19: end while
20: return 所有  $P$  中的非支配解

```

更新策略(算法 3-1 中的步骤 17)是 MOEA/D-DU 中具有独特性的步骤, 也是与原





始 MOEA/D 的显著不同之处。算法 3-2 详细阐述了该策略,其运行过程如下。一旦生成新解 \mathbf{y} ,则分别计算它与每个权向量 λ_j ,即对于 $j=1,2,\dots,N$,计算垂直距离 $d_{j,2}(\mathbf{y})$, $j=1,2,\dots,N$ 。然后从这 N 个距离中选择 K 个最小的距离,其中, $K \ll N$ 是一个控制参数。假设这 K 个最小的距离是 $d_{j_1,2}(\mathbf{y}), d_{j_2,2}(\mathbf{y}), \dots, d_{j_K,2}(\mathbf{y})$,且以非递减的顺序排列,即 $d_{j_1,2}(\mathbf{y}) \leq d_{j_2,2}(\mathbf{y}) \leq d_{j_K,2}(\mathbf{y})$ 。解 \mathbf{y} 与解 $\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \dots, \mathbf{x}_{j_K}$ 逐一进行比较,如果一个解 $\mathbf{x}_{j_k}, k \in \{1,2,\dots,K\}$,满足 $F_{j_k}(\mathbf{x}_{j_k}) > F_{j_k}(\mathbf{y})$,那么解 \mathbf{x}_{j_k} 将被解 \mathbf{y} 所替代,更新过程终止。由上述可见,MOEA/D-DU 使用了稳态模式,在当前种群中最多只有一个解可以被新产生的解 \mathbf{y} 所替代。

算法 3-2 UpdateCurrentPopulation($\mathbf{y}, \mathbf{z}^*, \mathbf{A}, P, K$)

```

1: for  $j \leftarrow 1$  to  $N$  do
2:     计算从  $\mathbf{y}$  到权向量  $\lambda_j$  的垂直距离,即  $d_{j,2}(\mathbf{y})$ 
3: end for
4: 从  $N$  个距离  $d_{j,2}(\mathbf{y}), j=1,2,\dots,N$  中选出最小的  $K$  个距离,得到
      
$$d_{j_1,2}(\mathbf{y}) \leq d_{j_2,2}(\mathbf{y}) \leq d_{j_K,2}(\mathbf{y})$$

      
$$d_{j_2,2}(\mathbf{y}) \leq \dots \leq d_{j_K,2}(\mathbf{y})$$

5: for  $k \leftarrow 1$  to  $K$  do
6:     if  $F_{j_k}(\mathbf{y}) < F_{j_k}(\mathbf{x}_{j_k})$  then
7:          $\mathbf{x}_{j_k} \leftarrow \mathbf{y}$ 
8:     return
9: end if
10: end for

```

3.4.2 增强 EFR

由 2.1.2 节可知,在 EFR 中,所有在 R_t 中的解均会参与每个适应度函数 F_j 的排序。然而,正如 3.3 节中所提及的,这可能会对解的选择产生误导。为了缓解这个问题,本章提出了 EFR 的一个新的版本,称为 EFR-RR,其中引入了限制排序的模式,即一个解只能被允许在部分适应度函数上进行排序,这部分函数的选取依据是该解在目标空间中与这些函数所对应的权向量较为接近。详细过程解释如下。对每个解 $\mathbf{x} \in R_t$,定义一个大小为 K ($K \ll N$) 的集合 $B(\mathbf{x})$,在总共 N 个权向量中,该集合包含与 \mathbf{x} 最为接近的 K 个权向量的下标,接近程度用公式(3.2)中所给出的垂直距离来衡量。解 \mathbf{x} 将仅参与那些 $j \in B(\mathbf{x})$ 的适应度函数 F_j 上的排序。因此,每个解 \mathbf{x} 只有 K 个排序位置,且每个适应度函数也仅对部分解进行排序。在本章中,适应度函数 F_j 设置为如公式(3.1)所示的改进的切比雪夫函数,且这里只考虑最大排序模式,所以 EFR-RR 中解 \mathbf{x} 的全局排序序号可由



式(3.4)给出:

$$R_g(\mathbf{x}) = \min_{j \in B(\mathbf{x})} r_j(\mathbf{x}) \quad (3.4)$$

如果 $K=N$, 式(3.4)实际上等价于式(2.2)。

3.4.3 可选归一化过程

归一化技术对于算法求解非归一化问题非常有效。例如,文献[7]中的研究表明,即使一个简单的归一化过程也能显著提高 MOEA/D 在非归一化问题上的性能。

多目标优化中的在线归一化技术本质上就是估计 \mathbf{z}^* 和 \mathbf{z}^{nad} , 因为在归一化中目标 $f_i(\mathbf{x})$ 会被式(3.5)所替代:

$$\tilde{f}_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - z_i^*}{z_i^{\text{nad}} - z_i^*} \quad (3.5)$$

一般来说,可以通过当前找到的目标 f_i 的最小值来有效估计 z_i^* 。然而, \mathbf{z}^{nad} 的估计是一个困难得多的任务,因为它需要整个 Pareto 前沿面的信息^[122,123]。实际中,归一化可以离线或在线执行。对于离线模式,可以在执行 MOEA 之前,使用一个专门的算法,如文献[123]和[124]所述,对 \mathbf{z}^{nad} 进行估计。对于在线模式, \mathbf{z}^{nad} 的估计是在优化过程中同时进行的。

在 NSGA-III^[45] 提出之前,大多 MOEAs 并不使用在线归一化技术,或者只使用很简单的在线归一化,如通过当前种群中目标 f_i 的最大值来估计 z_i^{nad} ^[7,73]。本章也提供一个类似于 NSGA-III 的在线归一化过程,它可以嵌入所提算法中以有效处理非归一化问题。因为本章的焦点并不是在线归一化,该过程的细节留待 4.2.4 节予以详述。

3.4.4 计算复杂度

MOEA/D-DU 主要的计算代价是在算法 3-2 所述的更新过程上。步骤 1~步骤 3 需要 $O(mN)$ 的计算开销来计算 $d_{j,2}(\mathbf{y}), j=1,2,\dots,N$ 。在步骤 4 中,需要 $O(N \log K)$ 的计算开销来选择最小的 K 个距离并对它们进行排序。步骤 8~步骤 13 最多需要 $O(mK)$ 的计算开销,因此 MOEA/D-DU 在每代中产生 N 个实验解的复杂度为 $O(mN^2)$ 和 $O(N^2 \log K)$ 中的较大者。

EFR-RR 的主要计算代价在于全局排序序号的计算。首先,所有垂直距离的计算总共需要 $O(mN^2)$ 的计算开销。对每个解 \mathbf{x} 计算 $B(\mathbf{x})$ 总共需要 $O(N^2 \log K)$ 的计算开销。假设 C_j 是参与适应度函数 F_j 上排序的解的数目,那么在所有的适应度函数上的排序将要花费 $O\left(\sum_{j=1}^N C_j \log C_j\right)$ 的计算代价。因为 $\sum_{j=1}^N C_j \log C_j < N^2 \log N$, 那么最差情形下总体的复杂度是 $\max\{O(mN^2), N^2 \log N\}$ 。



3.4.5 讨论

在描述了 MOEA/D-DU 和 EFR-RR 的细节后,本节将讨论所提出的算法与 3.2 节所提及算法的主要相似和不同之处。应当事先指出的是,所有这些算法均采用了一组权向量来指引搜索过程。

(1) MOEA/D-DU 与 MOEA/D-AWA: 尽管二者均保证了在目标空间中均匀分布的权向量会产生均匀分布的搜索方向,但是它们的实现方式稍有不同。MOEA/D-AWA 采用了一种新的权向量初始化方法,而 MOEA/D-DU 使用了一种改进的聚合函数。更重要的是,MOEA/D-AWA 旨在周期性地调整权向量以在具有复杂 Pareto 前沿面的问题上得到分布更加均匀的解,而 MOEA/D-DU 意在利用一组固定的权向量更好地平衡收敛性和多样性之间的关系。换言之,MOEA/D-AWA 强调权值调整的策略,而 MOEA/D-DU 强调新的更新种群的策略。

(2) MOEA/D-DU 与 MOEA/D-STM: 二者都使用改进的切比雪夫函数且利用 $d_{j,2}(\mathbf{x})$ 平衡收敛性和多样性,但是它们在实现机制上有着显著的不同。在 MOEA/D-STM 中,聚合函数值和 $d_{j,2}(\mathbf{x})$ 分别被看作两种代理(子问题和解)的偏好,然后使用延迟接受过程找到一个在子问题和解之间的稳定的偏好匹配,从而从联合的种群中选择一半的解作为下一代种群。在 MOEA/D-DU 中,聚合函数值和 $d_{j,2}(\mathbf{x})$ 之间的折中是用一个更加直接的方式来实现的,即新产生的解首先根据 $d_{j,2}(\mathbf{x})$ 选择它最接近的 K 个权向量,且该解只有机会与这 K 个权向量所对应的解进行竞争。另外,MOEA/D-STM 使用了生成模式,而 MOEA/D-DU 是一个稳态的算法。3.7.1 节将说明,在高维多目标优化中,MOEA/D-STM 的性能不能与 MOEA/D-DU 相提并论。

(3) MOEA/D-DU 与 I-DBEA: 二者均是通过改进原始 MOEA/D 中的更新过程来增强其性能,且在进化过程中都需要计算 $d_{j,2}(\mathbf{x})$,也都采用了稳态的模式;然而,MOEA/D-DU 的基本思想与 I-DBEA 有着本质上的不同。对于 I-DBEA,尽管其在 $d_{j,2}(\mathbf{x})$ 之外还使用了 $d_{j,1}(\mathbf{x})$ 来选择解,但是所采用的 $d_{j,2}(\mathbf{x})$ 对于 $d_{j,1}(\mathbf{x})$ 的简单的优先关系使得 $d_{j,1}(\mathbf{x})$ 起作用的概率很低,这是因为 $d_{j,2}(\mathbf{x})$ 是实数值,它几乎总是可以区别两个解。因此,类似于 NSGA-III, I-DBEA 也是强调非支配的且靠近参考线的解。与 I-DBEA 相比,MOEA/D-DU 与原始 MOEA/D 一样,并不依赖于任何 Pareto 支配关系,它通过同时最小化若干聚合函数使种群不断接近 Pareto 前沿面。此外, $d_{j,2}(\mathbf{x})$ 在 MOEA/D-DU 和 I-DBEA 中所起的作用也相当不同。在 I-DBEA 中, $d_{j,2}(\mathbf{x})$ 是直接作为解之间相互比较的指标。而在 MOEA/D-DU 中, $d_{j,2}(\mathbf{x})$ 是用来选择那些有机会被新生成的解所替代的解,且解相互之间仍然按照聚合函数值来比较。3.7.1 节将说明 MOEA/D-DU 在解决高维多目标优化问题方面较 I-DBEA 有着很强的竞争力。



(4) MOEA/D-DU 与 MOEA/D-GR: 二者均对可以被新解所替代的解进行了精心的选择;然而,这两个算法有着相当不同的初衷。MOEA/D-GR 旨在为一个新产生的解找到一个最合适的子问题,且 MOEA/D 中的子问题的邻域概念仍然在 MOEA/D-GR 的替换过程中被使用。与 MOEA/D-GR 不同,MOEA/D-DU 在更新过程中不再使用子问题的邻域概念。相反地,它采用了一个纯几何的观点,只利用 $d_{j,2}(\mathbf{x})$ 来尽可能地避免保留那些虽然取得好的聚合函数值,但是在目标空间中远离聚合函数所对应的权向量的解,从而在高维目标空间中保持解的理想的分布。3.7.1 节将展示在高维多目标优化问题上,MOEA/D-DU 相比于 MOEA/D-GR 有着较大的优势。

(5) EFR-RR 与 NSGA-III: 二者均使用生成模式且均将联合的种群分裂成若干“层”以选择解。然而,它们在环境选择中如何排序解方面存在着显著的不同。NSGA-III 仍然使用 Pareto 支配关系来促进解的收敛,且采用了由一组权向量辅助的小生境保持算子在最后可接受层上选择解,目的是维持种群的多样性。相比于 NSGA-III,EFR-RR 采用聚合函数而不是 Pareto 支配来促进收敛性。需要注意的是,EFR-RR 实际上也在权向量的辅助下执行了小生境过程,因此在 EFR-RR 中每个权向量同时起到了促进收敛性和多样性的作用。而在 NSGA-III 中,每个权向量主要是用来强调多样性。即使就小生境方法而论,NSGA-III 和 EFR-RR 也有着显著的区别。在 NSGA-III 中,每个解只能与一个唯一的权向量相关联,但是在 EFR-RR 中,每个解可以与若干权向量相关联。3.7.2 节将说明 EFR-RR 在非归一化问题上相比于 NSGA-III 是有一定优势的。

3.5 实验设计

本节进行实验设计方面的工作,以为后续研究 MOEA/D-DU 和 EFR-RR 的性能奠定基础。首先,将指定实验中使用的测试问题和性能指标。然后,列出了用来验证所提出算法的 8 种 MOEAs。最后,给出了本章中使用的实验设置。

3.5.1 测试问题

本章使用 DTLZ^[105] 和 WFG^[106] 测试集,并将这些问题分为两组。第一组问题均是归一化测试问题,包括 DTLZ1~4、DTLZ7 和 WFG1~9。需要注意的是,原始 DTLZ7 和 WFG1~9 问题的目标值在每个目标维度上范围不同,所以是非归一化问题。正如 2.4.1 节所述,它们真实的 Pareto 前沿面是已知的,所以可以按照如下方式修改它们的目标值使其转换为归一化问题。

$$f_i \leftarrow \frac{f_i - z_i^*}{z_i^{\text{nad}} - z_i^*}, \quad i = 1, 2, \dots, m \quad (3.6)$$



因此,对归一化后的 DTLZ7 和 WFG1~9 问题,理想点和最差点分别是 0 和 1。该组问题是用来测试不包含显式归一化过程的算法的性能,表 3.1 概括了它们的主要特征。

表 3.1 测试问题特征

问 题	特 征
DTLZ1	线性的、多峰的
DTLZ2	凹的
DTLZ3	凹的、多峰的
DTLZ4	凹的、有偏的
DTLZ7	混合的、非连通的、多峰的
WFG1	混合的、有偏的
WFG2	凸的、非连通的、多峰的、不可分解的
WFG3	线性的、退化的、不可分解的
WFG4	凹的、多峰的
WFG5	凹的、欺骗的
WFG6	凹的、不可分解的
WFG7	凹的、有偏的
WFG8	凹的、有偏的、不可分解的
WFG9	凹的、有偏的、多峰的、欺骗的、不可分解的

第二组问题都是非归一化问题,包括非归一化的 DTLZ1~2 问题^[45]和原始的 WFG4~9 问题。非归一化的 DTLZ1 和 DTLZ2 分别是原始 DTLZ1 和 DTLZ2 的修改版本。举例说明,如果比例因子是 10^i ,那么非归一化的 DTLZ1 以如下方式修改原始 DTLZ1 的目标函数。

$$f_i \leftarrow 10^{i-1} f_i, \quad i = 1, 2, \dots, m \quad (3.7)$$

该组测试问题用来测试嵌有归一化过程的算法在处理非归一化问题方面的性能。

如 2.4.1 节所述,上述问题均可扩展到任意的目标维数和决策变量维数。本章考虑目标数目 $m \in \{2, 5, 8, 10, 13\}$ 。DTLZ 问题的决策变量数目由 $n = m + k - 1$ 确定,其中, m 为目标数目,对 DTLZ1, k 设置为 5;对 DTLZ2~6, k 设置为 10;对 DTLZ7, k 设置为 20。对所有 WFG 问题,决策变量数目设置为 24,位置相关参数设置为 $m - 1$ 。对于非归一化 DTLZ1 和 DTLZ2 问题,本章对不同目标数目使用的比例因子如表 3.2 所示。



表 3.2 非归一化 DTLZ1 和 DTLZ2 问题的比例因子

目标数量(m)	比例因子	
	DTLZ1	DTLZ2
2	10^i	10^i
5	10^i	10^i
8	3^i	3^i
10	2^i	3^i
13	1.2^i	2^i

3.5.2 性能指标

本章使用 HV 指标作为主要评价指标,该指标是 EMO 领域最流行的评价指标之一,它是 Pareto 相容的^[115],它良好的理论特性保证了它是一个非常公平的指标^[48]。如 2.5 节所述,HV 可以在一定程度上综合体现算法的收敛性和多样性。本章在 HV 计算中所使用的参考点为 $1.1z^{\text{nad}}$,其中,每个测试问题的 z^{nad} 均可以解析获得。另外,根据文献 [119]和[125]中的做法,在计算 HV 时,丢弃那些不支配参考点的解。对于非归一化问题,在计算 HV 之前,还需要使用 z^{nad} 和 z^* 对所得解集和参考点的目标值进行归一化。因此,在本章实验中,对 m 目标问题, $HV \in [0, 1.1^m - V_m]$,其中, V_m 是归一化的真实 Pareto 前沿面与坐标轴围成的超体积。对于不超过 10 个目标的问题,使用最近提出的 WFG 算法^[126] 计算 HV;否则,利用文献[50]中提出的蒙特卡罗模拟方法计算 HV,且使用 10 000 000 个采样点以保证精确性。

另外,本章还使用 GD 和 DCI 作为两个辅助的性能指标,分别评价算法的收敛性和多样性。DCI 的计算需要使用一个参数 div 来设置网格环境,根据文献[116]的建议, div 的设置如表 3.3 所示。

 表 3.3 DCI 中 div 的设置

目标数量(m)	5	8	10	13
div	10	6	5	5

3.5.3 比较算法

MOEA/D-DU 和 EFR-RR 将与 8 个 MOEAs 进行对比,其中 4 个是 MOEA/D 变体。



第一个 MOEA/D 变体是 MOEA/D-DE^[58] 稍作修改后的版本。为了比较的公平,将 MOEA/D-DE 中的 DE 算子替代为 MOEA/D-DU 中使用的再生模式。这样做另外的原因是文献[45]中的结果显示 MOEA/D-DE 似乎并不太适合解决 MaOPs。在本章的实验中,该 MOEA/D 版本简写为 MOEA/D,它实际上是 MOEA/D-DU 的基础算法。

另外三个所比较的 MOEA/D 变体分别是 MOEA/D-STM^[73]、MOEA/D-GR^[120] 和 I-DBEA^[75],它们在某种程度上类似于 MOEA/D-DU。基于上述提到的相同的原因,将 MOEA/D-STM 和 MOEA/D-GR 中的 DE 算子替换为 MOEA/D-DU 中使用的再生模式。

原始的 EFR^[60] 也将参与比较,因为它是 EFR-RR 的前身算法。

另外,两个非基于聚合的 MOEAs 也将用来进行比较,即 GrEA^[37] 和 SDE^[46],这两个算法均是针对高维多目标优化专门设计的,它们的整体性能优于许多先进的高维多目标优化算法。对于 SDE,这里使用 SPEA2+SDE,因为它在文献[46]中所考虑的三个版本(NSGA-II+SDE、SPEA2+SDE 和 PESA-II+SDE)中,整体性能最优。

除了 MOEA/D-STM 和 I-DBEA,所有上述提到的 7 个算法在它们的原始文献中并没有采用任何显式的归一化技术。MOEA/D-STM 使用了一个简单的归一化过程,而 I-DBEA 使用了比较复杂的归一化过程。因为在线归一化方法并不是本章关注的重点,所以这里移除 MOEA/D-STM 和 I-DBEA 中的归一化过程。这 7 个算法将在 3.7.1 节与 MOEA/D-DU 和 EFR-RR 在归一化测试问题上进行比较。该做法是消除不同比例的目标值以及归一化过程对 MOEAs 性能的影响,单纯验证所提出策略(MOEA/D-DU 中基于距离的更新策略和 EFR-RR 中排序限制模式)的有效性。

为了研究非归一化测试问题对算法性能的影响,这里进一步将 3.4.3 节提供的归一化过程嵌入 MOEA/D-DU 和 EFR-RR 中,并在 3.7.2 节中比较其与 NSGA-III 的性能,因为 NSGA-III 的特征之一就是包含一个复杂的归一化过程以处理非归一化问题。

除了 GrEA 和 SDE,所有其他 MOEAs 均使用 4.2.2 节中介绍的方法来产生结构化的权向量。MOEA/D、MOEA/D-STM、MOEA/D-GR、MOEA/D-DU、EFR 和 EFR-RR 都使用式(3.1)中定义的改进的切比雪夫函数。

所有 MOEAs 包括 MOEA/D-DU 和 EFR-RR 均在 jMetal^[127] 框架下实现,且在 8GB 内存的 Intel Core i7 2.9GHz 处理器上运行。

3.5.4 实验设置

实验设置包括一般性设置和参数设置。一般性设置如下。

(1) 运行次数和终止条件:每个算法在每个实例上独立运行 30 次,记录性能指标的



平均值。每次运行,算法在进行 $20\,000 \times m$ 次目标函数评价后终止。

(2) 显著性检验:为了在某些情形下检测统计显著性,对两个比较算法所得指标值进行置信度为 95% 的 Wilcoxon 秩和检验^[128]。

对于参数设置,这里首先列出一些公共的设置。

(1) 种群大小:除了算法 GrEA 和 SDE,其他算法采用与权向量数目相同的种群大小,由于受权向量生成方式的限制,这些算法的种群大小并不是任意的,它由参数 H 所控制 ($N = \lfloor \frac{m}{H} + m - 1 \rfloor$)。对于 GrEA 和 SDE,种群大小可以设置为任何正整数。但是为了公平比较,所有算法对每个问题实例采用相同的种群大小。表 3.4 列出了对不同目标数目的问题所采用的种群大小。为了避免只产生位于边界的权向量,对多于 5 个目标的问题采用两层权向量,具体细节请参见 4.2.2 节。

表 3.4 种群大小设置

目标数目 (m)	分割数 (H)	种群大小 (N)
2	99	100
5	5	210
8	3,3	240
10	3,2	275
13	2,2	182

(2) 交叉和变异相关参数:所有算法均使用模拟两点交叉和多项式变异来生成新解,参数设置如表 3.5 所示。在 EFR、EFR-RR、I-DBEA 和 NSGA-III 中,根据文献[45]、[60]、[75]设置一个较大的交叉分布指数 ($\eta_c = 30$)。

表 3.5 交叉和变异的参数设置

参数名	参数值
交叉概率 (p_c)	1.0
变异概率 (p_m)	$1/n$
交叉的分布指数 (η_c)	20
变异的分布指数 (η_m)	20

(3) 邻域大小 T 和概率 δ :在 MOEA/D、MOEA/D-STM、MOEA/D-GR 和 MOEA/D-DU 中, T 设置为 20, δ 设置为 0.9。

(4) MOEA/D-DU 和 EFR-RR 中的参数 K :MOEA/D-DU 中 K 设置为 5,EFR-RR



中 K 设置为 2。3.6.1 节还将研究参数 K 对算法性能的影响。

MOEA/D、MOEA/D-GR、GrEA 和 SDE 还有它们各自特定的参数。在 MOEA/D 中,每个子代解最多可替代的解的数目(n_r)设置为 1。在 MOEA/D-GR 中,替代邻域的大小(T_r)设置为 5,且每个子代解最多只能替代 1 个解。在 GrEA 中,每个实例所采用的网格分割数(div)如表 3.6 所示,这是依据文献[37]中的建议进行调整的。在 SDE 中,外部存档的大小设置为与种群大小相同的值。

表 3.6 GrEA 中网格分割数的设置

问 题	目标数目(m)	网格分割数(div)
DTLZ1	2,5,8,10,13	18,15,14,16,19
DTLZ2	2,5,8,10,13	19,11,12,15,10
DTLZ3	2,5,8,10,13	19,19,19,19,20
DTLZ4	2,5,8,10,13	19,10,13,15,14
DTLZ7	2,5,8,10,13	20,13,12,11,9
WFG1	2,5,8,10,13	18,17,15,13,10
WFG2	2,5,8,10,13	19,16,18,20,16
WFG3	2,5,8,10,13	20,19,16,16,12
WFG4	2,5,8,10,13	18,13,12,11,9
WFG5	2,5,8,10,13	19,12,13,15,10
WFG6	2,5,8,10,13	20,12,13,12,10
WFG7	2,5,8,10,13	19,12,13,14,9
WFG8	2,5,8,10,13	20,14,12,12,14
WFG9	2,5,8,10,13	19,13,13,15,10

3.6 算法的性能分析

本节将分析所提出的改进算法的性能。首先,探讨参数 K 对 MOEA/D-DU 和 EFR-RR 性能的影响。然后,将探究所提出算法在高维多目标优化中平衡收敛性和多样性的能力。



3.6.1 参数 K 的影响

在 MOEA/D-DU 和 EFR-RR 中, K 是一个平衡收敛性和多样性的主要控制参数。为了研究它们对于 K 的敏感性, 以步长 1 在区间 $[1, 20]$ 上调整 K , 并将它们测试于所有的归一化问题上。除了 K 之外的其他参数设置与 4.3.4 节中的相同。限于篇幅, 图 3.3 只显示了在归一化的 DTLZ4、DTLZ7、WFG3 和 WFG9 问题上, HV 值随 K 值的变化情况, 这里只考虑了 5、8、10 和 13 目标的情况。

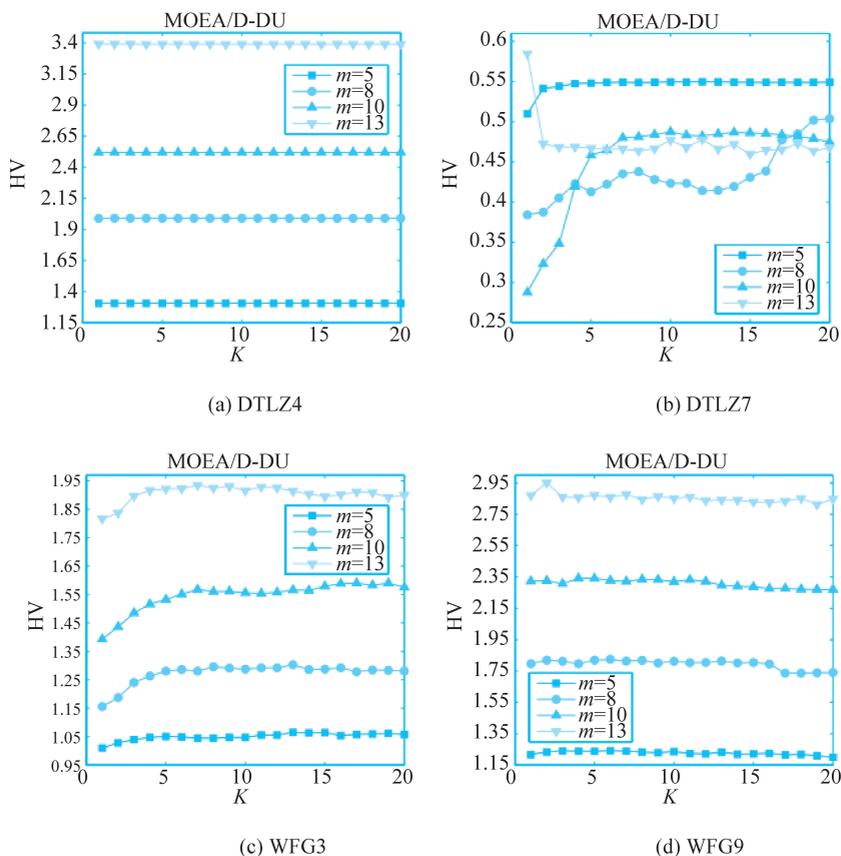


图 3.3 在不同目标维度 m 的情形下, K 在归一化 DTLZ4、DTLZ7、WFG3 和 WFG9 测试问题上对 MOEA/D-DU 和 EFR-RR 性能的影响(图中显示了 30 次独立运行所得的平均 HV 值)

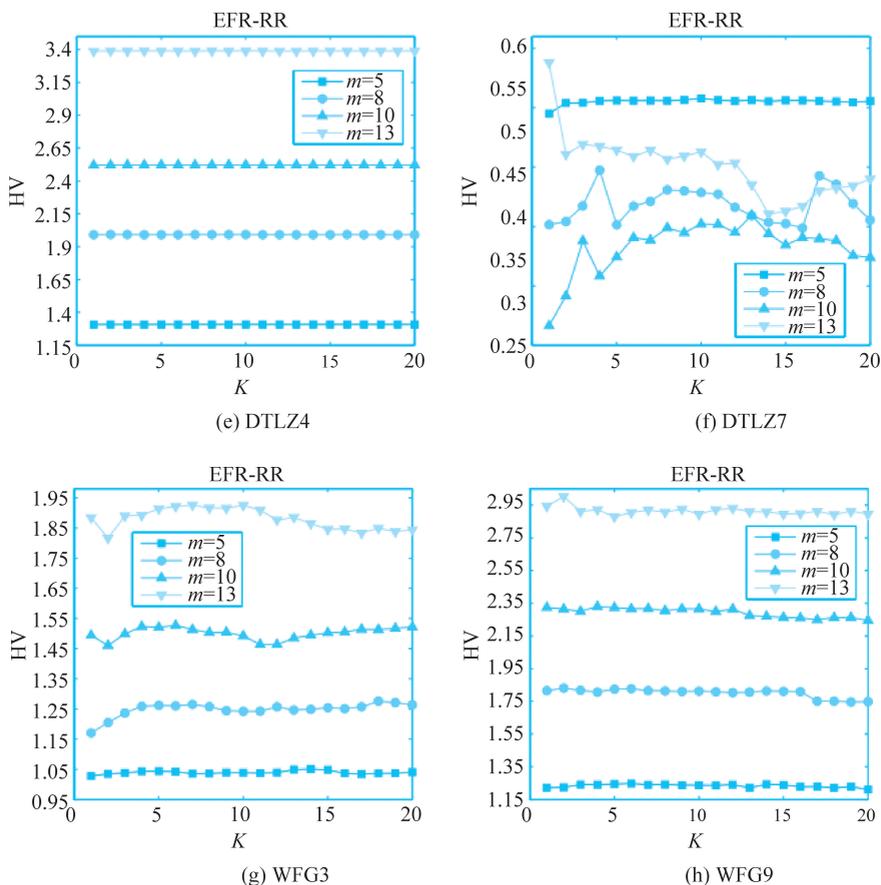


图 3.3 (续)

由图 3.3,可以得到如下实验观察结果。

- (1) 对每个实例, K 对 MOEA/D-DU 和 EFR-RR 的性能有着类似的影响。
- (2) K 的最合适的设置不仅依赖于所要解决的问题,也依赖于问题所具有的目标数目。
- (3) 对于有些问题,如 DTLZ4、MOEA/D-DU 和 EFR-RR 的性能对 K 的设置不是很敏感。二者均能在较大的 K 值范围内获得相当稳定的性能。
- (4) 对于有些问题,如 WFG3 和 WFG9, K 的设置不同可能只会引起微小的性能变化。
- (5) 对于有些问题,如 DTLZ7,需要对 K 进行比较仔细的设置。在这些情形下, K 的微小变化可能会导致性能的剧烈改变。



总体来说,尽管 MOEA/D-DU 和 EFR-RR 的性能会随着 K 值的变化而有着不同程度的波动,但是它们能在 K 的恰当设置下,实现收敛性和多样性的平衡。实验显示在区间 $[1,8]$ 上调整 K 是比较可靠的,因为 K 设置在这个范围内通常可以使 MOEA/D-DU 和 EFR-RR 达到最优或近优的性能。当为所考虑的实例调整 K 时,如果解收敛情况不佳,较大的 K 值一般会更加合适;如果解的多样性不理想,建议使用较小的 K 值。

3.6.2 收敛性和多样性的研究

本节将分别使用 GD 和 DCI 指标研究所提出算法的收敛性和多样性。这里只考虑归一化的 DTLZ1~4 和 WFG4~9 问题。因为这些问题的 Pareto 前沿面呈规则的几何形状,所以 GD 值可以通过分析的方法容易地确定,而不需要在 Pareto 前沿面上采样点。DTLZ1 的 Pareto 前沿面形状是一个超平面,而其他问题的 Pareto 前沿面形状是一个超球面。

MOEA/D-DU 和 EFR-RR 将与它们相应的前身算法(MOEA/D 和 EFR)在高维多目标的场景的情形下进行比较。表 3.7 列出了所得的平均 GD 和 DCI 结果。由这个表可见,MOEA/D-DU 和 EFR-RR 在保持多样性方面都优于它们的前身算法,它们在几乎所有涉及的实例上均获得了更好的 DCI 结果。对于收敛性,MOEA/D-DU 仍然总体上优于 MOEA/D,它在大多数实例上都得到了较好的平均 GD 值;EFR-RR 对于 EFR 也表现出了很强的竞争力,它们分别在不同的实例上获得了较好的 GD 结果。值得注意的是,相比于 EFR,EFR-RR 在 5 目标 DTLZ3 上所得 GD 和 DCI 结果要差得多。但是需要解释的是,这并不能说明 EFR-RR 在该情形下无法实现多样性的保持。这是因为尽管 DCI 是一个多样性比较指标,但是该指标并不考虑远离 Pareto 前沿面的那些解。而 DTLZ3 问题具有大量的局部 Pareto 前沿面,较小的 K 值会不利于 EFR-RR 收敛到 Pareto 前沿面上。另外,如果为 EFR-RR 设置较大的 K ,GD 和 DCI 结果会改善很多。

为了更加直观地描述解在高维目标空间中的分布,图 3.4 使用平行坐标分别绘出了 MOEA/D-DU、MOEA/D、EFR-RR 和 EFR 在 10 目标 WFG4 实例上一次运行所得到的最终的非支配解。正如图 3.4 所示,MOEA/D-DU 和 EFR-RR 均能在所有 10 个目标的 $f_i \in [0,1]$ 范围上找到广泛分布的解,且目标之间的折中可以从图中清晰地看出,而 MOEA/D 和 EFR 都未能对中间解进行有效的覆盖。

上述的比较结果表明,与它们相应的前身算法相比,MOEA/D-DU 和 EFR-RR 在多样性保持方面有着明显的优势,且在收敛性方面也有着不弱的性能,从而它们在收敛性和多样性之间实现了更好的折中。甚至在某些问题上,它们无论在收敛性或多样性方面均表现更优。



表 3.7 MOEA/D-DU(EFR-RR)和 MOEA/D(EFR)在归一化问题上所得的平均 GD 和 DCI 值的比较
(每个实例较优的平均结果以粗体标记)

问题	m	GD		DCI		GD		DCI	
		MOEA/D-DU	MOEA/D	MOEA/D-DU	MOEA/D	EFR-RR	EFR	EFR-RR	EFR
DTLZ1	5	0.000 020	0.004 803	0.969 315	0.896 647	0.000 668	0.000 188	0.943 488	0.920 854
	8	0.000 136	0.000 048	0.992 567	0.972 053	0.000 170	0.000 073	0.989 848	0.949 016
	10	0.000 022	0.000 004	0.991 502	0.959 821	0.000 215	0.000 579	0.988 164	0.922 184
DTLZ2	13	0.000 029	0.000 084	0.992 594	0.936 280	0.000 173	0.000 039	0.994 281	0.945 934
	5	0.000 023	0.000 054	0.939 564	0.836 140	0.000 052	0.000 105	0.933 494	0.854 702
	8	0.000 199	0.000 103	0.985 478	0.968 888	0.000 429	0.000 211	0.985 609	0.925 318
DTLZ3	10	0.000 087	0.000 059	0.976 237	0.930 026	0.000 573	0.000 148	0.978 438	0.876 171
	13	0.000 218	0.000 328	0.990 947	0.942 113	0.000 354	0.000 286	0.998 750	0.964 441
	5	0.011 119	0.009 032	0.814 999	0.701 524	0.312 651	0.002 989	0.007 747	0.995 126
DTLZ4	8	0.003 099	0.000 492	0.982 177	0.945 281	0.003 595	0.031 573	0.985 280	0.926 237
	10	0.001 925	0.000 282	0.961 370	0.914 810	0.005 250	0.001 787	0.984 499	0.857 481
	13	0.001 035	0.001 727	0.975 368	0.804 104	0.000 481	0.019 179	0.997 120	0.939 957
DTLZ4	5	0.000 016	0.000 027	0.957 419	0.934 750	0.000 024	0.000 027	0.955 842	0.946 478
	8	0.000 026	0.000 041	0.982 202	0.980 584	0.000 034	0.000 048	0.986 258	0.990 275
	10	0.000 014	0.000 021	0.998 616	0.981 274	0.000 022	0.000 030	0.998 315	0.995 138
	13	0.000 020	0.000 037	0.993 399	0.952 210	0.000 019	0.000 023	0.995 285	0.993 149





续表

问题	m	GD		DCI		GD		DCI	
		MOEA/D-DU	MOEA/D	MOEA/D-DU	MOEA/D	EFR-RR	EFR	EFR-RR	EFR
WFG4	5	0.000 036	0.000 108	0.958 943	0.237 922	0.000 098	0.000 058	0.960 109	0.286 770
	8	0.000 488	0.000 632	0.993 236	0.951 922	0.000 457	0.000 663	0.985 042	0.931 296
	10	0.000 131	0.000 199	0.989 878	0.907 584	0.000 192	0.000 208	0.998 226	0.848 858
WFG5	13	0.000 217	0.000 229	0.991 063	0.934 429	0.000 214	0.000 226	0.996 185	0.967 426
	5	0.001 296	0.001 421	0.883 937	0.771 605	0.001 341	0.001 361	0.816 306	0.735 464
	8	0.001 131	0.001 155	0.998 100	0.962 539	0.001 180	0.001 283	0.987 310	0.925 904
WFG6	10	0.000 703	0.000 832	0.986 966	0.934 614	0.000 792	0.000 878	0.998 997	0.841 463
	13	0.000 879	0.000 861	0.996 643	0.982 889	0.000 907	0.000 949	0.998 150	0.964 958
	5	0.001 173	0.001 152	0.926 435	0.315 379	0.001 284	0.001 275	0.974 729	0.187 833
WFG7	8	0.001 263	0.001 637	0.998 935	0.970 405	0.001 310	0.001 522	0.989 279	0.927 994
	10	0.000 789	0.000 937	0.993 133	0.932 979	0.000 910	0.001 007	0.999 700	0.819 165
	13	0.001 172	0.001 184	0.996 104	0.962 610	0.001 171	0.001 207	0.999 021	0.961 705
WFG7	5	0.000 066	0.000 136	0.899 289	0.480 338	0.000 099	0.000 064	0.945 729	0.332 050
	8	0.000 552	0.000 966	0.996 412	0.965 031	0.000 271	0.000 445	0.989 559	0.930 570
	10	0.000 128	0.000 168	0.975 515	0.954 160	0.000 096	0.000 186	0.999 086	0.838 123
	13	0.000 786	0.002 391	0.974 228	0.759 269	0.000 374	0.000 372	0.992 426	0.924 018





续表

问题	m	GD		DCI		GD		DCI	
		MOEA/D-DU	MOEA/D	MOEA/D-DU	MOEA/D	EFR-RR	EFR	EFR-RR	EFR
WFG8	5	0.003 256	0.014 374	0.982 783	0.125 524	0.003 480	0.006 584	0.990 924	0.095 518
	8	0.005 102	0.005 095	0.959 361	0.819 171	0.005 643	0.005 110	0.968 154	0.830 303
	10	0.005 566	0.003 776	0.988 397	0.717 678	0.006 110	0.004 606	0.996 009	0.756 254
	13	0.004 611	0.001 271	0.981 147	0.665 580	0.003 793	0.001 848	0.989 844	0.677 784
WFG9	5	0.000 704	0.001 363	0.980 032	0.152 752	0.001 065	0.001 042	0.978 193	0.190 482
	8	0.001 224	0.001 114	0.997 196	0.963 975	0.001 212	0.001 026	0.990 183	0.922 299
	10	0.000 668	0.000 675	0.997 815	0.927 091	0.000 927	0.000 584	0.998 744	0.880 118
	13	0.000 945	0.001 150	0.994 202	0.947 416	0.000 961	0.001 285	0.996 398	0.955 634

