arXiv:2304.03566v1 [cs.NE] 7 Apr 2023

关于使用参考点的基于偏好的进化多目标优化中的无界外部档案与种群规模

Ryoji Tanabe 横滨国立大学 日本神奈川县横滨市
rt.ryoji.tanabe@gmail.com

摘要

尽管种群规模是进化多目标优化（进化多目标优化）中的一个重要参数，但人们对其对基于偏好的进化多目标优化（PBEMO）的影响知之甚少。无界外部档案（UA）在PBEMO中的有效性同样缺乏充分理解，其中UA维护截至目前找到的所有非支配解。此外，现有的UA后处理方法无法处理决策者的偏好信息。在此背景下，本文首先提出了一种基于偏好的后处理方法，用于从UA中选择代表性解。随后，我们研究了UA和种群规模对PBEMO算法性能的影响。结果表明，通过使用UA和所提出的方法，PBEMO算法（例如R-NSGA-II）的性能可以得到显著提升。我们证明，在函数评估预算较小的情况下，对于大多数PBEMO算法而言，采用小于通常使用规模的种群规模是有效的，即使面对多目标问题也是如此。我们发现，在现实世界问题中，感兴趣区域的大小是选择PBEMO算法种群规模时一个不太重要的因素。

CCS 概念

• **计算数学 → 进化算法**。

关键词

基于偏好的进化多目标优化，无界外部档案，种群规模，基准测试

ACM 引用格式：

Ryoji Tanabe. 关于使用参考点的基于偏好的进化多目标优化中的无界外部档案与种群规模。在 遗传与进化计算大会（*GECCO '23*），*2023*年*7*月*15*日至*19*日，葡萄牙里斯本。美国纽约州纽约市ACM，24页。https://doi.org/10.1145/3583131.3590511

1 引言

一般背景。多目标优化的最终目标是寻找决策者偏好的帕累托最优解（决策者）[33]。进化多目标优化有助于寻找一个逼近帕累托前沿的解集。

GECCO '23，*2023*年*7*月*15*日至*19*日，葡萄牙里斯本©2023 版权所有归所有者/作者所有。出版权已授权给 ACM。这是作者版本的作品。此处发布仅供您个人使用。不得再分发。最终正式版本发表于遗传与进化计算会议（*GECCO '23*），*2023*年*7*月*15*日至*19*日，葡萄牙里斯本，https://doi.org/10.1145/3583131.3590511。

目标空间中的帕累托前沿（帕累托前沿）[15]。具有代表性的多目标进化优化算法包括NSGA-II [16], IBEA [52], 和MOEA/D [51]。多目标进化优化算法找到的解集通常用于事后决策，决策者从解集中选择一个解。当决策者无法提供任何偏好信息时，事后决策非常实用。

当决策者的偏好信息先验可用时，可以将其融入 EMO算法 [38]。基于偏好的 EMO（PBEMO）算法 [4, 12] 旨在逼近感兴趣区域（ROI），该区域是由决策者的偏好信息定义的帕累托前沿子区域 [40]。从优化算法的角度来看，逼近帕累托前沿的子区域通常比逼近整个帕累托前沿更容易，尤其是在多目标情况下。从决策者的角度来看，她/他只需考察一组可能符合其偏好的解，而不必考察一组无关的解。也就是说，利用决策者的偏好信息可以减轻她/他的工作负担。

尽管文献中已经提出了多种表达偏好的方式 [4], 使用参考点 [50] 是其中最流行的方法之一 [1, 30]。参考点由决策者所期望的目标值组成。与其他方法相比，基于参考点的方法直观易懂，因此决策者相对容易表达其偏好信-息。本文始终仅关注使用参考点的PBEMO。具有代表性的PBEMO算法包括R-NSGA-II [19], PBEA [49], 和MOEA/D-NUMS [28]。这些PBEMO算法旨在逼近感兴趣区域。

种群规模 μ 是进化算法中最重要的参数之一。一些先前的研究考察了种群规模对传统EMO算法（例如 NSGA-II，而非 R-NSGA-II）性能的影响。例如，Dy-mond等人 [22] 表明，NSGA-II 和 SPEA2 [53] 的最佳种群规模取决于函数评估预算。Brock- hoff等人 [10] 表明，种群规模较大的NSGA-II 在搜索的早期阶段表现不佳，但在后期阶段表现良好。Ishibuchi等人 [26] 表明，最佳种群规模取决于EMO算法的类型。一些先前的研究（例如， [5, 39]）使用自动算法配置器分析了种群规模对EMO算法的影响。

除了少数例外（例如 PESA [14] 和 ϵ-MOEA [18]），多目标进化优化算法在种群中只保留 μ 个解。因此，正如 [8], 所指出的，大多数多目标进化优化算法可能会丢弃决策者可能感兴趣的解。先前的研究（例如 [8, 32]）表明，这个问题可以通过使用无界外部档案（UA）轻松解决，该档案会保存

迄今为止找到的所有非支配解。UA 可以以插件方式集成到任何多目标进化优化算法中。由于 UA 独立于多目标进化优化算法运行，因此不会影响它们的搜索行为。采用 UA 的多目标进化优化算法通常优于未采用 UA 的算法 [5, 32, 46]。

使用不确定性分析的一个缺点是，其规模可能非常大，尤其是在多目标优化中。在实践中，决策者不希望检查如此多的解 [51]。为了解决这一问题，已有一些对不确定性分析进行后处理的方法被提出（例如，[8, 45]）。最简单的方法之一是基于距离的子集选择方法 [45],，其旨在从不确定性分析中选择目标空间内均匀分布的解的一个子集。

研究动机。 与进化多目标优化的情况不同，人们对种群规模 μ 对 PBEMO算法性能的影响知之甚少。文献中也尚未对 PBEMO 中的种群规模 μ进行标准化。表1列出了以往研究中使用的目标数量m、种群规模 μ和最大函数评估次数 max_evals。作为参考，表1展示了提出 NSGA-II 的先前研究 [16] 中的设置。如表1所示，四项先前研究为 μ和 max_evals 设定了不同的取值。PBEMO算法近似的是感兴趣区域，而非整个帕累托前沿。因此，直观地可以预期，与进化多目标优化相比，PBEMO 只需较小的 μ和 max_evals。然而，除 [49], 外，先前研究所用的 μ和 max_evals 相对 [16]而言都较大。

此外，UA在PBEMO领域很少受到关注。除了最早的研究 [23], 之外，以往大多数研究仅根据最终种群的质量讨论 PBEMO算法的性能。不使用UA的原因可能是，现有的UA后处理方法不太可能适用于PBEMO。这是因为现有的后处理方法无法处理决策者的偏好信息。

研究贡献。 受上述讨论的启发，首先，我们提出一种基于偏好的后处理方法，用于从不确定性分析中选择代表性解。然后，我们分析具有不同种群规模的PBEMO算法的行为。通过分析，我们探讨以下三个研究问题：

研究问题1：使用不确定性分析能否提升PBEMO算法的性能？不确定性分析的后处理方法是否应处理决策者的偏好信息？研究问题2：目标数量和函数评估预算如何影响最佳种群规模？以往研究所采用的种群规模是否合理？研究问题3：感兴趣区域的大小是否是选择PBEMO算法种群规模的关键因素？

概要。 第 2 节介绍一些预备知识。第 3 节介绍所提出的后处理方法。第 4 节描述实验设置。第 5 节展示分析结果，以回答三个研究问题。第 6 节总结本文。

补充文件。 图S.∗ 和表 S.∗ 分别表示补充文件中的一幅图和一个表格。

代码可用性。 所提出的后处理方法的实现可在 https://github.com/ryojitanabe/prefpp 获取。

表1： m、 μ**以及先前研究中使用的最大_次评估。**

Ref.	m	μ	max_次评估
NSGA-II [16]	2, 5	100	5×10^4
R-NSGA-II [19]	2、3、5、10	100–500	$5 \times 10^4 - 2.5 \times 10^5$
PBEA [49]	2, 5	20–200	$2 \times 10^3, 2 \times 10^4$
R-MEAD2 [36]	4, ……, 10	200–350	$6 \times 10^4, 9 \times 10^4, 1.05 \times 10^5$
NUMS [28]	2, 3, 5, 8, 10	100–660	$4 \times 10^4 - 1.188 \times 10^6$

2 预备知识

2.1 多目标优化

本文考虑最小化问题。多目标优化旨在寻找一个解 $\mathbf{x} \in \mathbb{X}$，使目标函数向量 $\mathbf{f}\colon \mathbb{X} \to \mathbb{R}^m$ 达到最小，其中 $\mathbb{X} \subseteq \mathbb{R}^n$ 是可行解空间，$\mathbb{R}^m$ 是目标空间。因此，n 是解空间的维度，m 是目标空间的维度。

如果对于所有 $i \in \{1, \ldots, m\}$ 都有 $f_i(\mathrm{x1}) \leq f_i(\mathrm{x2})$，并且至少存在一个索引 i 使得 $f_i(\mathrm{x1}) < f_i(\mathrm{x2})$，则称解 x1 支配 x2。我们用 $\mathrm{x}_1 \prec \mathrm{x2}$ 表示 x1 支配 x2。此外，如果对于所有 $i \in \{1, \ldots, m\}$ 都有 $f_i(\mathrm{x1}) \leq f_i(\mathrm{x2})$，则称 x1 弱支配 x2。如果解 x^* 不被 $\mathbb{X}$ 中的任何解支配，则称其为帕累托最优解。$\mathbb{X}$ 中所有帕累托最优解的集合称为帕累托最优解集 $\mathrm{X}^* = \{\mathrm{x}^* \in \mathbb{X} \mid \nexists \mathrm{x} \in \mathbb{X} \text{ s.t. } \mathrm{x} \prec \mathrm{x}^*\}$。帕累托最优解集在 $\mathbb{R}^m$ 中的映像也称为帕累托前沿 $\mathrm{f}(\mathrm{X}^*)$。理想点和最差点分别由 PF 中 m 个目标函数的最小值和最大值构成。

2.2 使用参考点的基于偏好的多目标优化

使用参考点的基于偏好的多目标优化 $\mathbf{z}$ 涉及在目标空间中的感兴趣区域内寻找一组解。在这里，感兴趣区域是由$\mathbf{z}$定义的帕累托前沿的一个子区域。虽然进化多目标优化旨在逼近帕累托前沿，但PBEMO旨在逼近感兴趣区域。预计逼近感兴趣区域相较于逼近帕累托前沿更为容易。此外，感兴趣区域中的大多数解都可能受到决策者的偏好。因此，在这种情况下，决策者的工作量相对较低。

本文根据欧氏距离，考虑相对于参考点 $\mathbf{f}(\mathbf{x}^{c*})$ 最近的帕累托最优目标向量所对应的感兴趣区域$\mathbf{z}$。该感兴趣区域在 [48] 中被称为 ROI-C。尽管 [48], 中回顾了三个感兴趣区域，但该感兴趣区域是最直观的感兴趣区域。从数学上来说，该感兴趣区域定义如下：

$$\mathrm{ROI} = \left\{\mathbf{f}(\mathbf{x}^*) \in \mathbf{f}(\mathcal{X}^*) \mid \mathtt{distance}\left(\mathbf{f}(\mathbf{x}^*), \mathbf{f}(\mathbf{x}^{c*})\right) < r\right\}, \quad (1)$$

$$\mathbf{x}^{c*} = \operatorname*{argmin}_{\mathbf{x}^* \in \mathcal{X}^*} \left\{\mathtt{distance}\left(\mathbf{f}(\mathbf{x}^*), \mathbf{z}\right)\right\},$$

其中，distance(·,·)返回两个输入之间的欧氏距离。在公式 (1) 中，r 决定了感兴趣区域的半径。

图1展示了双目标DTLZ1问题中感兴趣区域的一个示例，其帕累托前沿呈线性形状。如图1所示，感兴趣区域是以 $\mathbf{f}(\mathbf{x}^{c*})$ 为中心、半径为 r 的超球体内帕累托最优解的目标向量集合。正如 [48], 所讨论的，一些 PBEMO算法（例如，R-NSGA-II [19] 和 R-MEAD2 [36]）被隐式设计为逼近该感兴趣区域。

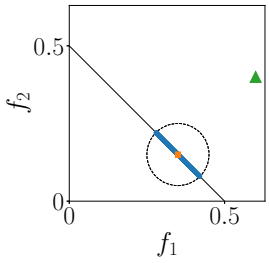


图1：DTLZ1上感兴趣区域的示例，其中 $m = 2$，其中 ▲是参考点 $\mathbf{z} = (0.6,\ 0.4)^\top$，■是距离 $\mathbf{f}(\mathbf{x}_{c*})$ 最近的帕累托最优解的目标向量 至 z，且 $r = 0.1$。虚线圆内帕累托最优解的目标向量集合 •即为感兴趣区域。

2.3 PBEMO的质量指标

质量指标用于评估一组目标向量 [31, 55] 的质量。质量指标在对 EMO算法 进行基准测试时发挥着至关重要的作用。具有代表性的质量指标包括超体积（HV） [54], 加性 ϵ-指标（$I_{\epsilon+}$） [55], 和反世代距离（IGD） [13]。例如，解集 X 的 IGD 值计算如下：

$$\mathrm{IGD}(\mathcal{X}) = \frac{1}{|\mathcal{S}|}\left(\sum_{\mathbf{s}\in\mathcal{S}} \min_{\mathbf{x}\in\mathcal{X}}\left\{\mathtt{distance}(\mathbf{f}(\mathbf{x}),\mathbf{s})\right\}\right), \tag{2}$$

其中，S是由IGD参考点组成的集合$\mathbf{s} \in \mathbb{R}^m$。理想情况下，S中的IGD参考点应在帕累托前沿上均匀分布。在公式（2）中，IGD衡量每个$\mathbf{s} \in$S到其最近目标向量$\mathbf{f}(\mathbf{x})$的平均距离。较小的IGD值表示X在收敛性和多样性方面都具有良好的质量。

由于IGD 并非帕累托相容，IGD 可能导致错误的结论 [24, 43]。为了解决这一问题，文献 [24]提出了一种弱帕累托相容版本的 IGD（IGD$^+$）。IGD$^+$ 使用以下距离度量，而不是公式（2）中的距离函数：距离$_{\mathrm{IGD}^+}(\mathbf{f}(\mathbf{x})$

$$, \mathbf{s}) = \sqrt{\sum_{i=1}^{m} (\max\{f_i(\mathbf{x}) - s_i, 0\})^2}.$$

传统的质量指标（例如 IGD）不会考虑由参考点 z定义的任何偏好信息。因此，传统的质量指标不适合用于对PBEMO算法进行基准测试。如 [48], 中所讨论的，PBEMO的质量指标应能够从以下三个方面评估 f(X)：*i*) 向帕累托前沿的收敛，*ii*)向参考点 z 的收敛，以及 *iii*) 感兴趣区域内的多样性。即使 f(X) 位于感兴趣区域之外，PBEMO的质量指标也应能够评估其质量。

尽管已经提出了一些用于基准测试 PBEMO算法的质量指标（例如 [2, 29, 35]），但在 [48]中推荐使用 IGD-C [36] 。IGD-C是针对公式（1）中的ROI 隐式设计的，能够处理这三个方面 *i*)– *iii*)。X 的 IGD-C 值通过公式（2）计算，使用 S′ ⊆S，而不是 S。虽然 S 中的 IGD 参考点分布在帕累托前沿上，但 S′ 中的 IGD 参考点仅分布在 ROI 上。在图1的示例中，S′ 只能位于虚线圆内。为了从S 中选择 S′，首先从S 中选择距离z 最近的点sc，即sc =最小化参数

$$_{\mathbf{s}\in} \quad \{\text{距离}(\mathbf{s},\mathbf{z})\}.\mathcal{S}$$

然后，S′ 是以sc为中心、半径为 r的超球体区域内所有点的集合，即 S′ $= \{\mathbf{s} \in \mathrm{S} \mid \text{距离}(\mathbf{s}, \mathbf{s}_c) < r\}$。

2.4 基于偏好的进化多目标优化算法 s

正如 [4], 所述，文献中已提出一些 PBEMO 算法。与 [30], 一样，本文关注以下六种 PBEMO 算法：R-NSGA-II [19], r-NSGA-II [41], g-NSGA-II [37], PBEA [49], R-MEAD2 [36], 和 MOEA/D-NUMS [28]。需要注意的是，这六种 PBEMO 算法均未使用不确定性分析。

R-NSGA-II、r-NSGA-II 和 g-NSGA-II是用于 [16] 基于偏好的多目标优化的 NSGA-II 扩展版本。R- NSGA-II 计算到参考点z 的距离，而不是拥挤距离。因此，R-NSGA-II偏好接近 z 的非支配个体。R-NSGA-II 还执行 ϵ-清除操作，以保持种群的多样性。r-NSGA-II 使用 r-支配关系，而不是帕累托支配关系。r-支配关系偏好接近 z 的个体，其偏好区域的大小由参数 $\delta \in [0, 1]$决定。类似地，g-NSGA-II 使用g-支配关系，而不是帕累托支配关系。g-支配关系偏好区域中的非支配个体，该区域是 $\mathbb{R}^m$ 其中所有支配 z 或被z 支配的目标向量的集合。

PBEA [49] 是 IBEA [52] 的扩展版本，它将二元加性 ϵ-指标（$I_{\epsilon+}$）替换为基于偏好的指标。PBEA 中的指标通过使用增强型成就标量化函数 [34]来处理决策者的偏好信息。

R-MEAD2 和 MOEA/D-NUMS 是基于标量化函数的 PBEMO算法，使用一组 μ 权重向量 W $= \{\mathbf{w}_i\}_{i=1}^{\mu}$。尽管 R-MEAD2 与 MOEA/D [51], 相似，但 R-MEAD2 会自适应地调整 μ 权重向量，使 μ 个体在目标空间中向z 移动。先前的研究 [28] 提出了一种非均匀映射方案（NUMS），使 μ 权重向量向z移动。与 R-MEAD2 不同，NUMS 以离线方式调整权重向量。MOEA/D-NUMS [28] 是使用由 NUMS 生成的 W 扩展而来的 MOEA/D。MOEA/D-NUMS 还使用增广成就标量化函数 [34]。

2.5 子集选择方法

如第 1 节所述，需要一种后处理方法，从不确定性分析 A 中选出 k 代表性解。这正是子集选择问题，其目标是从 A 中找到大小为 k 的子集 X [3]。由于子集选择问题通常是 NP 难的 [7], , 采用非精确方法（例如贪心搜索和局部搜索）是合理的首选。然而，计算成本与子集质量之间存在权衡。

Shang 等人 [45] 研究了若干基于距离的子集选择（DSS）方法的性能，这些方法旨在目标空间中找到均匀分布的解的子集。他们的结果表明，迭代式 DSS（IDSS）方法能够以较低的计算成本获得比其他方法更好的子集。IDSS 方法旨在找到大小为 k 的子集 X，使以下均匀性水平最大化 [42]。

$$\mathtt{uniformity}(\mathcal{X}) = \min_{\mathbf{x}_1\in\mathcal{X},\mathbf{x}_2\in\mathcal{X}\setminus\{\mathbf{x}_1\}} \mathtt{distance}(\mathbf{f}'(\mathbf{x}_1), \mathbf{f}'(\mathbf{x}_2)),$$

其中，$\mathbf{f}'(\mathbf{x})$ 是根据 f(X) 的最大值和最小值对 x的目标向量进行归一化后的结果。

算法1： IDSS [45]

1 **如果** $|\mathcal{A}| \leq k$ **则**
2 　**返回** $\mathcal{A}$;
3 **否则**
4 　X 从 A 中 ← 随机选择 k 个解;
5 　**对于** $t \in \{1, \ldots, t_{\text{最大值}}\}$ **执行**
6 　　$\mathbf{x}_{\text{新}}$ ← 从 A \ X 中随机选择一个解;
7 　　$X \leftarrow X \cup \{\mathbf{x}_{\text{新}}\}$;
8 　　$\mathbf{x}^{w} \leftarrow \underset{\mathbf{x} \in \mathcal{X}}{\text{最大值argmax}} \{\text{均匀性}(X \setminus \{\mathbf{x}\})\}$;
9 　　$X \leftarrow X \setminus \{\mathbf{x}_{w}\}$;
10 　**返回** $\mathcal{X}$;

算法1展示了IDSS方法。A 是仅包含非支配解的不确定性分析集。如果 A 的大小小于或等于 k，算法1将返回 A（第1–2行）。否则，将从 A 中随机选取 k 个解并将其设为 X（第4行）。随后，执行以下步骤 $t_{\max}$ 次。首先，从 A \ X 中随机选取新解 $\mathbf{x}_{\text{new}}$（第6行），并将 $\mathbf{x}_{\text{new}}$ 添加到 X 中（第7行）。然后，IDSS方法移除对 X 的均匀性水平贡献最差的解 $\mathbf{x}_{w}$（第8–9行）。如果 $t_{\max}$ 足够大，IDSS 很可能获得具有均匀分布的 $\mathbf{f}(X)$。

3 提出的后处理方法

本节介绍一种针对 PBEMO 中 UA 的基于偏好的后处理方法。设 A 为 UA，其中包含截至目前由 PBEMO 算法找到的所有非支配解。这里的目标是在目标空间的 ROI 中找到一个大小为 k 且呈均匀分布的子集 X。我们认为，该目标涵盖了 PBEMO 质量指标所要求的三个方面 *i*)– *iii*)（见第 2.3 节）。然而，现有的后处理方法（例如 IDSS方法 [45]）不太可能实现这一目标。这是因为它们旨在目标空间的整个 *PF* 上找到呈均匀分布的 X。为解决这一问题，我们提出一种能够处理决策者偏好信息的基于偏好的后处理方法。需要注意的是，所提出的方法针对公式（1）定义的 ROI 进行设计。此外，该方法甚至可以被纳入任何传统的多目标进化优化算法中。

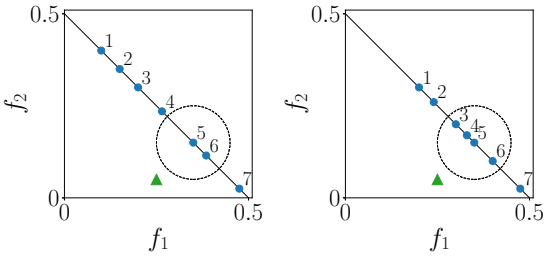
算法2展示了所提出的方法。与算法1一样，如果 $|A| \leq k$，算法2返回 A（第1–2行）。否则，执行以下步骤。首先，所提出的方法对现实世界问题中未知的感兴趣区域进行近似。在目标空间中，距离参考点 $\mathbf{x}_c$ 最近的解从 A 中选出（第4行）。我们将目标空间中以 $\mathbf{x}_c$ 为中心、半径为 r 的 an m维超球体内所有非支配解的集合表示为近似感兴趣区域。在公式（1）中，如果

一个 = X*。近似感兴趣区域中的所有非支配解都被设为 X（第 5 行）。如果 X 的大小恰好与 k 相同，算法2 将返回 X（第 11 行）。否则，考虑以下两种情况：

(1) X的大小小于 k（第 6–8 行），以及 (2)
X 的大小大于 k（第 9–10 行）。

算法2： 所提出的基于偏好的后处理-
用于 PBEMO 的后处理方法

1 **如果** $|\mathcal{A}| \leq k$ **则**
2 　**返回** $\mathcal{A}$;
3 **否则**
4 　$\mathbf{x}^{c} \leftarrow \underset{\mathbf{x} \in \mathcal{A}}{\text{最小化参数}} \{\text{距离}(\mathbf{f}(\mathbf{x}), \mathbf{z})\}$;
5 　$X \leftarrow \{\mathbf{x} \in A \mid \text{距离}(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}_c)) \leq r\}$;
6 　**当** $|\mathcal{X}| < k$ **时执行**
7 　　$\mathbf{x}^{c'} \leftarrow \underset{\mathbf{x} \in A \setminus X}{\text{最小化参数}} \{\text{距离}(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}_c))\}$;
8 　　$X \leftarrow X \cup \{\mathbf{x}_c{}'\}$;
9 　**如果** $|\mathcal{X}| > k$ **则**
10 　　X ← subset_选择(X k);
11 　**return** $\mathcal{X}$;



(a) 情况 (1)： $|\mathcal{X}| < k$　　**(b) 情况 (2)：** $|\mathcal{X}| > k$

图 2：DTLZ1 上的示例，其中 $m = 2$，其中 $|\mathcal{A}| = 7$ 和 $k = 3$。▲ 是参考点。

图 2 展示了这两种情况的示例，其中 $|A| = 7$ 和 $k = 3$。在我们的初步结果中，我们观察到情况 (1) 出现在搜索的早期阶段，而情况 (2) 出现在较晚阶段。

在情况（1）中，我们认为，尽管这些解位于近似感兴趣区域之外，但决策者仍然希望在目标空间中获得更接近xc的解。因此，从A\X中选择 $k - |X|$ 个非支配解是合理的。当 $|X| < k$时，重复执行以下步骤。首先，从A\ X 中选择目标空间内距离xc′ 最近的解 xc（第7行）。然后，将xc′ 添加到X中（第8行）。在图 2(a) 的示例中，X = {x5, x6}，且x5是 xc。在这种情况下，在目标空间中，x4 比 x1、x2、x3 和 x7 都更接近 x5。因此，将 x4 添加到 X 中。

在情况（2）中，应从 X 中移除 $|X| - k$ 个解，同时保持 X 的多样性。幸运的是，可以使用传统的子集选择方法从 X中选择 k 个解（第10行）。基于 [44]，中报告的有希望的结果，本文采用算法1所示的 IDSS方法。需要注意的是，IDSS 无法处理决策者的偏好信息，但 X 中解的目标向量已经位于近似感兴趣区域内。对 X应用 IDSS后，预计X中 k 个解的目标向量将在近似感兴趣区域内均匀分布。在图2(b) 的示例中，X = $\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$，以及 $k = 3$。在这种情况下，执行 IDSS 后，$\mathbf{x}_4$ 很可能会从 X 中被移除。

4 实验设置

这里，我们介绍分析所用的实验设置。维护不确定性分析（UA）需要一种非支配排序方法。我们使用了基于 ND-Tree 的更新方法 [27],，这是目前最先进的方法之一。我们使用了 [27] 的作者提供的基于 ND-Tree 的更新方法的 C++ 实现。与 [47], 一样，我们每隔 100、200、.....、1 000、2 000、.....、50 000 次函数评估更新一次存档。我们还用 C++ 实现了所提出的基于偏好的后处理方法，包括 IDSS 方法 [45]。除非另有说明，我们分别将子集 k 的大小和感兴趣区域（ROI）r 的半径设为 100 和 0.1。与 [45], 一样，我们将 IDSS 中的最大迭代次数 $t_{\max}$ 设为 10_4。

我们使用了第 2.4 节所述的六种 PBEMO算法。我们使用了 [30] 的作者提供的 PBEMO算法源代码。每种 PBEMO算法均进行了 31 次独立运行。我们将种群规模 μ 设为 8、20、40、100、200、300、400和 500。我们根据 [30] 设置了六种 PBEMO算法的其他参数。我们通过将其他参数固定为默认值，考察种群规模的影响。对所有参数（例如 R-NSGA-II 中的 ϵ）进行相关性分析超出了本文的范围。我们使用 MOEA/D-NUMS 作者提供的源代码生成权重向量集。我们还使用 [6], 提出的方法生成原始权重向量集，该方法可以生成任意大小的权重向量集。

如 [2], 所述，我们使用了 DTLZ1–DTLZ4 问题 [20]。正如 [46], 中 EMO算法的结果所显示的，我们认为，在具有不规则帕累托前沿的问题（例如 DTLZ5 和 DTLZ7）上，PBEMO算法需要较大的种群规模。对于目标具有不同尺度的问题（例如 SDTLZ1 [17]）的研究将留待未来开展。我们将目标数量 m 设置为：$m \in \{2, 3, 4, 5, 6\}$。决策变量的数量按照[20]进行设置。我们将参考点 z 设置如下：对于 $m = 2$，$\mathrm{z} = (0.6, 0.4)^\top$，$\mathrm{z} = (0.5, 0.3, 0.2)^\top$，$\mathrm{z} = (0.4, 0.3, 0.2, 0.1)^\top$，$\mathrm{z} = (0.3, 0.25, 0.2, 0.15, 0.1)^\top$，以及 $\mathrm{z} = (0.3, 0.2, 0.15, 0.13, 0.12, 0.1)^\top$，分别对应 2，.....，6。大多数参考点都是可达的，但 PBEMO算法（g-NSGA-II 除外）的行为对 z 的位置并不十分敏感，除非 z 距离帕累托前沿过远 [30, 48]。

尽管 [48], 中推荐了 IGD-C 指标，但 IGD-C 并非帕累托相容。为解决这一问题，我们将 IGD-C 中的 IGD 计算替换为 IGD^+ 计算。我们将此版本的 IGD-C 记作 IGD^+-C。就帕累托相容性而言，IGD^+-C 比 IGD-C 更可靠。

5 结果

本节介绍我们的分析结果。通过实验，第 5.1–5.3 节分别旨在回答第 1 节中提出的三个研究问题（**研究问题1– 研究问题3**）。

5.1 所提出的后处理方法的有效性

本节考虑由每种 PBEMO 算法找到的以下三个解集：

- **POP**：大小为 μ 的最终种群，
- **UA-IDDS**：通过 IDSS 方法对不确定性分析进行后处理后得到的大小为 k 的解子集，以及
- **UA-PP**：通过基于偏好的后处理（PP）方法对不确定性分析进行后处理后得到的大小为 k 的解子集。

我们重点关注具有 $\mu = 100$，即 $\mu = k$的PBEMO算法的结果。以往大多数研究都是基于其POP评估PBEMO算法的性能。相比之下，这是首项基于其UA-IDDS和UA-PP评估PBEMO算法性能的研究。

5.1.1 目标向量的分布。图3展示了双目标 DTLZ2 问题中 POP、UA-IDDS 和 UA-PP 内解的目标向量分布。图3展示了每种 PBEMO 算法单次运行的结果。图S.1–S.3分别展示了 DTLZ1、DTLZ3 和 DTLZ4 问题上的结果。**POP 的结果**。如图3(a)、(d)、(g)、(j)、(m)和(p)所示，六种 PBEMO 算法最终种群中的部分解的目标向量位于感兴趣区域附近或其中。然而，对于 R-NSGA-II、r-NSGA-II、R-MEAD2 和 MOEA/D-NUMS，最终种群在感兴趣区域内的多样性较差。这是因为它们都没有保留迄今为止最优的非支配解。尽管 PBEA 和 g-NSGA-II 得到的目标向量覆盖了目标空间中的感兴趣区域，但它们并未适配该区域。此处，g-NSGA-II旨在逼近一个不同于方程（1）的感兴趣区域 [48]。

UA-IDDS的结果。如图3(b)、(e)、(h)、(k)、(n)和(q)所示，将 IDDS方法应用于UA所发现的解子集的目标向量覆盖了感兴趣区域。然而，它们的分布过于宽泛。直观地说，PBEMO算法很可能只在目标空间中感兴趣区域附近生成解。相比之下，我们的结果表明，六种PBEMO算法能够生成远离感兴趣区域的非支配解。这可能是因为这六种PBEMO算法没有明确采用交配限制。

UA-PP 的结果。图 3(c)、(f)、(i)、(l)、(o) 和 (r) 彼此相似。将所提出的方法应用于 UA 后得到的解子集中的目标向量，在感兴趣区域内分布密集。与上述两种情况下讨论的结果相比，解子集中的目标向量似乎更贴合感兴趣区域。在图 S.1 和 S.2 所示的 DTLZ1 和 DTLZ3 问题中，我们也观察到了类似结果；在这些图中，一些 PBEMO算法未能逼近 DTLZ4 问题的感兴趣区域。

5.1.2 使用 *IGD* 的比较$^+$*-C.* 表 2 显示了三个解子集在 DTLZ1–DTLZ4 问题上经过 31 次运行得到的平均 IGD^+-C 值。表 2 展示了 R-NSGA-II 的结果。表 S.1 展示了其他 PBEMO算法的结果。在表 2 中，最佳和次佳数据分别以深灰色和灰色突出显示。我们来比较两个解集 X_1 和 X_2（例如，POP 和 UA-PP）在 31 次运行中的表现。表 2 中括号内的符号表示，根据带有 $p < 0.05$的 Wilcoxon秩和检验，在 IGD^+-C 指标上，X_1 的表现显著优于（+）或显著劣于（–）X_2。符号 $\approx$ 表示两项结果之间没有显著差异。我们比较了 1）UA-DDS 与 POP，2）UA-PP 与 POP，以及 3）UA-PP 与 UA-IDDS。例如，从表 2 可以看出，在 DTLZ1 上，UA-IDDS 的表现显著劣于 POP，$m = 4$。在 DTLZ1 上，UA-PP 的表现显著优于 POP，$m = 3$，但 UA-PP 与 UA-IDDS 之间没有显著差异。</final>

如表 2 所示，UA-IDDS 在许多情况下优于 POP。然而，在某些情况下，UA-IDDS 的表现显著差于 POP。

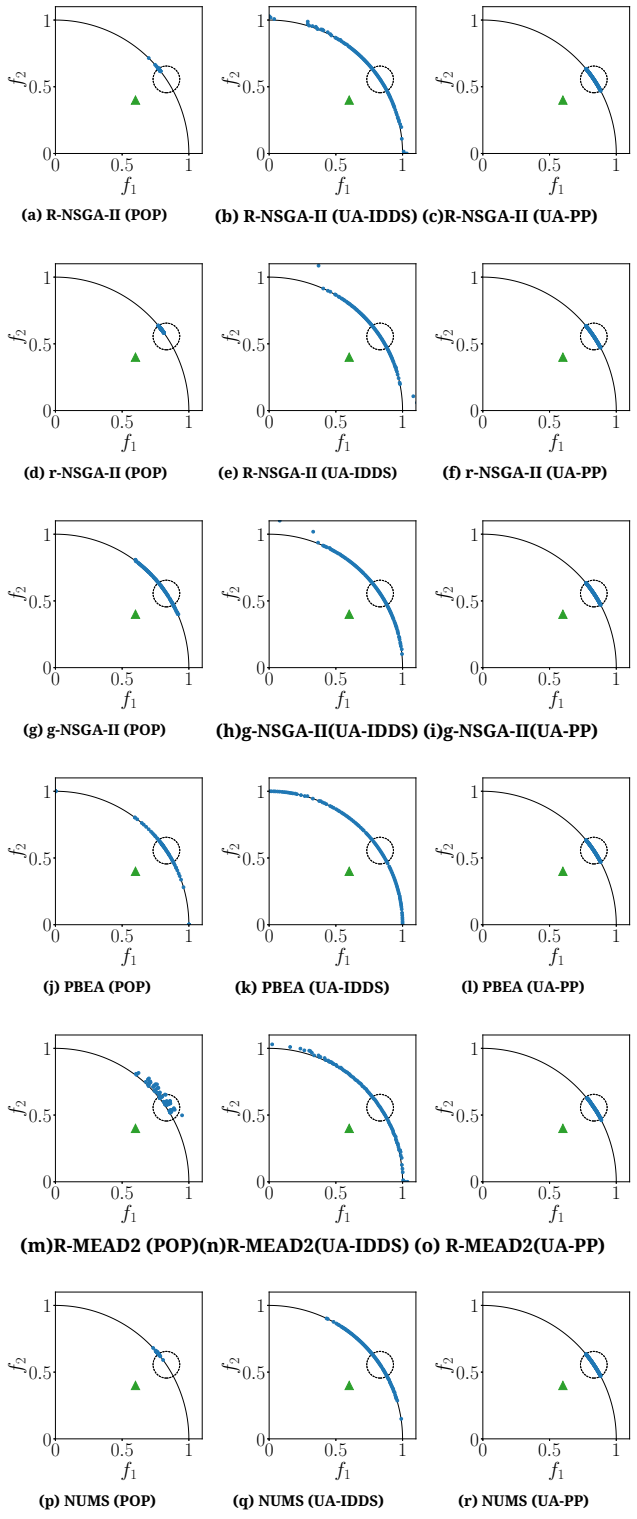

(a) R-NSGA-II (POP)
(b) R-NSGA-II (UA-IDDS)
(c)R-NSGA-II (UA-PP)
(d) r-NSGA-II (POP)
(e) R-NSGA-II (UA-IDDS)
(f) r-NSGA-II (UA-PP)
(g) g-NSGA-II (POP)
(h)g-NSGA-II(UA-IDDS)
(i)g-NSGA-II(UA-PP)
(j) PBEA (POP)
(k) PBEA (UA-IDDS)
(l) PBEA (UA-PP)
(m)R-MEAD2 (POP)
(n)R-MEAD2(UA-IDDS)
(o) R-MEAD2(UA-PP)
(p) NUMS (POP)
(q) NUMS (UA-IDDS)
(r) NUMS (UA-PP)

图3：DTLZ2 上三个解集中解的目标向量分布，其中 $m = 2$，其中▲ 为参考点 z。虚线圆表示真实感兴趣区域。"NUMS"代表 MOEA/D-NUMS。

表2：R-NSGA-II在DTLZ1–DTLZ4 问题上找到的三个解子集的平均 IGD⁺-C 值。

问题	m	POP	UA-IDDS	UA-PP
DTLZ1	2	0.0236	0.0018 (+)	0.0012 (+, +)
	3	0.0334	0.0211 (+)	0.0220 (+, ≈)
	4	0.0562	0.0743 (–)	0.0442 (+, +)
	5	0.0933	0.0782 (+)	0.0558 (+, +)
	6	0.1131	0.0779 (+)	0.0695 (+, +)
DTLZ2	2	0.0411	0.0016 (+)	0.0004 (+, +)
	3	0.1247	0.0276 (+)	0.0114 (+, +)
	4	0.1986	0.0720 (+)	0.0339 (+, +)
	5	0.2729	0.1162 (+)	0.0600 (+, +)
	6	0.2840	0.1540 (+)	0.0853 (+, +)
DTLZ3	2	0.0345	0.0083 (+)	0.0078 (+, ≈)
	3	0.1083	0.0460 (+)	0.0309 (+, +)
	4	0.1988	0.1736 (≈)	0.0636 (+, +)
	5	0.2370	0.2169 (≈)	0.1024 (+, +)
	6	0.8749	0.8287 (+)	0.7224 (+, +)
DTLZ4	2	0.1014	0.0829 (≈)	0.0818 (≈, ≈)
	3	0.0838	0.0528 (+)	0.0375 (+, +)
	4	0.1030	0.0741 (+)	0.0465 (+, +)
	5	0.3757	0.1206 (+)	0.0759 (+, +)
	6	0.3214	0.1144 (+)	0.0655 (+, +)

问题，例如带有 $m = 4$的DTLZ1。其他五种PBEMO算法的结果中也可以发现类似结果，如表S.1所示。这是因为UA-IDDS中的目标向量在帕累托前沿上广泛分布，如图3所示。

相比之下，在大多数情况下，UA-PP 的表现显著优于 POP 和 UA-IDDS。UA-PP 的表现不逊于 POP。这些结果表明，UA 在 PBEMO 中的有效性，以及在对 UA 进行后处理时处理决策者偏好信息的重要性。

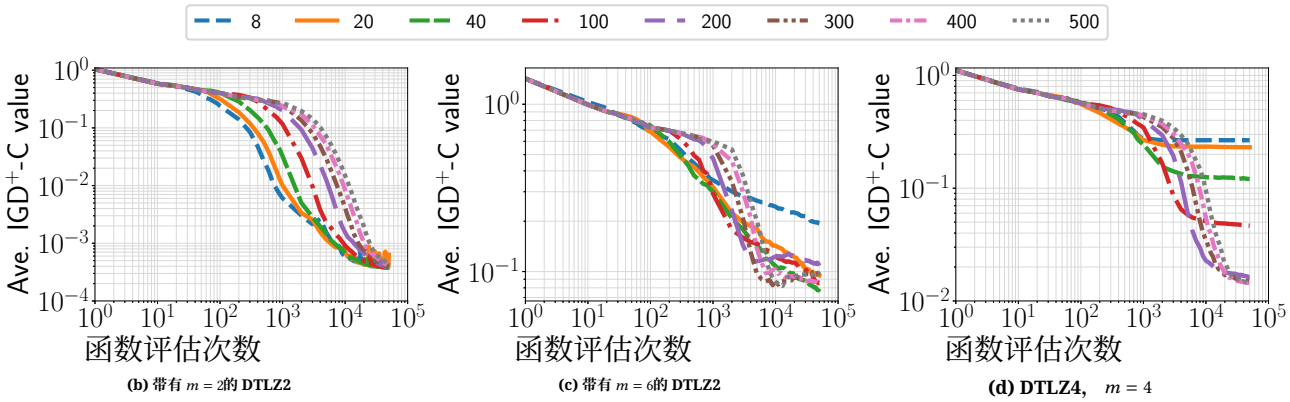
RQ1 的回答

基于 IGD⁺-C 得出的结果表明，使用不确定性分析可以显著提升六种 PBEMO 算法的性能。我们观察到，对不确定性分析应用所提出的基于偏好的后处理方法，可以找到更优的解子集。这一观察结果表明，在不确定性分析的后处理中纳入决策者的偏好信息十分重要。

5.2 种群规模的影响

基于第5节中在 DTLZ1 和 DTLZ4 问题上，针对六种 PBEMO 算法分别使用种群规模 $\mu \in \{8, \ldots\ldots, 500\}$ 进行了 31 次独立运行。随后，对于每种 PBEMO 算法，我们通过 Friedman 检验 [21] 计算种群规模 $\mu \in \{8, \ldots\ldots, 500\}$ 的平均排名。我们使用 CONTROLTEST 软件包（https://sci2s.ugr.es/sicidm）根据 IGD⁺-C 值计算排名。下文将讨论不同目标数量 m 和函数评估预算下排名最高的种群规模。

表3展示了在所有 DTLZ1–DTLZ4 问题上，针对 $m \in \{2, 4, 6\}$ 和 10^3，5×10^3，10^4，3×10^4，5×10^4 次函数评估时的最佳种群规模 μ



(b) 带有 $m = 2$的 DTLZ2　(c) 带有 $m = 6$的 DTLZ2　(d) DTLZ4，$m = 4$

图4：R-NSGA-II 在三个问题上采用不同种群规模时的平均 IGD⁺-C 值。

表3：DTLZ1–DTLZ4 问题中每个函数评估次数对应的最佳种群规模 μ 和 $m \in \{2,4,6\}$。

(a) $m = 2$

PBEMO	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
R-NSGA-II	8	20	8	8	200
r-NSGA-II	20	20	20	40	40
g-NSGA-II	20	40	40	200	100
PBEA	20	40	8	100	200
R-MEAD2	8	8	8	20	20
MOEA/D-NUMS	8	8	8	20	20

(b) $m = 4$

PBEMO	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
R-NSGA-II	8	40	20	20	20
r-NSGA-II	40	40	40	40	40
g-NSGA-II	40	100	200	400	400
PBEA	20	100	40	100	200
R-MEAD2	20	100	100	100	20
MOEA/D-NUMS	8	20	20	20	20

(c) $m = 6$

PBEMO	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
R-NSGA-II	20	40	100	40	100
r-NSGA-II	40	40	100	100	100
g-NSGA-II	8	8	300	8	8
PBEA	20	40	100	300	300
R-MEAD2	8	100	200	20	40
MOEA/D-NUMS	20	40	40	20	20

DTLZ1–DTLZ4 问题。如表3所示，$\mu = 500$ 在任何情况下都不是最佳选择。表S.2还展示了八种 μ 值的排名。如表S.2所示，在大多数情况下，$\mu = 500$的排名最差。尽管此前的一些研究采用了相对较大的种群规模，如表1所示，但这一结果表明，对于 PBEMO 算法，采用小于常用规模的种群规模是有效的。由于我们采用了不确定性分析，六种 PBEMO 算法无需维持适用于决策的优良解。这可能是原因之一。

本研究中六个PBEMO算法不需要较大种群规模的原因。

如表3(a)所示，最佳种群规模取决于可用的函数评估预算。例如，对于 R- NSGA-II，$\mu = 8$ 和$\mu = 200$ 分别在 10_3 次和 5×10^4 次函数评估时表现最佳。对于多目标情形，这一特性在 PBEA 的结果中更加明显。例如，如表3(c)中的 PBEA 结果所示，对于 $m = 6$，$\mu = 20$ 和 $\mu = 300$ 分别在 10_3 次和 5×10^4 次函数评估时表现最佳。这些结果与先前研究的结论一致；这些研究考察了种群规模对 EMO算法的影响（例如，[10, 22, 26, 46]）。

相比之下，我们在一些PBEMO算法中观察到了意料之外的结果。例如，如表3(c)所示，对于 $m = 6$ 和 10_3 次函数评估，$\mu = 8$ 适用于 R-MEAD2。对于 10_4 次函数评估，$\mu = 200$ 最适用于 R-MEAD2。然而，对于 3×10^4 和 5×10^4 次函数评估，$\mu = 20$ 和 40 又分别适用于 R-MEAD2。

下面，我们讨论各测试问题上的结果。图4展示了在 DTLZ2（$m \in \{2,6\}$）和 DTLZ4（$m = 4$）上，采用不同种群规模的 R-NSGA-II 的平均 IGD⁺-C 值。图S.4–S.9展示了六种 PBEMO算法在所有测试问题上的结果。由于篇幅限制，我们不讨论图S.4–S.9，但其结果与图4相似。

如图4(a)所示，在函数评估次数较少时，种群规模最小（$\mu = 8$）的R-NSGA-II在 $m = 2$ 的DTLZ2上表现最佳。然而，随着函数评估次数增加，种群规模为 $\mu = 8$ 的R-NSGA-II表现不佳。如图4(b)所示，对于多目标问题，这一现象更加明显。如图4(b)所示，随着函数评估次数增加，R-NSGA-II在某些 μ 下的平均IGD⁺-C值并未单调下降。这或许是种群规模相对较大（例如，$\mu \in \{200, 300\}$）的R-NSGA-II在10_4 次函数评估中表现良好的原因之一，如上文所述。使用基于指标的子集选择方法 [3] 代替IDDS方法，或许可以使IGD⁺-C值单调下降。然而，基于指标的子集选择方法通常比基于距离的子集选择方法（例如IDDS方法）需要更高的计算成本。

进一步的研究是另一项未来工作。如图4(c)所示，随着 $m = 4$ 搜索的推进，种群规模较小的 R-NSGA-II 在 DTLZ4 上表现不佳。这一观察表明，对于解的密度分布存在偏差的问题，PBEMO算法需要较大的种群规模。

RQ2 的答案

我们的结果表明，即使面对多目标，在函数评估预算较小的情况下，六种PBEMO算法采用小于通常使用的种群规模也是有效的。我们观察到，随着搜索的推进，PBEMO算法需要较大的种群规模。然而，我们的结果表明，一般性的经验法则"在函数评估预算较大的多目标优化问题中使用较大的种群规模"可能并不能很好地帮助确定PBEMO中的种群规模。当然需要注意的是，最佳种群规模还取决于PBEMO算法的类型。

5.3 ROI 的大小如何影响最佳种群规模

在第5.1节和第5.2节中，我们将感兴趣区域 r 的半径设为0.1。感兴趣区域的大小对PBEMO算法性能的影响也知之甚少。直观地说，PBEMO算法需要较大的种群规模来覆盖帕累托前沿的较大子区域。因此，最佳种群规模很可能取决于 r。

表4显示了 R-NSGA-II 在 DTLZ1–DTLZ4 问题上、取值为 $r \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$ 时的最佳种群规模 μ，其中 $r = 0.1$ 的结果与表3相同。与表3类似，我们根据 Friedman 检验的平均排名确定最佳种群规模。表 S.3–S.7 展示了其他 PBEMO算法的结果，但这些结果与表4类似。

如表4所示，对于 $r \in \{0.01, 0.05, 0.1\}$，最佳的 μ 值几乎相同。此外，对于函数评估预算较小的情况，较小的种群规模是最佳选择。相比之下，从表4中 5×10^4 函数评估的结果可以看出，随着 r 增加，最佳种群规模也会变大。然而，我们认为 $r = 0.3$ 作为感兴趣区域的规模过大。回想一下，在图1的示例中，r 被设为0.1。如果 $r = 0.3$，图1中的感兴趣区域几乎与帕累托前沿相同。在基于偏好的多目标优化中，决策者不太可能希望得到覆盖目标空间中整个帕累托前沿的解集（见第2.2节）。

研究问题3的答案

我们的结果表明，当 r、m以及函数评估预算较大时，六种PBEMO算法需要较大的种群规模。然而，决策者不太可能偏好较大的 r 值。对于一些现实世界问题 [11]，可用的函数评估预算也十分有限。因此，我们认为，在实践中，种群规模相对较小是一个合理的首选。

6 结论

本文首先提出了用于对 PBEMO中的 UA 进行后处理的基于偏好的方法。与现有的后处理方法不同，所提出的方法能够处理决策者的偏好

表4：R-NSGA-II 在 DTLZ1– DTLZ4 问题上的最佳种群规模 μ，其中 $r \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$。

(a) $m = 2$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	8	20	8	20	20
0.05	8	20	8	20	20
0.1	8	20	8	8	200
0.2	20	20	8	100	200
0.3	20	40	8	100	300

(b) $m = 4$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	8	40	20	20	40
0.05	8	40	20	40	40
0.1	8	40	20	20	20
0.2	8	40	200	40	40
0.3	8	40	200	20	20

(c) $m = 6$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	20	40	300	40	40
0.05	20	40	300	40	40
0.1	20	40	100	40	100
0.2	20	100	100	200	300
0.3	20	100	100	200	300

由参考点 z 定义的信息。随后，我们研究了所提出的基于偏好的后处理方法在 DTLZ 问题上的有效性 [20]，目标数量为 $m \in \{2, \ldots\ldots, 6\}$。结果表明，在大多数情况下，所提出的方法能够从不确定性分析中获得比 IDDS方法更好的子集 [45] 。结果还表明，使用不确定性分析和所提出的方法，能够显著提升六种 PBEMO算法（R-NSGA-II [19], r-NSGA-II [41], g-NSGA-II [37], PBEA [49], R-MEAD2 [36], 和 MOEA/D-NUMS [28]）的性能。我们的结果揭示了在 PBEMO算法中选择合适种群规模的难度。尽管最佳种群规模取决于一些因素，但我们的结果表明，对于PBEMO算法而言，较小的种群规模（例如 $\mu \in \{8, 20, 40\}$）是一个不错的首选，即使面对多目标问题也是如此。

尽管基准测试方法已在进化多目标优化文献中得到研究（例如， [9, 10, 25]），但在PBEMO文献中却很少受到关注。正如 [1], 中所述，对PBEMO算法进行基准测试是一项具有挑战性的任务。我们的研究结果有助于建立一种用于对PBEMO算法进行基准测试的系统方法。本文仅关注使用单个参考点的PBEMO算法。未来工作需要研究多个参考点的情形。对使用其他偏好表达（例如，价值函数）的PBEMO算法进行分析是未来工作的另一个主题。

致谢

本研究得到 JSPS KAKENHI 资助，项目编号为 21 17 824 K，以及日本文部科学省 LEADER 项目的支持。

参考文献

[1] Bekir Afsar、KaisaMiettinen 和 Francisco Ruiz。2021。评估交互式多目标优化方法的性能：一项调查。 *ACM Comput. Surv.* 54, 4 2021 85 1 85 27 10 1145 3448301 ( ), ： –: 。 https://doi.org/ 。

[2] / SunithBandaru 和 Henrik Smedberg。2019。。一种基于参考点的多目标进化算法的无参数性能指标。载于 遗传与进化计算会议（*GECCO*），AnneAuger 和 Thomas Stützle（编）。ACM, 499 506 10 1145 3321707 3321757 – 。 https://doi.org/ 。

[3] /。 Matthieu Basseur 、 Bilel Derbel 、 AdrienGoëffon 和 Arnaud Liefooghe。2016。关于贪心和局部搜索启发式方法用于*d*维超体积子集选择的实验。载于 *2016*年遗传与进化计算会议论文集，美国科罗拉多州丹佛，*2016* 年 *7* 月 *20* 日至 *24* 日，Tobias Friedrich、Frank Neumann 和Andrew M. Sutton（编）。ACM，541–548。https://doi.org/ 10 1145 2908812 2908949 。 / 。

[4] Slim Bechikh、Marouane Kessentini、Lamjed Ben Said 和 KhaledGhédira。2015。第四章——进化多目标优化中的偏好融合：现状调查。 *Adv. Comput.* 98（2015），141–207。https://doi.org/ 10 1016 2015 03 001 。 /bs.adcom. 。 。

[5] Leonardo C. T. Bezerra、Manuel López-Ibáñez和 Thomas Stützle。2019。存档器对先进多目标和多目标进化算法性能的影响。载于 遗传与进化计算会议论文集，*GECCO 2019*，捷克共和国布拉格，*2019* 年 *7* 月 *13* 日至 *17* 日，AnneAuger 和 Thomas Stützle（编）。ACM, 620 628 10 1145 3321707 3321789 – 。

[6] https://doi.org/ 。/ 。 Julian Blank、Kalyanmoy Deb、Yashesh D. Dhebar、SunithBandaru 和 HaithamSeada。2021。在进化多目标优化中生成单位单纯形上的均匀间隔点。*IEEE Trans. Evol. Comput.* 25，1（2021），48–60。https://doi.org/ 10 1109 2020 2992387 。 /TEVC。 。

[7] Karl Bringmann、Sergio Cabello和Michael T.M. Emmerich。2017。锚定盒的最大体积子集选择。载于 第*33*届国际计算几何研讨会，*SoCG2017*，*2017* 年 *7* 月*4* 日至 *7* 日，澳大利亚布里斯班（*LIPIcs*，第 *77* 卷），Boris Aronov 和 Matthew J.Katz（编）。SchlossDagstuhl - Leibniz信息学中心, 22 1 22 15 10 4230 2017 22 : –: 。 https://doi.org/ 。/LIPIcs.SoCG。 。

[8] Karl Bringmann、Tobias Friedrich 和 PatrickKlitzke。。通过子集选择对超体积和 Epsilon指标进行通用后处理。载于 自然并行问题求解——*PPSNXIII*——第 *13* 届国际会议，斯洛文尼亚卢布尔雅那，*2014* 年 *9* 月*13* 日至 *17*日。论文集（计算机科学讲义，第 *8672* 卷），Thomas Bartz-Beielstein、Jürgen Branke、BogdanFilipic 和 Jim Smith（编）。Springer, 518 527 10 1007 978 3 319 10762 2_51 – 。 https://doi.org/ 。

[9] / -- - - Dimo Brockhoff 、 Anne Auger 、 NikolausHansen 和 Tea Tusar。2022。。在多目标黑箱优化测试套件中使用易于理解的单目标函数。 *Evol. Comput.* 30，2（2022），165–193。https://doi.org/10.1162/evco_a_00298[10] Dimo Brockhoff、Thanh-DoTran 和 NikolausHansen。2015。重新审视数值多目标优化器的基准测试。载于 遗传与进化计算会议论文集，*GECCO 2015*，西班牙马德里，*2015* 年 *7* 月 *11* 日至 *15* 日，SaraSilva和 Anna IsabelEsparcia-Alcázar（编）。ACM，639–646。https://doi.org/ 10 1145 2739480 2754777 。 /。

[11] Tinkle Chugh、Karthik Sindhya、Jussi Hakanen 和Kaisa Miettinen。2019。使用进化算法处理计算代价高昂的多目标优化问题的调查。 *Soft Comput.* 23，9（2019），3137–3166。https://doi.org/ 10 1007 00500 017 2965 0 。 /s - -- CarlosA.Coello Coello。 。处理进化多目标优化中的偏好：一项调查。2000。载于 *2000*年进化计算大会论文集，*CEC 2000*，美国加利福尼亚州拉霍亚，*2000* 年*7* 月 *16* 日至*19* 日，Ali M. S. Zalzala（编）。IEEE, 30 37 10 1109 2000 870272 – 。 https://doi.org/ 。 /CEC。 。

[13] Carlos A. CoelloCoello和MargaritaReyes Sierra。2004。协同进化多目标进化算法并行化研究。载于 墨西哥国际人工智能会议（*MICAI*）。688–697。https://doi.org/ 10 1007 978 3 540 24694 7_71 。 / - - - - David Corne、Joshua D. Knowles 和 MartinJ. Oates。2000。用于多目标优化的 Pareto 包络基选择算法。载于 自然并行问题求解——*PPSN VI*，第 *6* 届国际会议，法国巴黎，*2000* 年 *9* 月*18* 日至 *20*日，论文集（计算机科学讲义，第*1917* 卷），Marc Schoenauer、Kalyanmoy Deb、Günter Rudolph、Xin Yao、Evelyne Lutton、Juan Julián Merelo Guervós 和 Hans-Paul Schwefel（编）。Springer，839–848。https://doi.org/ 10 1007 3 540 45356 3_82 。 / - - - Kalyanmoy Deb。 。 使用进化算法进行多目标优化。Wiley。2001。[16] Kalyanmoy Deb、Samir Agrawal、AmritPratap 和T. Meyarivan。2002。一种快速且具有精英保留机制的多目标遗传算法：NSGA-II。*IEEE Trans. Evol. Comput.* 6, 2 2002 182 197 10 1109 4235 996017 ( ), – 。 https://doi.org/ 。 /。

[17] KalyanmoyDeb 和 Himanshu Jain。2014。。一种使用基于参考点的非支配排序方法的进化多目标优化算法，第一部分：求解具有盒约束的问题。*IEEE Trans. Evol. Comput.* 18, 4 2014 577 601 10 1109 2013 2281535 ( ), – 。 https://doi.org/ 。 /TEVC。 。

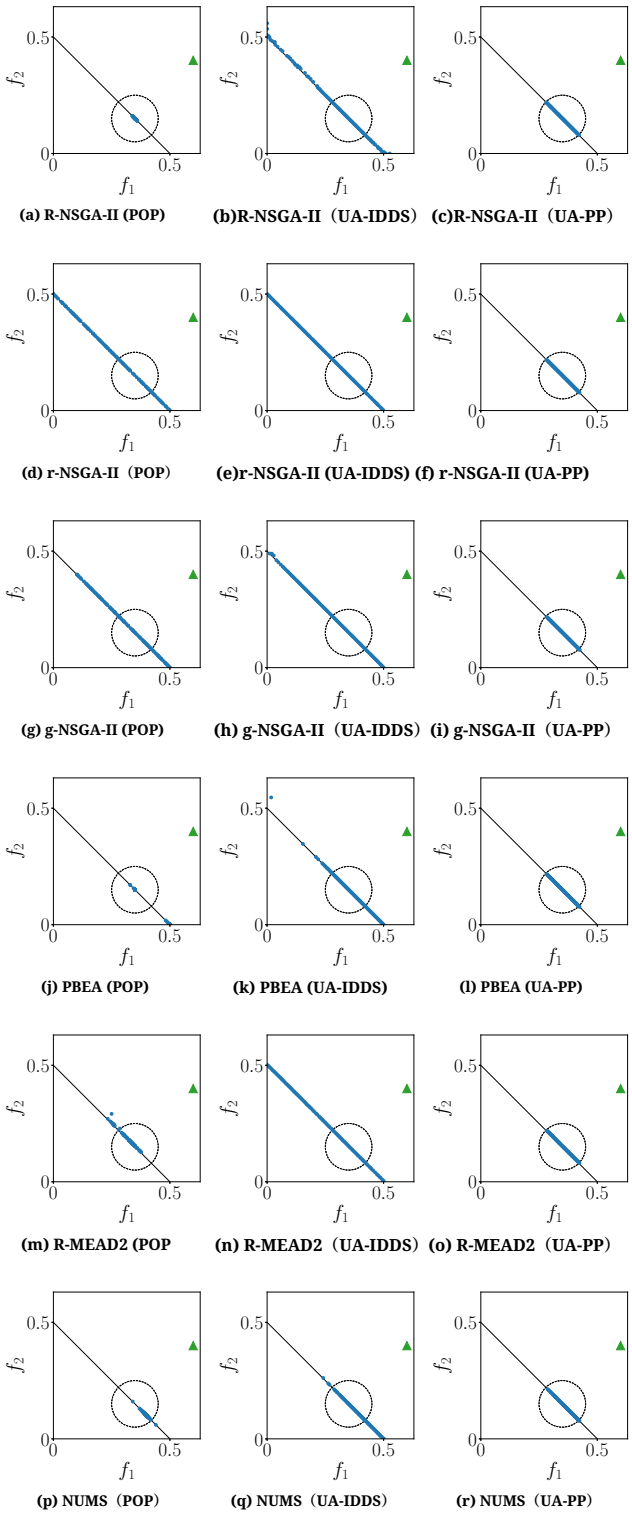
[18] Kalyanmoy Deb、Manikanth Mohan和Shikhar Mishra。2005。。评估基于 epsilon-支配的多目标进化算法以实现快速帕累托最优解的计算。Evol. Comput. 13, 4 (2005), 501–525。https://doi.org/ 10 1162 106365605774666895 。 / Kalyanmoy Deb、J. Sundar、Udaya Bhaskara 和 Shamik Chaudhuri。2006。基于参考点的进化算法多目标优化。Int. J. Comput. Intell. 2, 3 (2006), 273–286。[20] Kalyanmoy Deb、Lothar Thiele、Marco Laumanns 和 Eckart Zitzler。2005。进化多目标优化的可扩展测试问题。载于《进化多目标优化：理论进展与应用》。Springer，105–145。[21] Joaquín Derrac、Salvador García、Daniel Molina 和 Francisco Herrera。2011。使用非参数统计检验作为比较进化算法和群体智能算法的方法：实用教程。Swarm Evol. Comput. 1, 1 2011 3 18 10 1016 2011 02 002 ( ), – 。 https://doi.org/ 。 /j.swevo. . . Antoine S. D. Dymond、Schalk Kok 和 P. Stephan Heyns。2013。。多目标优化算法的性能对目标函数评估预算的敏感性。载于《IEEE进化计算大会论文集，CEC 2013》，墨西哥坎昆，2013年6月20日至23日。IEEE，1868–1875。https://doi.org/ 10 1109 2013 6557787 。/CEC. . Carlos M. Fonseca 和 Peter J. Fleming。1993。用于多目标优化的遗传算法：形式化、讨论与推广。载于《第5届国际遗传算法会议论文集》，美国伊利诺伊州厄巴纳-香槟，1993年6月，Stephanie Forrest（编）。Morgan Kaufmann，416–423。[24] Hisao Ishibuchi、Hiroyuki Masuda、Yuki Tanigaki 和 Yusuke Nojima。2015。世代距离和反世代距离中的修正距离计算。载于《进化多准则优化——第8届国际会议，EMO 2015》，葡萄牙吉马良斯，2015年3月29日至4月1日。论文集，第二部分（《计算机科学讲义》，第9019卷），António Gaspar-Cunha、Carlos Henggeler Antunes 和 Carlos A. Coello Coello（编）。Springer，110–125。https://doi.org/ 10 1007 978 3 319 15892 1_8 。/ - - - - Hisao Ishibuchi、Lie Meng Pang 和 Ke Shang。2022。多目标进化算法公平性能比较中的难点 [研究前沿]。IEEE Comput. Intell. Mag. 17, 1 (2022), 86–101。https://doi.org/10.1109/MCI.2021.3129961[26] Hisao Ishibuchi、Yu Setoguchi、Hiroyuki Masuda 和 Yusuke Nojima。2016。如何在不同种群规模和存档规模设置下比较多目标算法。载于《IEEE进化计算大会，CEC 2016》，加拿大不列颠哥伦比亚省温哥华，2016年7月24日至29日。IEEE，1149–1156。https://doi.org/10.1109/CEC.2016.7743917[27] Andrzej Jaszkiewicz 和 Thibaut Lust。2018。基于ND树的更新：动态非支配问题的快速算法。IEEE Trans. Evol. Comput. 22, 5 2018 778 791 10 1109 2018 2799684 ( ), – 。 https://doi.org/ 。/TEVC. . Ke Li、Renzhi Chen、Geyong Min 和 Xin Yao。2018。。分解式多目标优化中的偏好集成。IEEE Trans. Cybern. 48, 12 (2018), 3359 3370 10 1109 2018 2859363 – 。https://doi.org/ 。/TCYB. .

[29] Ke Li 、 Kalyanmoy Deb 和 Xin Yao。2018。。R 度量：使用参考点评估基于偏好的进化多目标优化的性能。IEEE Trans. Evol. Comput. 22, 6 (2018), 821–835。https://doi.org/10.1109/TEVC.2017.2737781[30] Ke Li、Minhui Liao、Kalyanmoy Deb、Geyong Min 和 Xin Yao。2020。偏好是否总能带来帮助？基于参考点的偏好型进化多目标优化整体研究。IEEE Trans. Evol. Comput. 24, 6 2020 1078 1096 10 1109 2020 2987559 ( ), – 。 https://doi.org/ 。 /TEVC. . Miqing Li 和 Xin Yao。2019。。多目标优化中解集的质量评估：综述。ACM Comput. Surv. 52, 2 (2019), 26:1–26:38。https://doi.org/ 10 1145 3300148 。/ Manuel López-Ibáñez、Joshua D. Knowles 和 Marco Laumanns。2011。目标向量的序贯在线归档。载于《进化多准则优化——第6届国际会议，EMO 2011》，巴西欧鲁普雷图，2011年4月5日至8日。论文集（《计算机科学讲义》，第6576卷），Ricardo H. C. Takahashi、Kalyanmoy Deb、Elizabeth F. Wanner 和 Salvatore Greco（编）。Springer, 46 60 10 1007 978 3 642 19893 9_4 – 。https://doi.org/ 。/ - - - - Kaisa Miettinen。1998。。非线性多目标优化。Springer。[34] Kaisa Miettinen 和 Marko M. Mäkelä。2002。多目标优化中的标量化函数。OR Spectr. 24, 2 (2002), 193–213。https://doi.org/10.1007/s00291-001-0092-9[35] Asad Mohammadi、Mohammad Nabi Omidvar 和 Xiaodong Li。2013。一种面向用户偏好的多目标进化算法新性能指标。载于《IEEE进化计算大会（CEC）》。IEEE，2825–2832。https://doi.org/10.1109/CEC.2013.6557912[36] Asad Mohammadi、Mohammad Nabi Omidvar、Xiaodong Li 和 Kalyanmoy Deb。2014。面向多目标优化的用户偏好与分解方法集成。载于《IEEE进化计算大会（CEC）》。IEEE, 421 428 10 1109 2014 6900595 – 。https://doi.org/ 。/

[37] CEC. . Julián Molina、Luis V. Santana-Quintero、Alfredo García Hernández-Díaz、Carlos A. Coello Coello 和 Rafael Caballero。2009。g-支配：基于参考点的多目标元启发式支配关系。Eur. J. Oper. Res. 197, 2 (2009), 685–692。https://doi.org/10.1016/j.ejor.2008.07.015.

[38] Robin C. Purshouse、Kalyanmoy Deb、Maszatul M. Mansor、SanazMostaghim 和 Rui Wang。2014。《混合进化多准则决策方法综述》。载于*IEEE* 进化计算大会 *(CEC)*。IEEE, 1147 1154 10 1109 2014 6900368 – 。https://doi.org/ 。/CEC。
[39] 。Andreea Radulescu 、 Manuel López-Ibáñez 和 Thomas Stützle。2013。《自动改进多目标进化算法的任意时刻行为》。载于进化多准则优化——第七届国际会议，*EMO 2013*，英国谢菲尔德，*2013*年 *3* 月 *19–22*日。会议录（计算机科学讲义，第 *7811* 卷），Robin C. Purshouse、Peter J. Fleming、Carlos M. Fonseca、Salvatore Greco和 Jane Shaw（编）。Springer，825–840。https://doi.org/ 10 1007 978 3 642 37140 0_61 。/ - - - - Ana Belen Ruiz 、 RubénSaborido 和 MarianoLuque。2015。《多目标优化的基于偏好的进化算法：加权成就标量函数遗传算法》。*J. Glob. Optim.* 62，1（2015），101–129。https://doi.org/10.1007/s10898-014-0214-y [41] Lamjed Ben Said、SlimBechikh和 KhaledGhédira。2010。《r-支配：交互式进化多准则决策的一种新支配关系》。*IEEE Trans. Evol. Comput.*14，5（2010），801–818。https://doi.org/10.1109/TEVC.2010.2041060[42] Serpil Sayin。2000。《衡量多目标数学规划中有效集离散表示的质量》。*Math. Program.* 87，3（ 2000 543 560 10 1007 101070050011 ），– 。https://doi.org/ 。/s Oliver Schütze、Xavier Esquivel、Adriana Lara 和 Carlos A. Coello Coello。2012。《使用平均 Hausdorff 距离作为进化多目标优化中的性能度量》。*IEEE Trans. Evol. Comput.* 16，4（2012），504 522 10 1109 2011 2161872 – 。https://doi.org/ 。/TEVC。 。
[43] [44] Ke Shang、Hisao Ishibuchi和 Weiyu Chen。 。2021。《面向多目标优化的贪心近似超体积子集选择》。载于*GECCO '21*：遗传与进化计算大会，法国里尔，*2021* 年 *7* 月 *10 14* 日，448 456-Francisco Chicano 和 Krzysztof Krawiec（编）。ACM，– 。https: //doi.org/ 10 1145 3449639 3459390 。/ 。Ke Shang、HisaoIshibuchi和 Yang Nan。
[45] 2021。《基于距离的子集选择再探》。载于*GECCO '21*：遗传与进化计算大会，法国里尔，*2021* 年 *7* 月 *10-14* 日，Francisco Chicano和 Krzysztof Krawiec（编）。ACM，439 447 10 1145 3449639 3459391 – 。https://doi.org/ 。/。Ryoji Tanabe
[46] 和 HisaoIshibuchi。 。2018。《两种不同优化场景下 MOEA/D 控制参数分析》。*Appl. Soft Comput.* 70（ 2018 22 40 10 1016 2018 05 014 ），–。
[47] https://doi.org/ 。/j.asoc。 。 。Ryoji Tanabe、HisaoIshibuchi和 Akira Oyama。 。2017。《两种优化场景下多目标与多目标进化算法的基准测试》。*IEEE Access* 5 2017 19597 19619 10 1109 2017 2751071 ( ), – 。
[48] https://doi.org/。/ACCESS。 。RyojiTanabe 和KeLi。 。2023。《使用参考点的基于偏好的进化多目标优化质量指标：综述与分析》。*CoRR* abs/2301.12148（2023）。https://doi.org/10.48550/arXiv.2301.12148 arXiv:2301.12148[49] Lothar Thiele、Kaisa Miettinen、Pekka J. Korhonen 和Julián Molina Luque。2009。《基于偏好的多目标优化进化算法》。*Evol. Comput.* 17, 3 2009 411 436 10 1162 2009 17 3 411 ( ), – 。
[50] https://doi.org/ 。/evco。 。 。Andrzej P. Wierzbicki。 。1980。《多目标优化中参考目标的使用》。载于多准则决策理论与应用。Springer，468 486 10 1007 978 3 642 48782 8_32 – 。https://doi.org/ 。/ - - - -
[51] QingfuZhang 和 Hui Li。 。2007。《MOEA/D：一种基于分解的多目标进化算法》。*IEEE Trans. Evol. Comput.* 11，6（2007），712–731。https://doi.org/ 10 1109 2007 892759 。/TEVC。 。Eckart Zitzler 和 Simon Künzli。
[52] 。2004。《多目标搜索中的基于指标的选择》。载于自然界中的并行问题求解 *(PPSN)*。832–842。https://doi.org/ 10.1007/978-3-540-30217-9 84[53] Eckart Zitzler、Marco Laumanns 和 LotharThiele。2001。*SPEA2*：改进强度*Pareto* 进化算法。技术报告 103。瑞士联邦理工学院（ETHZ）。[54] Eckart Zitzler 和 Lothar Thiele。1998。《使用进化算法进行多目标优化——比较案例研究》。载于自然界中的并行问题求解 *(PPSN)*。Springer，292 304 10 1007 0056872 – 。
[55] https://doi.org/ 。/BFb Eckart Zitzler、Lothar Thiele、Marco Laumanns、Carlos M. Fonseca 和 VivianeGrunertda Fonseca。2003。《多目标优化器的性能评估：分析与综述》。*IEEE Trans. Evol. Comput.*7，2（2003），117–132。https://doi.org/10.1109/TEVC.2003.810758

补充材料



(a) R-NSGA-II (POP) (b)R-NSGA-II（UA-IDDS） (c)R-NSGA-II（UA-PP）

(d) r-NSGA-II（POP） (e)r-NSGA-II (UA-IDDS) (f) r-NSGA-II (UA-PP)

(g) g-NSGA-II (POP) (h) g-NSGA-II（UA-IDDS） (i) g-NSGA-II（UA-PP）

(j) PBEA (POP) (k) PBEA (UA-IDDS) (l) PBEA (UA-PP)

(m) R-MEAD2 (POP (n) R-MEAD2（UA-IDDS） (o) R-MEAD2（UA-PP）

(p) NUMS（POP） (q) NUMS（UA-IDDS） (r) NUMS（UA-PP）

图 S.1：DTLZ1 上三个解集中的解的目标向量分布，其中$m = 2$，其中 ▲ 是参考点z。点状圆圈表示真实感兴趣区域。“NUMS”代表 MOEA/D-NUMS。

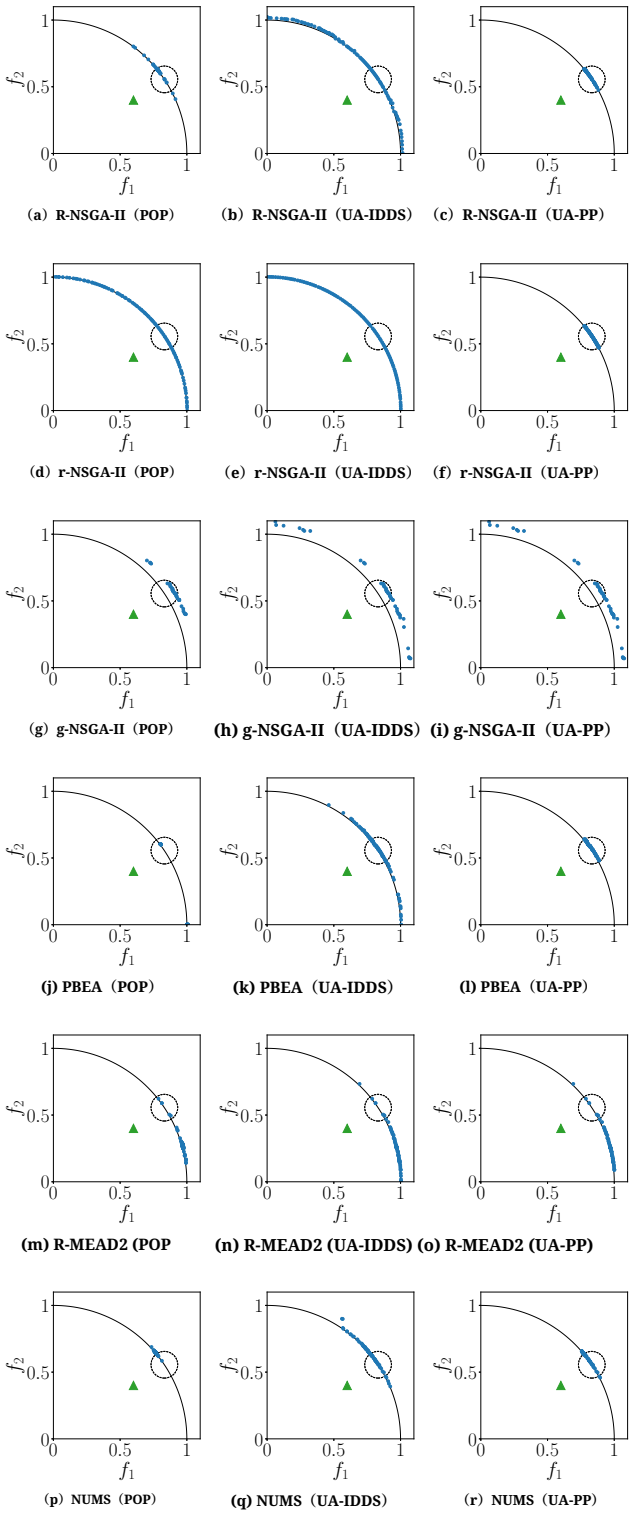


图 S.2：在带有$m = 2$的 DTLZ3 上，三个解集中解的目标向量分布，其中 ▲ 是参考点 z。点状圆表示真实感兴趣区域。“NUMS”代表 MOEA/D-NUMS。

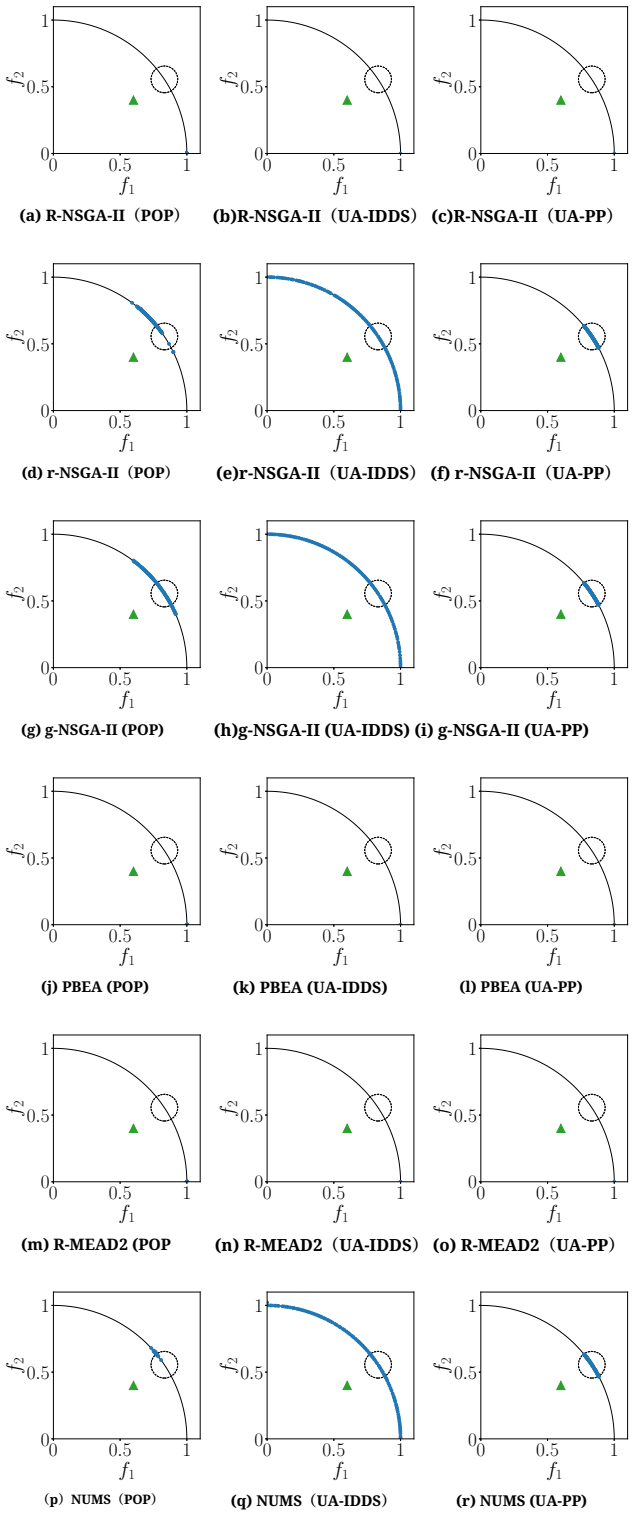


图 S.3：在 DTLZ4 上使用$m = 2$时，三个解集中的解的目标向量分布，其中 ▲ 是参考点 z。点状圆表示真实感兴趣区域。“NUMS”表示 MOEA/D-NUMS。

表 S.1：r-NSGA-II、g-NSGA-II、PBEA、R-MEAD2 和 MOEA/D-NUMS 在 DTLZ1–DTLZ4 问题上找到的三个解子集的平均 IGD⁺-C 值。

(a) r-NSGA-II

问题	m	POP	UA-IDDS	UA-PP
DTLZ1	2	0.0018	0.0015 (+)	0.0007 (+, +)
	3	0.0221	0.0152 (+)	0.0079 (+, +)
	4	0.6009	0.4523 (≈)	0.4361 (≈, ≈)
	5	2.5538	1.8340 (+)	1.2448 (+, +)
	6	2.9928	3.0074 (≈)	1.3174 (+, +)
DTLZ2	2	0.0312	0.0015 (+)	0.0004 (+, +)
	3	0.0979	0.0255 (+)	0.0085 (+, +)
	4	0.1700	0.0618 (+)	0.0222 (+, +)
	5	0.1569	0.0967 (+)	0.0306 (+, +)
	6	0.2044	0.1223 (+)	0.0505 (+, +)
DTLZ3	2	0.0076	0.0075 (≈)	0.0060 (+, +)
	3	0.0905	0.0861 (≈)	0.0713 (+, +)
	4	5.7675	5.5092 (≈)	5.3614 (≈, ≈)
	5	11.0159	10.3284 (≈)	9.1448 (≈, ≈)
	6	12.4904	11.4337 (≈)	9.7152 (≈, ≈)
DTLZ4	2	0.0553	0.0454 (+)	0.0438 (+, +)
	3	0.0241	0.0280 (–)	0.0062 (+, +)
	4	0.0306	0.0552 (–)	0.0153 (+, +)
	5	0.0657	0.0941 (–)	0.0278 (+, +)
	6	0.1120	0.1155 (≈)	0.0367 (+, +)

(b) g-NSGA-II

问题	m	POP	UA-IDDS	UA-PP
DTLZ1	2	0.0267	0.0265 (≈)	0.0259 (+, +)
	3	10.7277	3.7391 (+)	3.7306 (+, ≈)
	4	77.7690	20.7678 (+)	15.7325 (+, +)
	5	123.8328	50.6386 (+)	23.8382 (+, +)
	6	145.9085	74.0552 (+)	30.8597 (+, +)
DTLZ2	2	0.0013	0.0019 (–)	0.0004 (+, +)
	3	0.0264	0.0332 (–)	0.0090 (+, +)
	4	0.2756	0.1978 (+)	0.0999 (+, +)
	5	1.1044	0.5408 (+)	0.3901 (+, +)
	6	1.3364	0.7812 (+)	0.4915 (+, +)
DTLZ3	2	5.4893	5.3352 (≈)	5.3352 (≈, ≈)
	3	30.6638	27.3401 (≈)	27.3305 (≈, ≈)
	4	266.2339	127.7268 (+)	119.3209 (+, ≈)
	5	608.8715	352.6806 (+)	257.7681 (+, +)
	6	851.7529	481.3857 (+)	340.8093 (+, +)
DTLZ4	2	1.3208	0.0347 (+)	0.0329 (+, +)
	3	0.2915	0.0346 (+)	0.0116 (+, +)
	4	0.0884	0.0851 (≈)	0.0344 (+, +)
	5	0.4019	0.2268 (+)	0.1172 (+, +)
	6	0.7191	0.3082 (+)	0.2047 (+, +)

(c) PBEA

问题	m	POP	UA-IDDS	UA-PP
DTLZ1	2	0.0339	0.0009 (+)	0.0006 (+, +)
	3	0.0411	0.0121 (+)	0.0081 (+, +)
	4	0.0609	0.0258 (+)	0.0175 (+, +)
	5	0.0628	0.0385 (+)	0.0285 (+, +)
	6	0.0915	0.0647 (+)	0.0535 (+, +)
DTLZ2	2	0.0012	0.0023 (–)	0.0003 (+, +)
	3	0.0420	0.0263 (+)	0.0053 (+, +)
	4	0.0550	0.0590 (–)	0.0296 (+, +)
	5	0.0799	0.1030 (–)	0.0527 (+, +)
	6	0.1494	0.1536 (≈)	0.1412 (+, +)
DTLZ3	2	0.0571	0.0061 (+)	0.0059 (+, ≈)
	3	0.1553	0.0804 (+)	0.0963 (+, ≈)
	4	0.2022	0.1328 (+)	0.1664 (+, ≈)
	5	0.2503	0.2114 (≈)	0.2389 (≈, ≈)
	6	0.3300	0.2740 (+)	0.3157 (≈, –)
DTLZ4	2	0.0768	0.0775 (–)	0.0763 (+, +)
	3	0.0580	0.0470 (+)	0.0281 (+, +)
	4	0.0897	0.0886 (≈)	0.0660 (+, +)
	5	0.1183	0.1309 (–)	0.0884 (+, +)
	6	0.1506	0.1488 (≈)	0.1174 (+, +)

(d) R-MEAD2

问题	m	POP	UA-IDDS	UA-PP
DTLZ1	2	0.0140	0.0072 (+)	0.0067 (+, ≈)
	3	0.1785	0.1344 (≈)	0.1324 (≈, ≈)
	4	0.2730	0.2247 (+)	0.2235 (+, ≈)
	5	0.2630	0.2297 (≈)	0.2286 (+, ≈)
	6	0.2874	0.2567 (≈)	0.2554 (≈, ≈)
DTLZ2	2	0.0112	0.0032 (+)	0.0019 (+, +)
	3	0.0931	0.0352 (+)	0.0141 (+, +)
	4	0.2894	0.1176 (+)	0.0948 (+, +)
	5	0.3439	0.2084 (+)	0.1837 (+, ≈)
	6	0.4480	0.3216 (+)	0.3077 (+, ≈)
DTLZ3	2	0.1931	0.1437 (≈)	0.1439 (≈, ≈)
	3	0.6909	0.6211 (+)	0.6211 (+, ≈)
	4	0.8050	0.7381 (+)	0.7381 (+, ≈)
	5	0.7892	0.7068 (+)	0.7093 (+, ≈)
	6	0.8368	0.7823 (+)	0.7824 (+, ≈)
DTLZ4	2	0.1639	0.1580 (≈)	0.1579 (≈, ≈)
	3	0.1734	0.1711 (≈)	0.1710 (≈, ≈)
	4	0.2456	0.2404 (≈)	0.2401 (≈, ≈)
	5	0.2958	0.2662 (+)	0.2629 (+, ≈)
	6	0.2598	0.2409 (+)	0.2443 (≈, ≈)

(e) MOEA/D-NUMS

问题	m	POP	UA-IDDS	UA-PP
DTLZ1	2	0.0148	0.0007 (+)	0.0014 (+, –)
	3	0.0278	0.0092 (+)	0.0054 (+, +)
	4	0.0423	0.0263 (+)	0.0149 (+, +)
	5	0.0339	0.0341 (≈)	0.0231 (+, +)
	6	0.0378	0.0449 (–)	0.0311 (+, +)
DTLZ2	2	0.0452	0.0014 (+)	0.0004 (+, +)
	3	0.1464	0.0426 (+)	0.0345 (+, +)
	4	0.2161	0.1256 (+)	0.1012 (+, +)
	5	0.1696	0.1360 (+)	0.0965 (+, +)
	6	0.2392	0.2457 (–)	0.1780 (+, +)
DTLZ3	2	0.0834	0.0424 (+)	0.0426 (+, ≈)
	3	0.2056	0.1373 (+)	0.1405 (+, ≈)
	4	0.2958	0.2411 (+)	0.2257 (+, ≈)
	5	0.4276	0.4491 (≈)	0.3911 (+, +)
	6	0.7108	0.7999 (–)	0.6825 (≈, +)
DTLZ4	2	0.0453	0.0030 (+)	0.0007 (+, +)
	3	0.1476	0.0425 (+)	0.0430 (+, ≈)
	4	0.2308	0.0929 (+)	0.0831 (+, +)
	5	0.1802	0.1255 (+)	0.0926 (+, +)
	6	0.2639	0.2134 (+)	0.2046 (+, +)

表S.2：在DTLZ1–DTLZ4 问题上，每个函数评估次数和 $m \in \{2,4,6\}$ 对应的八个 μ 值的排名（第1/第2/第3/第4/第5/第6/第7/第8）。

(a) $m = 2$

PBEMO	1K 函数评估次数	5K 函数评估次数	10K 次函数评估次数	30K 函数评估次数	50K 次函数评估次数
R-NSGA-II	8/20/40/100/200/300/400/500	20/40/8/100/200/300/400/500	8/20/40/100/200/300/400/500	8/100/40/200/20/300/400/500	200/100/300/40/400/8/20/500
r-NSGA-II	20/40/8/100/200/300/400/500	20/40/8/100/200/300/400/500	20/8/40/100/200/300/400/500	40/20/100/8/200/300/400/500	40/20/100/8/200/300/400/500
g-NSGA-II	20/8/40/100/200/300/400/500	40/100/20/200/8/300/400/500	40/100/200/20/300/400/8/500	200/100/40/300/400/20/500/8	100/200/40/300/20/400/500/8
PBEA	20/40/8/100/200/300/400/500	40/20/8/200/100/300/400/500	8/20/40/100/200/300/400/500	100/40/20/300/200/8/400/500	200/40/100/300/20/400/8/500
R-MEAD2	8/20/40/100/200/300/400/500	8/20/40/100/200/300/400/500	8/20/40/100/200/300/400/500	20/40/100/8/200/300/400/500	20/40/100/200/300/8/400/500
MOEA/D-NUMS	8/20/40/100/200/300/400/500	8/20/100/40/200/300/400/500	8/20/40/200/100/300/400/500	20/100/40/8/200/300/400/500	20/100/40/200/8/300/400/500

(b) $m = 4$

PBEMO	1K 函数评估次数	5K 函数评估次数	10K 次函数评估次数	30K 次函数评估次数	50K 函数评估次数
R-NSGA-II	8/20/40/100/200/300/400/500	40/100/20/200/8/300/400/500	20/200/300/40/100/8/400/500	20/40/300/400/8/100/200/500	20/40/100/400/300/500/8/200
r-NSGA-II	40/20/8/100/200/300/400/500	40/20/100/200/8/300/400/500	40/20/100/200/8/300/400/500	40/20/100/8/200/300/400/500	40/20/8/100/200/300/400/500
g-NSGA-II	40/20/8/100/200/300/400/500	100/200/300/40/20/400/500/8	200/300/100/400/500/40/20/8	400/500/300/200/100/40/20/8	400/500/200/300/100/40/20/8
PBEA	20/40/8/100/200/300/400/500	100/40/20/200/300/8/400/500	40/100/20/200/300/8/400/500	100/200/300/400/40/500/20/8	200/300/400/40/100/500/20/8
R-MEAD2	20/8/40/100/300/200/400/500	100/20/200/8/40/300/400/500	100/200/8/20/300/40/400/500	100/20/8/200/400/40/300/500	20/8/100/300/500/40/200/400
MOEA/D-NUMS	8/20/40/100/200/400/300/500	20/8/40/100/200/300/400/500	20/8/40/100/200/300/400/500	20/40/8/200/100/300/400/500	20/40/8/100/200/300/400/500

(c) $m = 6$

PBEMO	1K 函数评估次数	5K 函数评估次数	10K 函数评估次数	30K 函数评估次数	50K 函数评估次数
R-NSGA-II	20/40/100/8/200/300/400/500	40/100/200/20/300/8/400/500	100/200/300/20/40/8/400/500	40/20/200/400/100/300/500/8	100/40/400/20/500/8/200/300
r-NSGA-II	40/20/100/8/200/300/400/500	40/100/200/8/20/300/400/500	100/40/20/200/8/300/400/500	100/20/40/200/8/300/400/500	100/40/20/200/8/300/400/500
g-NSGA-II	8/20/200/100/400/40/300/500	8/500/300/400/200/20/40/100	300/8/500/400/200/20/40/100	8/500/400/300/20/200/40/100	8/500/400/20/300/200/40/100
PBEA	20/40/8/100/200/300/400/500	40/100/200/20/300/400/8/500	100/200/40/300/20/400/8/500	300/100/200/400/500/40/20/8	300/400/200/100/500/40/20/8
R-MEAD2	8/40/20/100/200/300/400/500	100/200/8/20/40/300/400/500	200/300/8/20/100/400/40/500	20/200/40/100/400/500/300/8	40/20/100/200/400/500/8/300
MOEA/D-NUMS	20/8/40/100/200/300/400/500	40/100/8/20/200/300/400/500	40/20/100/8/200/400/300/500	20/40/100/8/200/300/400/500	20/40/100/8/200/300/400/500

表 S.3 ：DTLZ1–DTLZ4 问题上 r-NSGA-II 的最佳种群规模 μ，其中 $r \in \{0.01,0.05,0.1,0.2,0.3\}$。

(a) $m = 2$

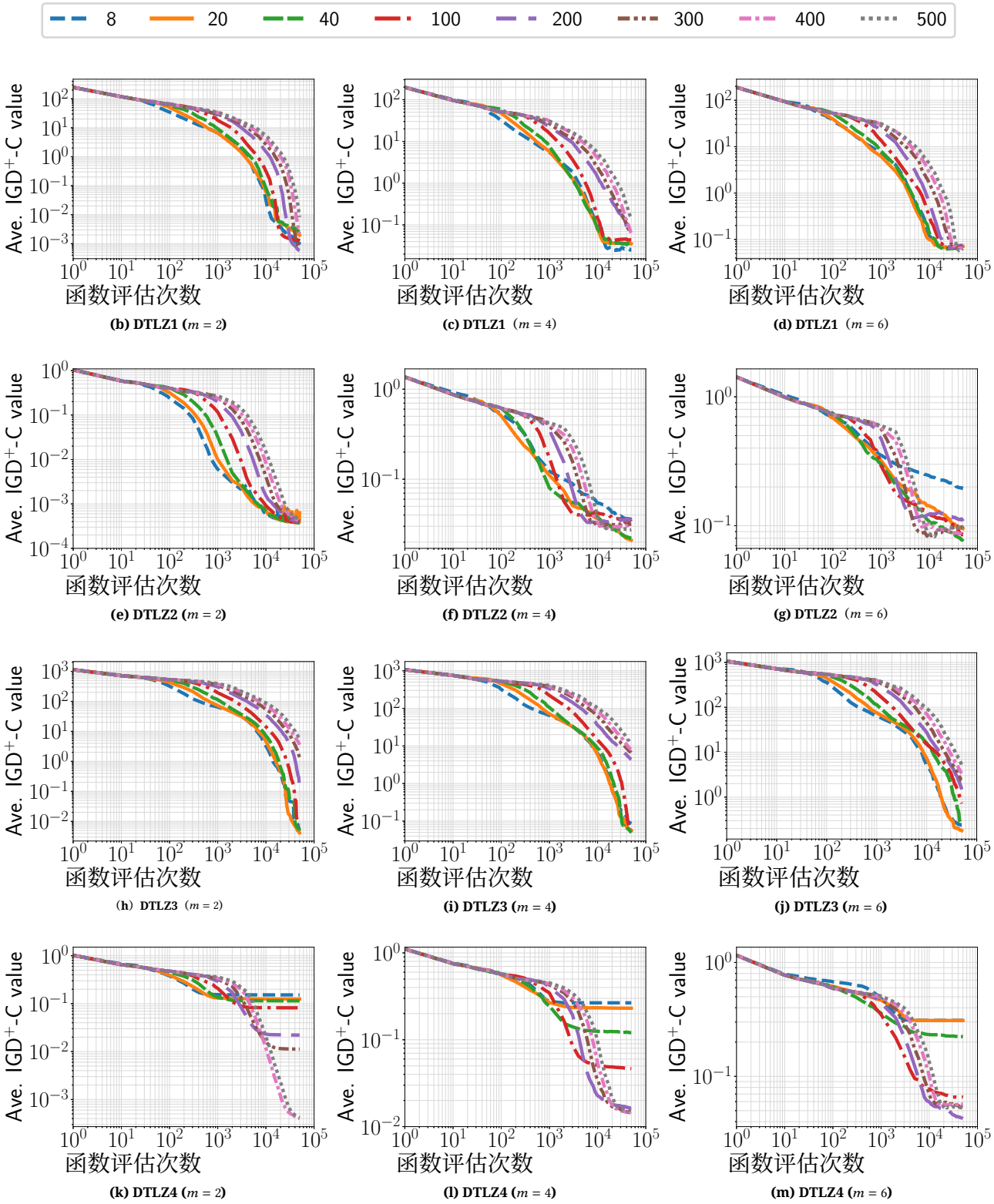
r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	20	20	8	100	40
0.05	20	20	20	100	40
0.1	20	20	20	40	40
0.2	20	20	8	40	40
0.3	20	20	8	40	40

(b) $m = 4$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	40	40	40	40	40
0.05	40	40	40	40	40
0.1	40	40	40	40	40
0.2	20	40	40	40	40
0.3	20	40	20	40	40

(c) $m = 6$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	40	40	40	100	100
0.05	40	40	40	100	100
0.1	40	40	100	100	100
0.2	40	40	100	100	40
0.3	40	40	40	40	40



(b) DTLZ1 ($m = 2$) (c) DTLZ1（$m = 4$) (d) DTLZ1（$m = 6$)

(e) DTLZ2 ($m = 2$) (f) DTLZ2 ($m = 4$) (g) DTLZ2（$m = 6$)

(h) DTLZ3（$m = 2$) (i) DTLZ3 ($m = 4$) (j) DTLZ3 ($m = 6$)

(k) DTLZ4 ($m = 2$) (l) DTLZ4 ($m = 4$) (m) DTLZ4 ($m = 6$)

图 S.4：R-NSGA-II 在 DTLZ1–DTLZ4 问题上采用不同种群规模时的平均 IGD$^+$-C 值，其中 $m \in \{2,4,6\}$。

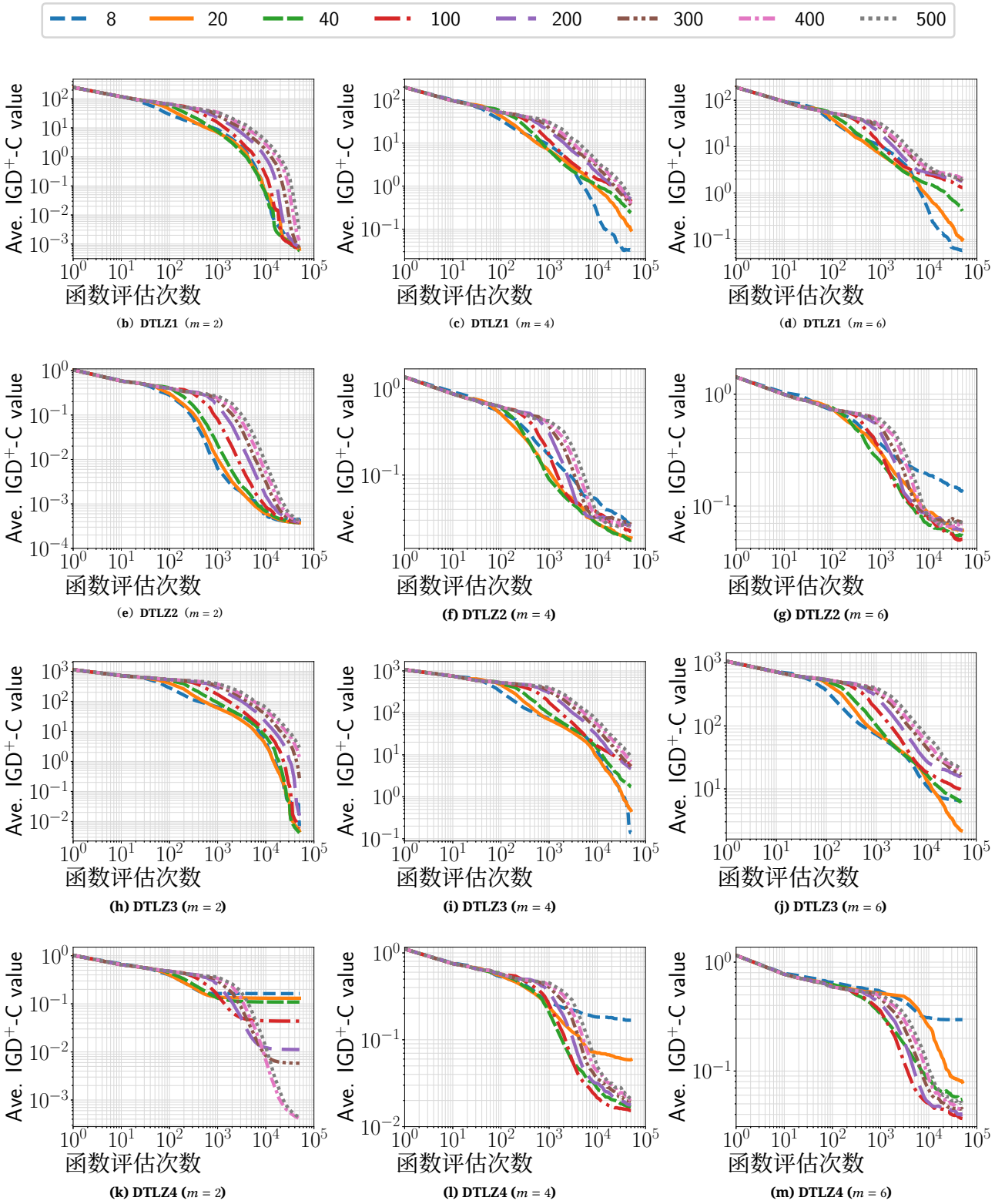
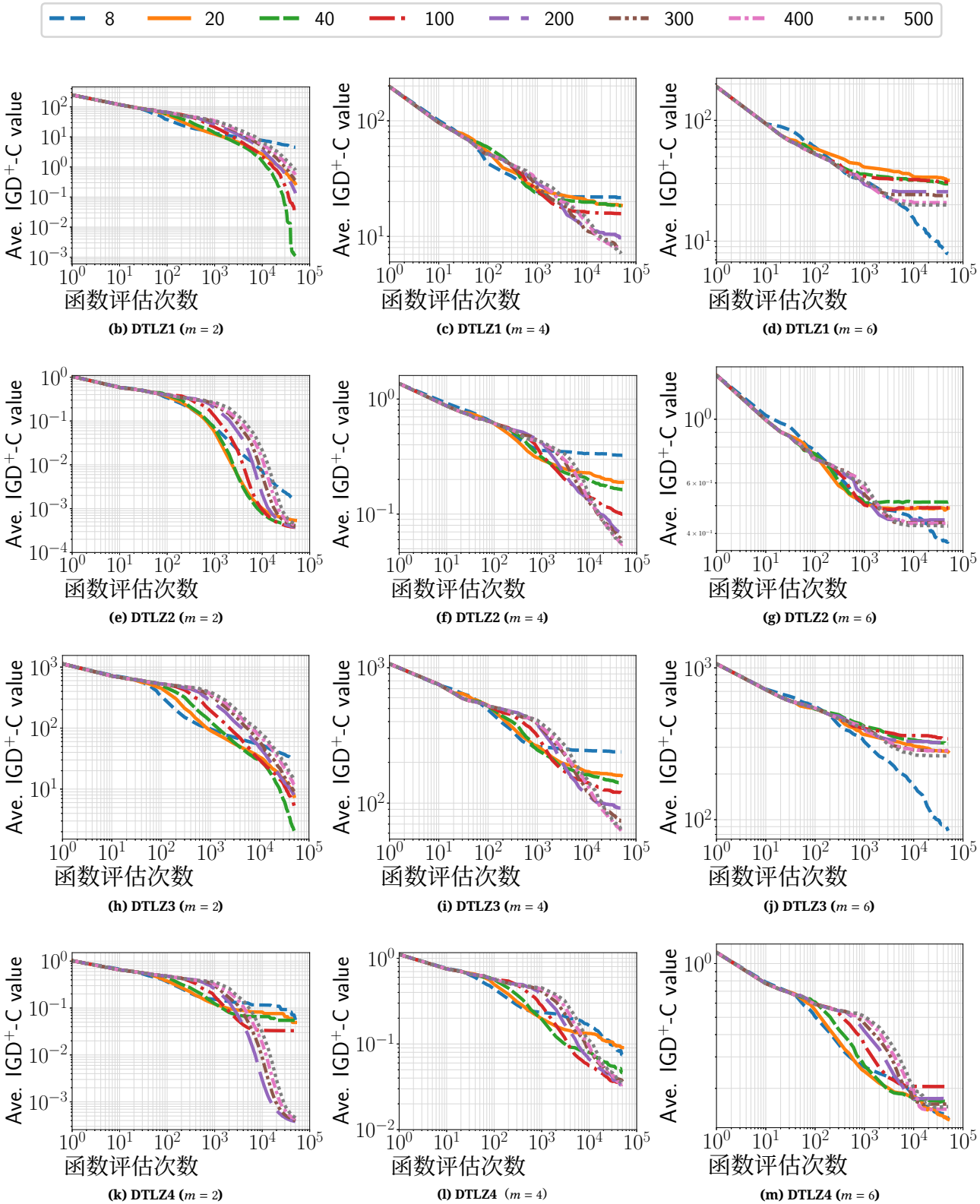


图 S.5：在带有 $m \in \{2,4,6\}$的 DTLZ1–DTLZ4 问题上，不同种群规模的 r-NSGA-II 的平均 IGD⁺-C 值。



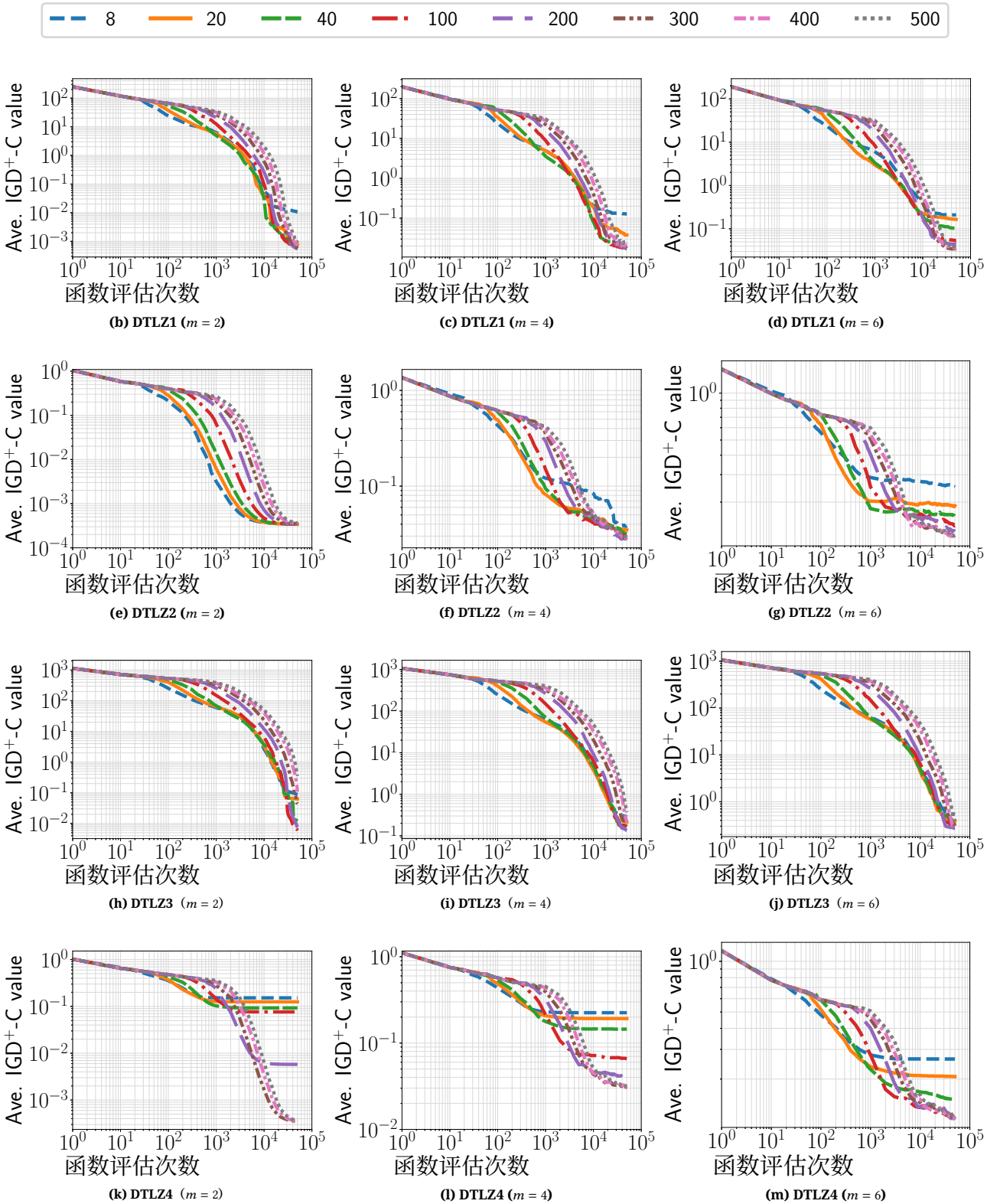
(b) DTLZ1 ($m = 2$) (c) DTLZ1 ($m = 4$) (d) DTLZ1 ($m = 6$)

(e) DTLZ2 ($m = 2$) (f) DTLZ2 ($m = 4$) (g) DTLZ2 ($m = 6$)

(h) DTLZ3 ($m = 2$) (i) DTLZ3 ($m = 4$) (j) DTLZ3 ($m = 6$)

(k) DTLZ4 ($m = 2$) (l) DTLZ4（$m = 4$) (m) DTLZ4 ($m = 6$)

图 S.6：g-NSGA-II 在DTLZ1–DTLZ4 问题上采用不同种群规模时的平均 IGD⁺-C 值，种群规模为 $m \in \{2,4,6\}$。



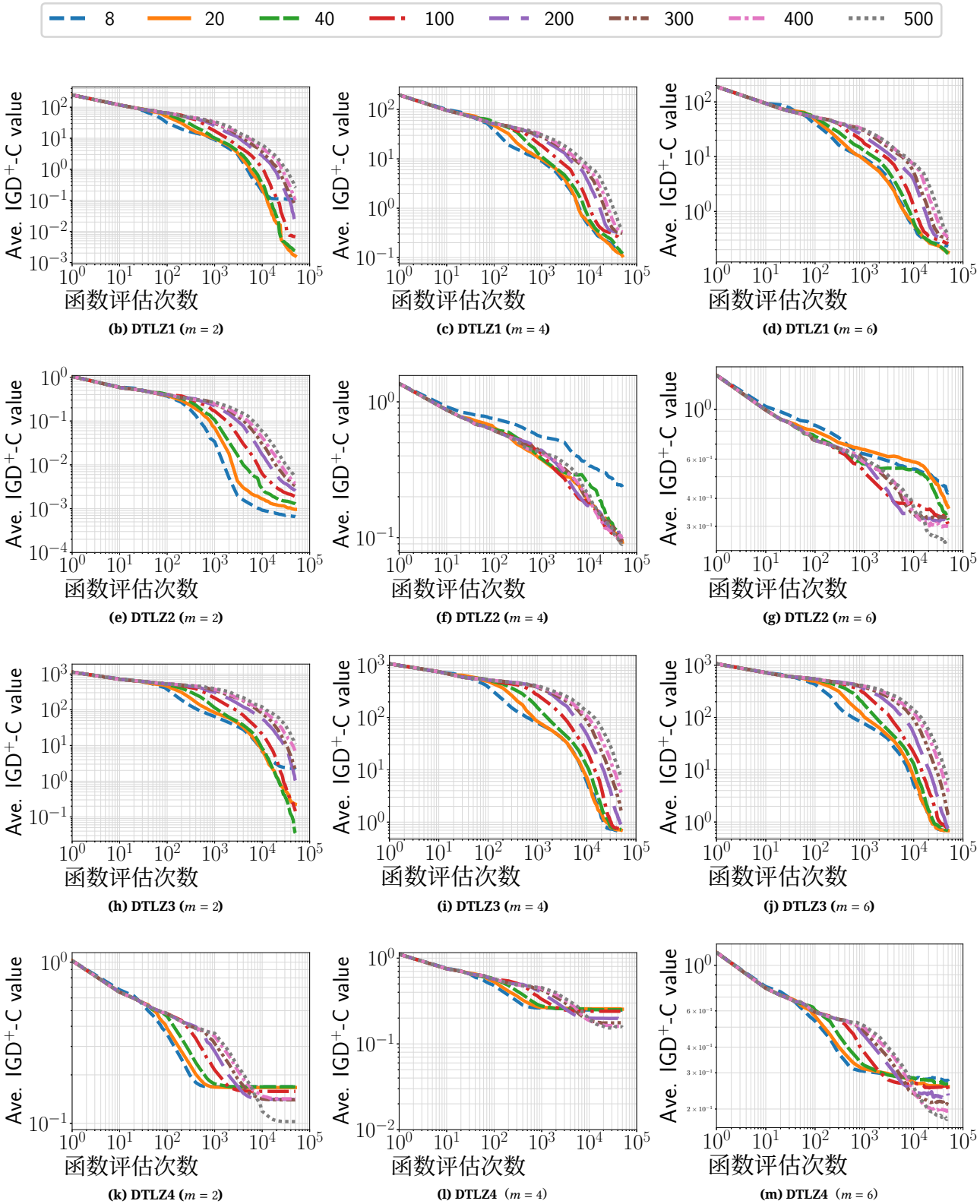
(b) DTLZ1 ($m = 2$) (c) DTLZ1 ($m = 4$) (d) DTLZ1 ($m = 6$)

(e) DTLZ2 ($m = 2$) (f) DTLZ2（$m = 4$) (g) DTLZ2（$m = 6$)

(h) DTLZ3（$m = 2$) (i) DTLZ3（$m = 4$) (j) DTLZ3（$m = 6$)

(k) DTLZ4（$m = 2$) (l) DTLZ4 ($m = 4$) (m) DTLZ4 ($m = 6$)

图 S.7：在 $m \in \{2,4,6\}$ 下，PBEA 在 DTLZ1–DTLZ4 问题上采用不同种群规模时的平均 IGD⁺-C 值。



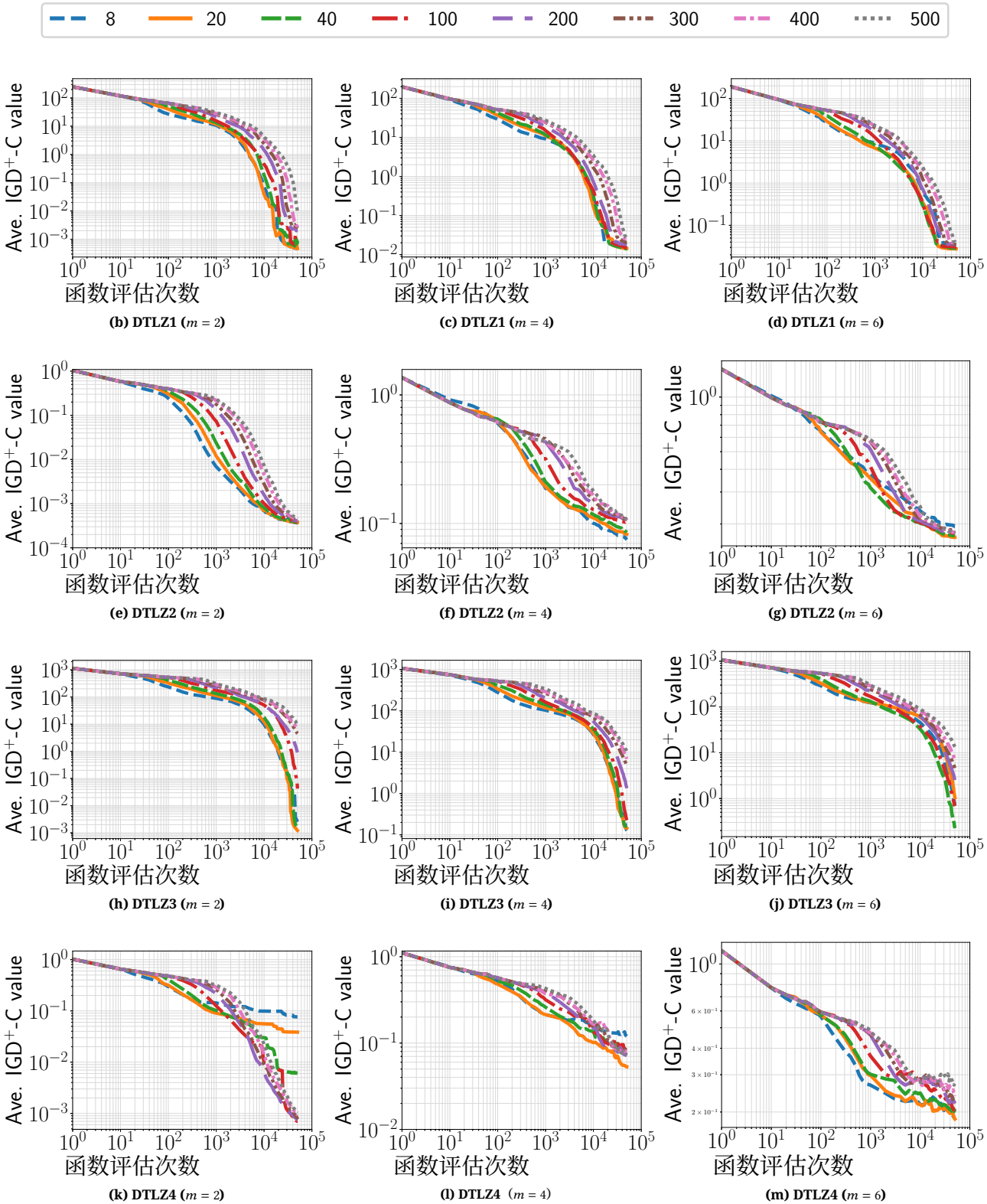
(b) DTLZ1 ($m = 2$) (c) DTLZ1 ($m = 4$) (d) DTLZ1 ($m = 6$)

(e) DTLZ2 ($m = 2$) (f) DTLZ2 ($m = 4$) (g) DTLZ2 ($m = 6$)

(h) DTLZ3 ($m = 2$) (i) DTLZ3 ($m = 4$) (j) DTLZ3 ($m = 6$)

(k) DTLZ4 ($m = 2$) (l) DTLZ4（$m = 4$） (m) DTLZ4（$m = 6$）

图 S.8：R-MEAD2 的平均 IGD⁺-C 值 $m \in \{2, 4, 6\}$ 在 DTLZ1–DTLZ4 问题上采用不同种群规模时，



(b) DTLZ1 ($m = 2$) (c) DTLZ1 ($m = 4$) (d) DTLZ1 ($m = 6$)

(e) DTLZ2 ($m = 2$) (f) DTLZ2 ($m = 4$) (g) DTLZ2 ($m = 6$)

(h) DTLZ3 ($m = 2$) (i) DTLZ3 ($m = 4$) (j) DTLZ3 ($m = 6$)

(k) DTLZ4 ($m = 2$) (l) DTLZ4（$m = 4$) (m) DTLZ4 ($m = 6$)

图 S.9：MOEAD-NUMS 在 DTLZ1–DTLZ4 问题上使用不同种群规模时的平均 IGD⁺-C 值，种群规模为$m \in \{2,4,6\}$。

表 S.4：在 $r \in \{0.01,0.05,0.1,0.2,0.3\}$ 时，g-NSGA-II 在 DTLZ1–DTLZ4 问题上的最佳种群规模 μ。

(a) $m = 2$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	20	40	40	200	100
0.05	20	40	40	200	100
0.1	20	40	40	200	100
0.2	20	40	40	40	100
0.3	20	40	40	200	100

(b) $m = 4$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	40	100	200	400	400
0.05	40	100	200	400	400
0.1	40	100	200	400	400
0.2	40	100	200	400	400
0.3	40	100	200	400	400

(c) $m = 6$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	8	8	300	8	8
0.05	8	8	300	8	8
0.1	8	8	300	8	8
0.2	8	8	8	8	8
0.3	8	8	8	8	8

Tab补充表 S.5：当 $r \in \{0.01,0.05,0.1,0.2,0.3$时，PBEA 在 DTLZ1–DTLZ4 问题上的最佳种群规模为 μ }.

(a) $m = 2$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	20	40	8	100	40
0.05	20	40	8	100	200
0.1	20	40	8	100	200
0.2	20	40	20	100	40
0.3	20	40	40	100	300

(b) $m = 4$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	20	100	40	200	200
0.05	20	100	40	200	200
0.1	20	100	40	100	200
0.2	20	100	40	200	200
0.3	20	100	40	200	300

(c) $m = 6$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	20	40	100	300	200
0.05	20	40	100	300	400
0.1	20	40	100	300	300
0.2	20	40	100	300	400
0.3	20	40	100	300	400

表 S.6：在 $r \in \{0.01,0.05,0.1,0.2,0.3\}$ 下，R-MEAD2 在 DTLZ1–DTLZ4 问题上的最佳种群规模为 μ。

(a) $m = 2$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	8	8	8	20	20
0.05	8	8	8	20	20
0.1	8	8	8	20	20
0.2	8	8	8	20	20
0.3	8	8	8	20	20

(b) $m = 4$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	20	100	20	20	20
0.05	20	100	20	20	20
0.1	20	100	100	100	20
0.2	20	100	100	100	20
0.3	20	100	200	100	100

(c) $m = 6$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	8	100	200	20	40
0.05	8	100	200	20	40
0.1	8	100	200	20	40
0.2	8	100	200	20	40
0.3	8	100	300	20	20

表 S.7：MOEA/D-NUMS 在 DTLZ1–DTLZ4 问题上的最佳种群规模为 μ，对应的 $r \in \{0.01,0.05,0.1,0.2,0.3\}$。

(a) $m = 2$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	8	8	8	100	20
0.05	8	8	8	20	40
0.1	8	8	8	20	20
0.2	8	20	20	20	20
0.3	8	8	20	8	20

(b) $m = 4$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	8	20	20	20	20
0.05	8	20	20	20	20
0.1	8	20	20	20	20
0.2	8	20	20	20	20
0.3	8	20	20	20	20

(c) $m = 6$

r	1K 次函数评估	5K 次函数评估	10K 次函数评估	30K 次函数评估	50K 次函数评估
0.01	20	40	40	20	20
0.05	20	40	40	20	20
0.1	20	40	40	20	20
0.2	8	40	40	20	20
0.3	8	40	40	20	20