一种用于少对多优化的高效进化算法

Ke Shang, *IEEE*高级会员，Hisao Ishibuchi，*IEEE*会士，Zexuan Zhu，*IEEE*高级会员，Qingfu Zhang，*IEEE*会士

arXiv:2601.06387v1 [cs.NE] 10 Jan 2026

摘要——**少对多（F4M）优化是近期在多目标优化中提出的一种新范式，旨在寻找一小组解，以有效处理大量相互冲突的目标。与传统的多目标优化方法不同，后者通常致力于全面覆盖帕累托前沿，F4M优化强调寻找规模较小的代表性解集，以高效应对高维目标空间。鉴于F4M优化的计算复杂度及其实用价值，本文提出一种专为高效求解F4M优化问题而设计的新型进化算法。受SMS-EMOA启发，我们提出的方法采用由F4M优化目标引导的$(\mu+1)$进化策略。此外，为了进行严格的性能评估，我们利用R2指标与F4M公式之间的相似性，提出了一套专为F4M优化设计的新型基准测试套件。该测试套件具有很高的灵活性，可通过使用加权切比雪夫函数进行标量化，将任意现有多目标优化问题转化为相应的F4M实例。在基准测试上的全面实验评估表明，与现有最先进算法相比，我们的算法性能更优，尤其是在涉及大量目标的实例上。所提出算法的源代码将公开发布。源代码可在 https://github.com/MOL-SZU/SoM-EMOA 获取。**

录用版本。本文已被 *IEEE/CAA* 自动化学报 接收发表。本版本为作者录用稿，尚未经过 IEEE 编辑、排版或最终格式处理。

索引词—**少对多优化、多目标优化、多目标优化、进化算法。**

本研究得到国家自然科学基金（项目编号：62472292、62471310、62376115）、广东省基础与应用基础研究基金（项目编号：2025A1515011638）以及中国香港特别行政区研究资助局（GRF 项目编号：CityU11215622）的资助。（通讯作者：*Ke Shang*。）

Ke Shang 和 Zexuan Zhu 就职于深圳大学人工智能学院，地址：中国深圳 518060（电子邮箱：kshang@foxmail.com；zhuzx@szu.edu.cn）。Hisao Ishibuchi 就职于南方科技大学计算机科学与工程系，地址：中国深圳 518055（电子邮箱：hisao@sustech.edu.cn）。Qingfu Zhang 就职于香港城市大学计算机科学系，地址：中国香港特别行政区（电子邮箱：qingfu.zhang@cityu.edu.hk）。

© 2026 IEEE。允许个人使用本材料。所有其他用途均须事先获得 IEEE 的许可，无论是在当前还是未来的任何媒介中，包括为广告或宣传目的而重印或再版本材料、创作新的合集、转售或重新分发至服务器或列表，或在其他作品中重复使用本作品中任何受版权保护的部分。

I. 引言

MULTI-objective 优化（MOO）在过去几十年间受到进化计算领域的广泛关注，期间涌现了大量进化多目标优化算法（EMO） [1]–[6]。通常，这些进化多目标优化算法旨在识别一组多样化的非支配解，以充分表征或近似给定多目标优化问题的帕累托前沿（PF） [7]。

然而，随着目标数量增加，尤其是超过三个时，传统的进化多目标优化算法会因帕累托前沿维数增加而面临严峻挑战 [8], [9]。具体而言，对于目标数量极多的问题（例如超过 100 个），帕累托前沿的维数可能变得极高。因此，要有效覆盖整个帕累托前沿，所需的非支配解数量会呈指数级增长，从而带来沉重的计算负担和决策困难 [10]–[12]。

为缓解这些挑战，近期研究提出了一种称为少数应对多数（F4M）优化的新范式，也称为多目标覆盖问题（MaCP） [13], [14]。与传统的进化多目标优化方法不同，F4M优化侧重于识别一个紧凑的解集，其规模远小于目标数量（例如，针对 100 多个目标，解的数量少于 10 个），并通过相互补充共同应对所有目标。因此，紧凑解集中的至少一个解能够对每个目标实现近似优化。

这一范式的理论依据在于，人们观察到高维目标空间往往存在结构冗余和目标间相关性 [15],[16]。在许多现实世界的多目标问题中，各目标并非相互独立；相反，某些目标组具有相似的权衡模式，或可由同一设计区域同时优化。因此，与其维护数量呈指数级增长的解来逼近整个帕累托前沿，通常只需选取少量具有代表性且彼此互补的解，共同覆盖这些相关的目标区域 [13], [14], [17]。

F4M优化与众多实际场景高度相关。例如，F4M优化可用于

药物设计等领域，在这些领域中，需要一小组候选药物共同满足多样化的药理学标准 [18]，；还可用于个性化广告系统，旨在仅使用最少的广告版本，高效服务于多样化的受众群体 [19], [20]。此外，在数据挖掘和机器学习场景中，构建一组最小规模的模型，使其能够共同在各种不同任务上表现良好，通常也十分有益 [21], [22]。

尽管 F4M优化具有重要的实际意义，但目前对其研究仍相对不足。据我们所知，现有针对 F4M优化的研究成果有限。Liu 等人 [13] 正式提出了多目标覆盖问题（MaCP），证明了该问题具有 NP难性，并提出了基于聚类的群体优化器（CluSO）以及用于评估的解耦多目标测试套件（DC-MaTS）。与此同时，Lin等人 [14] 提出了切比雪夫集合标量化方法（TCH-Set标量化），将传统标量化技术扩展到高效处理数量较少的一组解所对应的大量目标，展现出令人期待的理论和实践优势。这些工作共同标志着形式化定义并解决 F4M范式迈出了重要的第一步，但也凸显出仍存在巨大差距：我们尚未建立通用的基准测试框架，也缺乏专门针对 F4M优化的有效进化算法。

本研究对 F4M 优化领域作出了以下主要贡献：

1) **一种用于 F4M 优化的新型进化算法**。我们提出了最小值之和驱动的进化多目标优化算法（*SoM-EMOA*），这是一种直接优化集合级覆盖目标的新型进化算法。与 NSGA-II、MOEA/D 或 NSGA-III 等传统进化多目标优化算法不同，这些算法使用大规模种群来逼近整个帕累托前沿，而 SoM-EMOA 旨在演化出紧凑种群($k \ll m$)，以最大化整体目标覆盖度。其关键创新包括：

- 一种集合级选择机制，根据覆盖度的边际损失移除个体，而非依据支配关系或分解；
- 一种基于档案的互补性驱动交配策略，使子代生成偏向代表性不足的目标；

这些设计共同在基于种群的搜索框架中落实了 F4M 原则。

2) **用于 F4M 优化的可扩展基准测试套件**。我们通过引入一种基于R2的变换，将低维帕累托前沿扩展为多个相关标量化目标，构建了一个全新的F4M基准测试套件。该套件基于DTLZ和WFG等成熟的多目标优化问题，能够对目标数量（m）和期望集合大小（k）进行可控扩展，从而能够系统地评估F4M算法。

3) **全面的实证研究与洞见**。我们在所提出的测试套件上开展了广泛实验，并将 SoM-EMOA 与具有代表性的基线算法进行比较，包括最先进的 EMO 算法、CluSO [13]，和 MOCOBO [23]。结果表明，SoM-EMOA 在高维目标空间中实现了卓越的覆盖质量和鲁棒性。

本文其余部分的组织如下。第二节给出了 F4M优化的形式化定义，并回顾了相关工作。第三节概述了所提出的进化算法。第四节介绍了针对 F4M优化所提出的测试套件。第五节详细描述实验设置，并讨论实证研究结果。最后，第六节总结全文，并展望未来研究的潜在方向。

II. 预备知识

本节介绍少数对多（F4M）优化问题。我们首先回顾多目标优化的基础知识，然后对 F4M问题进行形式化定义，并简要概述相关工作。

A. 多目标优化

多目标优化（MOP）旨在同时优化多个相互冲突的目标函数。从数学上讲，具有 m 个目标的 MOP 可表述如下：

$$\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})), \tag{1}$$

其中，$\mathcal{X} \subseteq \mathbb{R}^d$ 表示决策（解）空间，f_i（$\boldsymbol{x}$）：$\mathbb{R}^d \to \mathbb{R}$ 是第 i 个目标函数。通常，由于目标之间存在冲突，没有单个解能够同时优化所有目标 [24], [25]。因此，引入帕累托支配的概念来定义解的最优性：

定义 1（支配关系）。若一个解 $\boldsymbol{x}^a$ 支配另一个解 $\boldsymbol{x}^b$，记作 $\boldsymbol{x}^a \prec \boldsymbol{x}^b$，当且仅当

$$\forall i \in \{1, \ldots, m\} : f_i(\boldsymbol{x}^a) \leq f_i(\boldsymbol{x}^b)$$

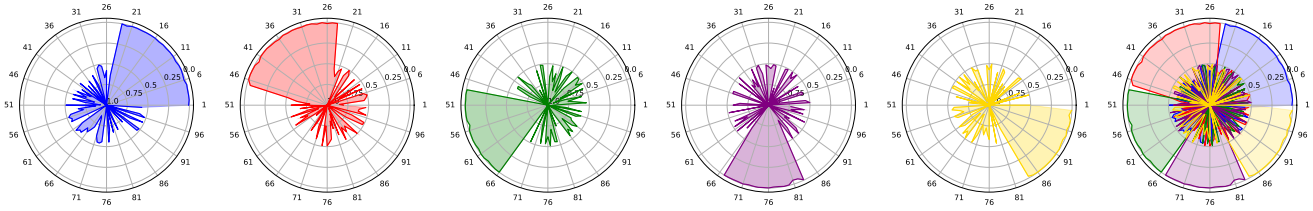
and

$$\exists j \in \{1, \ldots, m\} : f_j(\boldsymbol{x}^a) < f_j(\boldsymbol{x}^b).$$

基于支配关系，我们有以下概念 [24]:

定义 2 (帕累托最优性)。 如果不存在其他解 $\boldsymbol{x} \in \mathcal{X}$ 支配解 $\boldsymbol{x}^*$ $\boldsymbol{x}^*$，则该解为帕累托最优解。所有帕累托最优解组成的集合称为帕累托最优解集（PS），PS 在目标空间中的映像称为帕累托前沿（PF），形式化定义如下：

$$\text{PS} = \{\boldsymbol{x}^* \in \mathcal{X} | \nexists \boldsymbol{x} \in \mathcal{X} : \boldsymbol{x} \prec \boldsymbol{x}^*\}, \quad \text{PF} = \{\boldsymbol{f}(\boldsymbol{x}^*) | \boldsymbol{x}^* \in \text{PS}\}.$$



(a) 解 1 (b) 解 2 (c) 解 3 (d) 解 4 (e) 解 5 (f) 所有解

图 1. F4M优化示意图，包含 $m = 100$ 个目标和大小为 $k = 5$ 的覆盖集，转载自 [14]。每个极坐标图代表一个解，其中每个角度刻度对应一个特定的目标索引，每个径向刻度表示目标值。(a)–(e) 中的五个解以互补的方式共同处理不同的目标子集。图 (f) 展示了这五个解的并集，能够有效处理整个目标空间。

在实践中，对于现实世界问题，通常无法找到完整的帕累托最优解集。因此，诸如 NSGA-II [1], MOEA/D [2], 和 NSGA-III [3]等进化多目标优化（EMO）算法已得到广泛研究，它们通过生成有限数量的多样化且具有代表性的解来近似帕累托前沿。然而，当目标数量增加到三个以上（称为多目标优化，MaO）时，由于计算复杂度以及可视化和解释解集的困难，要实现对整个帕累托前沿的良好表示变得愈发具有挑战性 [26], [27]。

B. 少对多优化

为应对处理大量目标所面临的挑战，Liu 等人 [13] 最近提出了少对多 （F4M）优化范式，也称为多目标覆盖问题（MaCP）。F4M优化并非旨在逼近整个帕累托前沿，而是寻求一个共同在大量目标上表现良好的小型解集。

形式上，给定 m 个目标，目标是找到一个紧凑的 k 个解组成的集合（$k \ll m$），使集合中的至少一个解能够很好地处理每个目标。因此，F4M优化可视为多目标多解优化的一种特殊形式，其中，权衡是在少量候选解中，根据各目标所能达到的最佳性能进行的。图 1 展示了 F4M 优化的基本思想，即用 5 个解处理 100 个目标。

定义 3 （少解多目标（F4M）优化）。给定m 个目标函数 $f_1(\boldsymbol{x})$ $f_2(\boldsymbol{x})\ldots$， $f_m(\boldsymbol{x})$以及一个较小的整数 $k \ll m$，F4M 优化问题旨在寻找一个解集 $X_k = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(k)}\} \subseteq \mathcal{X}$，在目标之间实现总体权衡，使得每个目标 f_i 至少由 X_k中的一个解实现优化。数学上，这一一般问题可表述为：

$$\min_{X_k \subseteq \mathcal{X}} \boldsymbol{f}(X_k) = \big(f_1(X_k), f_2(X_k), \ldots, f_m(X_k)\big), \quad (2)$$

其中， $f_i(X_k) = \min_{1 \le j \le k} f_i(\boldsymbol{x}^{(j)})$表示第 i 个集合级目标函数，它反映了 X_k 在目标 f_i上的最佳性能。

当基数为 $|X_k| = 1$ 时，式 (2) 中的标准 F4M优化退化为式 (1) 中的标准多目标优化问题。换言之，经典多目标优化问题可以视为代表性解仅有一个的 F4M优化特例。

与标准多目标优化问题类似，式 (2) 中的 F4M 问题可以通过各种标量化策略转化为单目标优化问题。两种具有代表性的形式包括：

- **Weighted sum** [13]:

$$\min_{X_k \subseteq \mathcal{X}} G_{\text{ws}}(X_k) = \sum_{i=1}^{m} f_i(X_k) = \sum_{i=1}^{m} \min_{1 \le j \le k} f_i(\boldsymbol{x}^{(j)}). \quad (3)$$

  这种形式也称为最小值之和（*SoM*）方法，因其简洁且易于解释而得到广泛采用。

- **切比雪夫标量化** [14]:

$$\begin{aligned} \min_{X_k \subseteq \mathcal{X}} G_{\text{tch}}(X_k) &= \max_{1 \le i \le m} \{\lambda_i \left(f_i(X_k) - z_i^*\right)\} \\ &= \max_{1 \le i \le m} \left\{\lambda_i \left(\min_{1 \le j \le k} f_i(\boldsymbol{x}^{(j)}) - z_i^*\right)\right\}, \end{aligned} \quad (4)$$

  其中， $\lambda_i > 0$ 表示权重系数， z_i^* 表示第 i 个目标的参考（理想）值。

从这一角度看，SoM 形式（式 (3)）可视为 F4M优化的一个特例。本文采用这种形式，是因为它简洁、易于解释，且与经典覆盖问题密切相关。不过，切比雪夫标量化等其他形式也是值得未来研究的有前景的替代方案。

F4M优化通过将覆盖的概念从离散结构扩展到具有多个相互冲突目标的连续优化问题，推广并区别于经典覆盖问题（如集合覆盖 [28]和顶点覆盖 [29]）。Liu 等人 [13] 通过将 F4M优化归约为离散聚类问题，证明了 F4M优化是一个 NP-hard 问题，并凸显了其中的计算挑战。

F4M优化为多目标优化提供了一种新视角：通过直接协同优化少量解，它为依赖帕累托前沿密集覆盖的传统多目标优化策略提供了一种实用且计算高效的替代方案。

C. F4M优化的相关工作

近期有几项研究探讨了如何找到一组规模较小的解，以有效应对大量目标；在这类问题中，目标数量远远超过所需解的数量。这些研究大致可分为三类：进化算法、基于代理模型的贝叶斯优化，以及基于梯度的优化方法。

进化算法。据我们所知，唯一直接研究使用进化算法进行 F4M 优化的工作是 Liu 等人的研究 [13]，，该研究提出了 CluSO——一种基于聚类的群体优化方法。该方法明确应对了如何以少量解处理众多目标这一挑战，并引入了 DC-MaTS 测试套件，用于评估算法性能。我们的工作沿袭了这一进化方法，旨在通过提出一种专为 F4M 优化设计的新进化算法，进一步提高搜索效率和解的质量。

基于代理模型的贝叶斯优化。近期，Maus 等人 [23] 提出了多目标覆盖贝叶斯优化（MOCOBO），这是一种针对覆盖优化问题的贝叶斯优化框架；该问题等价于公式 (3) 中的 F4M 公式。MOCOBO 为每个目标构建基于高斯过程的代理模型，并设计了期望覆盖率改进（ECI）采集函数，以贪心方式构建覆盖集；同时结合局部信赖域贝叶斯优化（TuRBO 风格），以处理高维连续搜索空间以及分子、肽等结构化领域。MOCOBO 的设计旨在适应大量函数评估和众多目标，因此从概念上讲，它也适用于本文考虑的 F4M 场景。

基于梯度的算法。另一条研究路线着眼于使用基于梯度的方法解决同类问题，这类问题通常称为最小值之和（SoM）优化。Lin等人 [14]提出了切比雪夫集合标量化，这是一种新颖的标量化技术，旨在为多目标问题优化一个小规模解集。类似地，Li等人 [17] 和Ding等人 [30] 开发了基于梯度的框架，面向机器学习领域的应用，例如多任务学习 [31] 和混合提示学习 [32]。

尽管这些基于梯度的方法提出了优雅的形式化方案，并在连续情境下取得了出色的实证结果，但它们主要针对能够获取梯度信息且梯度信息可靠的应用场景。相比之下，我们的工作聚焦于黑箱优化问题或组合域，在这些问题中，梯度不可用或不适合使用 [33],[34]。因此，我们在实验研究中不考虑这些基于梯度的方法。

III. 提出的算法

本节介绍为解决 F4M 优化问题而开发的进化算法。该算法基于 ($\mu + 1$) 进化策略，整合了用于指导子代生成的档案机制，以及与 F4M 目标相一致的删除策略。

A. 框架

受广泛采用的 SMS-EMOA 框架 [35], 启发，我们提出的算法采用了一种专为高效解决 F4M 优化而设计的 ($\mu + 1$)-进化策略。该策略的核心原则是，在 F4M 优化的 SoM 目标函数（定义见式（3））的指导下，通过生成子代并有选择地删除解，迭代地演化候选解种群。

我们所提出算法的详细伪代码，即由最小值之和（SoM）驱动的进化多目标优化算法（SoM-EMOA），见算法1。

算法1 SoM-EMOA

输入：目标数量 m，种群规模 k，最大代数 T。1：初始化解集 S，使用 N 个随机解，其中 $N \gg k$。2：评估 S 中所有解的目标。3：$A =$档案集_初始化(S)。4：随机选择 k 个来自 S 的解，以构成初始种群 P。5：for $t = 1$ 到 T do 6：$\boldsymbol{x}^{p_1}, \boldsymbol{x}^{p_2} =$交配_选择($P, A$)。7：$\boldsymbol{x}^{\text{offspring}} =$变异 (交叉($\boldsymbol{x}^{p_1}, \boldsymbol{x}^{p_2}$))。8：评估 $\boldsymbol{x}^{\text{offspring}}$的目标。9：$A =$ 档案集_更新($A, \boldsymbol{x}^{\text{offspring}}$)。10：$P' = P \cup \{\boldsymbol{x}^{\text{offspring}}\}$。11：确定 $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in P'} G_{\text{ws}}($ $P' \setminus \{\boldsymbol{x}\}$)。12：更新种群：$P \leftarrow P' \setminus \{\boldsymbol{x}^*\}$。13：end for 输出：最终种群 P。

我们的算法整体框架可概括为以下几个主要步骤：

1) 在可行决策空间内随机初始化一个由 k 个候选解组成的种群 P。2) 在每一代中，使用遗传算子生成且仅生成一个子代解，并将其加入

当前种群，从而形成一个大小为 $k+1$ 的临时种群。

3）从该临时种群中谨慎地删除一个解，以确保剩余解根据式（3）中 F4M优化的目标函数，能够最优地处理所有目标。4）重复此过程，直至达到预定义的终止准则（例如最大迭代代数）。

从数学上讲，设 $P=\{\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\ldots,\boldsymbol{x}^{(k)}\}$ 为候选解的当前种群。在每次迭代中，我们生成一个子代解 $\boldsymbol{x}^{\text{offspring}}$，并构成一个临时种群：

$$P'=P\cup\{\boldsymbol{x}^{\text{offspring}}\},\quad |P'|=k+1.$$

接下来，从 P' 中删除一个解 $\boldsymbol{x}^*$，使所得种群能够最优地处理 F4M公式，具体表示如下：

$$\boldsymbol{x}^*=\arg\min_{\boldsymbol{x}\in P'} G_{\text{ws}}\left(P'\setminus\{\boldsymbol{x}\}\right).$$

下面详细介绍子代的生成方式以及删除策略的实施方法。

B. 档案初始化

在传统的进化多目标优化算法中，子代通过基于当前种群的交叉和变异生成。然而，这不适用于我们的算法，因为种群规模 k 通常很小（例如，10）。基于小规模种群很难探索搜索空间。在 SoM-EMOA 中，我们使用档案集 A 辅助子代生成。档案集 A 用于存储迄今为止针对每个目标发现的最佳解，使 SoM-EMOA 能够生成明确针对当前种群中代表性不足的目标的子代。

搜索开始时，随机生成一个规模为 $N\gg k$ 的大型临时解集 S。随后，通过对每个目标f_i（$\boldsymbol{x}$），从 S 中选择在该目标上取得最优值的解来初始化档案集：

$$A=\left\{\arg\min_{\boldsymbol{x}\in S} f_1(\boldsymbol{x}),\ldots,\arg\min_{\boldsymbol{x}\in S} f_m(\boldsymbol{x})\right\}. \tag{5}$$

这确保了每个目标在初始阶段至少有一个优秀的代表解。这种设计在概念上类似于基于分解的进化多目标优化算法 [2]中使用的逐目标精英档案，以及 ParEGO [36]等多目标贝叶斯优化框架中的初始化。该初始化提供了一组多样且有针对性的示例解，引导后续搜索转向决策空间中的潜力区域。

C. 子代生成

为了高效生成高质量的子代，我们维护一个向量 $\boldsymbol{u}\in\mathbb{R}^m$，用于存储档案集 A中每个目标的最小值，即：

$$\boldsymbol{u}=\left(\min_{\boldsymbol{x}\in A} f_1(\boldsymbol{x}),\ldots,\min_{\boldsymbol{x}\in A} f_m(\boldsymbol{x})\right), \tag{6}$$

以及一个向量 $\boldsymbol{v}\in\mathbb{R}^m$，用于存储当前种群 P 中每个目标的最小值，即：

$$\boldsymbol{v}=\left(\min_{\boldsymbol{x}\in P} f_1(\boldsymbol{x}),\ldots,\min_{\boldsymbol{x}\in P} f_m(\boldsymbol{x})\right). \tag{7}$$

因此，我们有 $\boldsymbol{u}\le\boldsymbol{v}$，而向量（$\boldsymbol{v}-\boldsymbol{u}$）反映了当前种群在处理各个目标时的质量。（$\boldsymbol{v}-\boldsymbol{u}$）中的值（即元素）越大，意味着当前种群越难以有效处理相应目标。因此，我们可以利用这些信息来改进优化过程。具体而言，我们利用信息（$\boldsymbol{v}-\boldsymbol{u}$）进行交配选择，如算法2所示。

在算法2中，我们从当前种群 P 中选择一个解 $\boldsymbol{x}^{p_1}$，并从档案集 A 中选择一个解 $\boldsymbol{x}^{p_2}$，作为两个父代。首先，从 P 中随机选择$\boldsymbol{x}^{p_1}$。为选择 $\boldsymbol{x}^{p_2}$，我们首先计算一个关于 m 个目标的概率分布 $\boldsymbol{p}$，如下所示：

$$\boldsymbol{p}=\frac{\boldsymbol{v}-\boldsymbol{u}}{\|\boldsymbol{v}-\boldsymbol{u}\|_1}, \tag{8}$$

其中，$\boldsymbol{p}$ 中的较大概率表示当前种群未能充分优化相应目标。

然后，我们根据概率分布 $\boldsymbol{p}$ 从档案集 A 中选择 $\boldsymbol{x}^{p_2}$，以便通过 $\boldsymbol{x}^{p_2}$ 增强对相应目标的优化。

算法 2 $\boldsymbol{x}^{p_1},\ \boldsymbol{x}^{p_2}=$交配_选择($P,\ A$)**输入：** 当前种群 P、档案集 A、向量 $\boldsymbol{u}\in\mathbb{R}^m$ 和向量 $\boldsymbol{v}\in\mathbb{R}^m$。1：从 P 中随机选择一个解，记为 $\boldsymbol{x}^{p_1}$。2：计算概率分布 $\boldsymbol{p}=\frac{\boldsymbol{v}-\boldsymbol{u}}{\|\boldsymbol{v}-\boldsymbol{u}\|_1}$。3：根据 $\boldsymbol{p}$ 从 A 中选择一个解，记为 $\boldsymbol{x}^{p_2}$。**输出：** 父代解 $\boldsymbol{x}^{p_1}$ 和 $\boldsymbol{x}^{p_2}$。

子代 $\boldsymbol{x}^{\text{offspring}}$ 根据 $\boldsymbol{x}^{p_1}$ 和 $\boldsymbol{x}^{p_2}$ 通过交叉和变异生成。

我们提出的交配选择方法提高了进化搜索的效率，引导搜索探索解空间中的潜力区域。

D. 档案更新

每生成并评估一个子代后，便将其在每个目标上的表现与当前档案成员进行比较，以更新档案集：

$$A=\left\{\arg\min_{\boldsymbol{x}\in A\cup\{\boldsymbol{x}^{\text{offspring}}\}} f_1(\boldsymbol{x}),\ldots,\arg\min_{\boldsymbol{x}\in A\cup\{\boldsymbol{x}^{\text{offspring}}\}} f_m(\boldsymbol{x})\right\}. \tag{9}$$

因此，只要子代在任何一个目标上有所改进，就会立即取代 A 中此前的最佳解。该机制无需存储完整的搜索历史，即可实现逐目标精英保留，并使计算开销随 m 线性增长。从概念上讲，这与 RVEA [4] 中的“参考点精英保留”以及 MOCOBO [23], 中采用的信赖域档案维护类似，但此处的操作层级是目标，而非区域。

E. 解移除策略

将子代纳入当前种群后，需要进行种群缩减步骤。SoM-EMOA在移除过程中，明确考虑了公式（3）中F4M优化的目标函数。具体而言，我们希望移除一个能够使F4M优化整体性能得到最大改善的解，即 $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in P'} G_{\text{ws}}(P' \setminus \{\boldsymbol{x}\})$。

然而，在每一代中为每个解 $\boldsymbol{x} \in P'$ 从头计算移除准则 G_{ws} 可能效率低下，尤其是在 m较大时。这是因为，对于每个解 $\boldsymbol{x} \in P'$，从头计算$G_{\text{ws}}(P' \setminus \{\boldsymbol{x}\})$的时间复杂度为 $\mathcal{O}(mk)$。因此，识别 $\boldsymbol{x}^*$ 的总时间复杂度为 $\mathcal{O}(mk^2)$。

算法3 识别 $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in P'} G_{\text{ws}}(P' \setminus \{\boldsymbol{x}\})$

输入： 当前种群 P，子代 $\boldsymbol{x}^{\text{offspring}}$, $P' = P \cup \{\boldsymbol{x}^{\text{offspring}}\}$，向量 $\boldsymbol{v}$。

1 $G^{\min} = \sum_{i=1}^{m} v_i$：计算：设置

2 $\boldsymbol{x}^* = \boldsymbol{x}^{\text{offspring}}$ 和 $I = \emptyset$

3: **对** $i \in \{1, ..., m\}$ **执行**

4: **如果** $f_i(\boldsymbol{x}^{\text{offspring}}) <= v_i$ **则**

5: $v_i' = f_i(\boldsymbol{x}^{\text{offspring}})$

6: $I = I \cup \{i\}$

7: **结束如果**

8: **结束循环**

9: **如果** $|I| == m$ **则**

10: 从 P 中随机选择一个解作为 $\boldsymbol{x}^*$

11: 更新 $\boldsymbol{v} = \boldsymbol{v}'$

12: **否则**

13: **对于** $\boldsymbol{x} \in P$，**执行**

14: **对于** $i \in \{1, ..., m\} \setminus I$，**执行**

15: **如果** $f_i(\boldsymbol{x}) == v_i$，**则**

16: $v_i' =$ 最小值$_{\boldsymbol{x}' \in P \setminus \{\boldsymbol{x}\} \cup \{\boldsymbol{x}^{\text{offspring}}\}} f_i(\boldsymbol{x}')$

17: **else**

18: $v_i' = v_i$

19: **结束如果**

20: **结束循环**

21: 计算 $G_{\text{ws}}(P' \setminus \{\boldsymbol{x}\}) = \sum_{i=1}^{m} v_i'$

22: **如果** $G_{\text{ws}}(P' \setminus \{\boldsymbol{x}\}) < G^{\min}$ **则**

23: $G^{\min} = G_{\text{ws}}(P' \setminus \{\boldsymbol{x}\})$

24: 更新 $\boldsymbol{x}^* = \boldsymbol{x}$ 和 $\boldsymbol{v}^* = \boldsymbol{v}'$

25: **结束 if**

26: **结束 for**

27: 更新 $\boldsymbol{v} = \boldsymbol{v}^*$

28: **结束 if**

输出： 解 $\boldsymbol{x}^*$ 和向量 $\boldsymbol{v}$。

为了提高识别 $\boldsymbol{x}^*$的效率，我们提出了一种高效算法，如算法3所示。算法3的主要思想是在计算 G_{ws}时避免不必要的比较。

在最佳情况下，当子代 $\boldsymbol{x}^{\text{offspring}}$ 改善所有目标（$|I| = m$ 第 9 行）时，算法仅执行常数时间的更新和随机选择，因此时间复杂度为 $\mathcal{O}(m)$。在最坏情况下，当多个个体在某些目标上取得最小值时，第 16 行可能会针对每个个体和每个目标重新扫描几乎整个种群，从而导致时间复杂度为 $\mathcal{O}(mk^2)$。在平均情况下，假设目标值是从连续分布中独立抽取的，则每个最小值以高概率都是唯一的，因此代价高昂的重新扫描预计仅会在每个目标上发生一次。这使得平均时间复杂度为 $\mathcal{O}(mk)$。

F. 独特的少数覆盖多数特征

需要强调的是，所提出的 SoM-EMOA 与传统的多目标或多目标进化算法存在根本区别。传统的进化多目标优化算法（EMO algorithms），如 NSGA-II [1], MOEA/D [2], 或 NSGA-III [3]，旨在通过维护一个规模较大且具有多样性的种群，均匀表示所有目标之间的权衡，从而逼近整个帕累托前沿。相比之下，SoM-EMOA 是依据“少数覆盖多数”原则设计的：其目标并非表示每一种权衡方向，而是寻找一个由互补解组成的紧凑解集，以共同实现对所有目标的良好覆盖度。

从算法角度来看，这种概念上的差异体现在三个鲜明的设计要素中：

1) **集合级目标评估**。 SoM-EMOA 不再仅依据帕累托支配或分解来评估个体，而是显式优化式（3）中定义的集合级覆盖目标 $G_{\text{ws}}(X_k)$。该目标衡量当前种群整体对所有 m 目标的覆盖程度，确保每个目标至少都有一个表现良好的解。因此，选择和移除算子直接作用于整个解集的覆盖度，而非个体适应度。

2) **互补性驱动选择**。传统的进化多目标优化算法通过目标空间中的多样性（例如拥挤距离、参考向量）来奖励解，而 SoM-EMOA 则通过互补性来促进多样性。如果某个候选解能够改善未得到充分覆盖的目标的覆盖度，它就具有价值。这通过概率引导交配策略（算法2）实现：子代生成会偏向当前种群中代表性不足的目标。

3) **紧凑种群演化**。传统进化多目标优化算法通常维护较大的种群（包含数百或数千个解），以近似高维帕累托前沿。SoM-EMOA,

然而，它刻意采用较小且固定的种群规模 $k \ll m$，体现了少对多设定。每个解都充当一组目标的代表性原型，类似于覆盖优化中的聚类质心 [13]。

通过这些机制，SoM-EMOA 将优化重点从逼近整个帕累托集转向构建一个小型、高效的覆盖集，以共同表示高维目标空间。这一区别体现了 F4M优化 的核心理念，并使 SoM-EMOA 有别于传统的多目标进化多目标优化框架。

IV. F4M优化基准测试套件

为了评估 F4M优化算法 的性能，采用合适的基准测试套件至关重要。本节首先讨论现有基准测试的局限性，随后介绍一个专为 F4M优化 设计的新型基于 R2 的测试套件，该套件具有更高的灵活性和挑战性。

A. 现有 *F4M*优化测试套件的局限性

尽管针对传统多目标优化已有许多成熟的基准测试套件，但它们通常并不适用于 F4M优化。

首先，诸如 DTLZ [37] 和 WFG [38] 这样的标准多目标测试套件，本质上旨在评估算法逼近整个帕累托前沿的能力。在许多此类问题中，一个解往往能够同时在大量目标上表现良好。例如，正如 Ishibuchi 等人 [39] 和 Liu 等人 [13], 所指出的，具有 $x_1 = 0$ 的单个解可以优化 DTLZ1 的前 $(m-1)$ 个目标。这种情况与现实应用并不相符，因为每个目标往往都有各自不同的最优解。因此，使用这类问题评估 F4M优化算法，可能会对其性能和泛化能力得出误导性结论。

为解决上述问题，Liu 等人 [13] 提出了解耦多目标测试套件（DC-MaTS），该套件专为 F4M 优化设计，确保每个目标的最优解彼此独立。这种设计使单个解不可能优化所有目标，从而更好地反映了 F4M 优化的本质。然而，尽管 DC-MaTS 在概念上适用，但作为基准问题，它相对简单，问题复杂度较低。例如，在 DC-MaTS 中，决策变量之间不存在相互作用，且所有目标函数均为单峰。这种简单性限制了它区分更复杂的少数对多优化求解器性能的能力。

鉴于这些局限性，我们显然需要更具挑战性且更贴近实际的测试问题来进行 F4M优化。为此，我们将在以下小节中提出一个新的基于 R2 指标的 F4M优化测试套件。该测试套件建立在 R2 指标 [40], [41] 与 F4M 优化目标之间的联系之上，具有高度灵活性，可将任何标准多目标优化问题转化为复杂度可定制且具有挑战性的 F4M优化实例。

B. R2 指标与 *F4M* 优化之间的关系

F4M 优化旨在最小化式（3）中定义的目标函数 G_{ws}。有趣的是，这种表述与 R2 指标 [40], [41],十分相似；R2 指标是进化多目标优化领域广泛使用的性能度量。具体而言，采用切比雪夫标量化时，R2 指标表示如下：

$$R_2(X, W, \boldsymbol{z}^*) = \frac{1}{|W|} \sum_{\boldsymbol{w} \in W} \min_{\boldsymbol{x} \in X} g_{\text{tch}}(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z}^*), \tag{10}$$

其中，X 是一组候选解，W 是一组权重向量，其中每个权重向量为 $\|\boldsymbol{w}\|_1 = 1$和 $\boldsymbol{w} \geq 0$，$\boldsymbol{z}^*$ 是乌托邦点，而 $g_{\text{tch}}(\cdot)$是定义如下的切比雪夫标量化函数：

$$g_{\text{tch}}(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z}^*) = \max_{1 \leq i \leq m} \{w_i |f_i(\boldsymbol{x}) - z_i^*|\}. \tag{11}$$

显然，式（3）中 F4M优化的目标 $G_{\text{ws}}(X_k)$ 和式（10）中的 R2 指标都涉及最小值求和形式。这种相似性表明，可以将 R2 指标用于 F4M优化，从而开发系统化的测试套件来评估 F4M优化算法。

C. 提出的 *F4M*优化基准测试套件

受 F4M优化的形式与 R2 指标之间密切关系的启发，我们基于 R2 指标的形式，提出了一套专为 F4M优化定制的基准测试套件。

所提出的测试套件通过以下步骤构建：

步骤 1： 使用单纯形格点设计 [42]或均匀分布 [43]等标准方法生成一组权重向量 $W = \{\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \ldots, \boldsymbol{w}^{(m)}\}$。

步骤 2： 使用与每个权重向量相对应的切比雪夫标量化，为 F4M 优化问题定义目标函数。具体而言，给定一个原始的 q目标优化问题（例如 DTLZ 和 WFG）：

$$\min_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x})),$$

我们将 F4M优化 的每个新目标函数 $F_i(\boldsymbol{x})$定义为：

$$\begin{aligned} F_i(\boldsymbol{x}) &= g_{\text{tch}}(\boldsymbol{x}|\boldsymbol{w}^{(i)}, \boldsymbol{z}^*) \\ &= \max_{1 \leq j \leq q} \{w_j^{(i)} |f_j(\boldsymbol{x}) - z_j^*|\}, i = 1, \ldots, m, \end{aligned} \tag{12}$$

其中， z^* 是原始 q目标空间中的乌托邦点。

步骤 3： 将所得的 F4M 优化问题表述为:

$$\min_{X_k=\{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(k)}\}\subseteq\mathcal{X}} \sum_{i=1}^{m} \min_{1\le j\le k} F_i(\boldsymbol{x}^{(j)}). \tag{13}$$

遵循这一流程，可以将任何现有的多目标优化问题系统地转化为 F4M 优化问题，从而大幅提升所提出测试套件的灵活性和通用性。测试套件的灵活性使我们能够系统地考察算法在各种特性下的行为。这有助于进行全面评估，并推动对 F4M 优化算法开展严格的基准测试。

D. 示例：将双目标 *DTLZ2* 问题转化为 *F4M* 实例

为说明如何在所提出的基准测试框架下将标准多目标优化问题（MOP）转化为 F4M 实例，下面以双目标 DTLZ2问题 [37] 为例。

双目标DTLZ2问题定义为

$$\min_{\boldsymbol{x}\in[0,1]^d} \begin{cases} f_1(\boldsymbol{x}) = (1+g(x_M))\cos\left(\frac{\pi}{2}x_1\right), \\ f_2(\boldsymbol{x}) = (1+g(x_M))\sin\left(\frac{\pi}{2}x_1\right), \end{cases} \tag{14}$$

其中， $g(x_M)$ 取决于最后的 $(d-1)$ 个变量。对于所有满足 $g(x_M) = 0$ 的帕累托最优解 $\boldsymbol{x}$ ，帕累托前沿满足

$$f_1^2 + f_2^2 = 1, \quad f_1, f_2 \ge 0, \tag{15}$$

也就是说，目标空间第一象限中的四分之一圆弧。

在传统的多目标优化中，NSGA-II 或 MOEA/D 等算法旨在生成沿圆弧分布的大量非支配点，以逼近这条连续的帕累托前沿。在 F4M范式下，我们不再以逼近整个前沿为目标，而是转而选取少量代表性解，使其共同覆盖更多标量化目标。

为此，我们生成 m 均匀分布的权重向量。每个权重向量 $\boldsymbol{w}^{(i)}$ 定义一个标量化目标函数 $F_i(\boldsymbol{x})$，如公式（12）所示。这里，每个 $F_i(\boldsymbol{x})$对应一个权重向量 $\boldsymbol{w}^{(i)}$。例如，若我们设置 $m = 10$ 个如图 2 所示、间距相等的权重向量，则会得到 10 个覆盖目标 F_1， F_2，……， F_{10}，它们代表四分之一圆弧上均匀分布的 10 个方向。公式（13）中的 F4M 目标旨在圆弧上或其附近寻找少量 k 个点，使每个方向 $\boldsymbol{w}^{(i)}$ 至少有一个解能在该方向上取得良好表现。例如，当 $k = 3$时，算法会倾向于找到三个大致位于圆弧上 $20°$、 $45°$和 $70°$ 处的解，如图 2 所示，因此

它们的并集能够共同提供对 10 个标量化目标的良好覆盖度。这一设定展示了在F4M公式下，如何用紧凑的互补解集近似连续帕累托前沿。

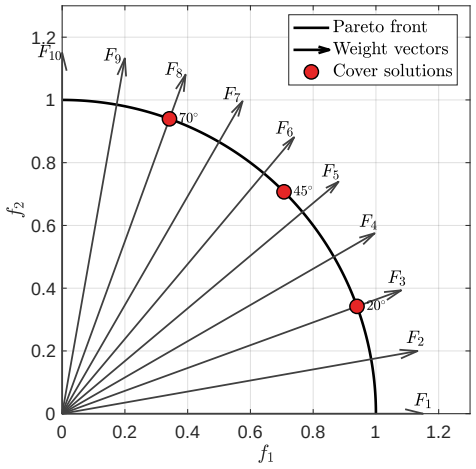


图 2. 双目标DTLZ2问题上的F4M公式示意图。四分之一圆曲线表示DTLZ2的帕累托前沿。图中展示了十个方向性标量化目标，它们从乌托邦点出发，以箭头指向前沿上均匀分布的方向。位于附近的三个代表性解共同覆盖了这些方向性目标，展示了F4M如何利用紧凑的解集近似连续的帕累托前沿。

V. 实验

本节将对我们提出的算法进行全面的实证评估。我们的主要目标是证明，与最先进的进化算法相比，所提出的算法在F4M优化中的有效性和效率。

A. 实验设置

为确保比较公平且一致，我们严格遵循 Liu 等人近期关于 F4M优化的研究中所述的实验设置；该研究介绍了CluSO算法。具体而言，我们采用相同的参数配置和性能指标。所有实验均在 PlatEMO 上进行。

1）基准问题：我们使用现有的合成基准问题，以及第IV节所述的基于R2的F4M优化测试套件。对于现有基准问题，我们选择了 [13] 提出的DC-MaTS问题，以及 [14]研究的含噪混合线性回归（NMLR）问题。对于所提出的测试套件，我们使用与一组均匀分布的权重向量相关联的标量化函数，将广泛使用的DTLZ [37] 和WFG[38]问题转换为F4M优化实例（分别记为F4M-DTLZ和F4M-WFG）。原始DTLZ和WFG问题的目标数量设为 $q = 3$。F4M-DTLZ和F4M-WFG的决策变量数量按照 q 设定为默认值。

所有F4M优化问题的目标数量均设为 $m \in \{25, 50, 75, 100\}$。

此外，我们还考虑了 [45] 提出的两个真实世界的多目标优化问题 DDMOP1 和 DDMOP4。DDMOP1 是一个车辆性能优化问题，也称为汽车驾驶室设计问题，包含 11 个决策变量和 9 个目标。决策变量描述车身的结构和尺寸参数（例如 B柱、地板侧内板和车门防撞梁的厚度），而目标则刻画重量、燃油经济性、加速时间、噪声水平和车厢宽敞度等关键性能方面。另一方面，DDMOP4 是一个电子电路设计问题，涉及一种具有九个谐振支路的 LTLCL 开关纹波抑制器。它包含 13 个决策变量和 10 个目标，其中设计变量对应于电容、电感和电阻等元件参数，目标是使电感总成本以及多个谐振频率处的谐波幅值最小化。这两个问题都具有高度复杂、黑箱且计算成本高昂的特点。

2）对比算法：我们在实验中选取以下算法进行比较。

- CluSO [13]：一种基于聚类的群体优化器，是首个专为 F4M优化提出的进化算法。
- MOCOBO[1] [23]：一种基于贝叶斯优化的方法，通过构建多个信赖域并训练高斯过程代理模型，迭代改进一小组覆盖解。
- NSGA-II [1]，PREA [46]，NRV-MOEA [47]，MOEA/D-UR [48]，和 HEA [49]：纳入比较作为参考的先进进化多目标优化算法，尽管它们并非专为 F4M优化而设计。这些算法旨在逼近 m目标问题的整个帕累托前沿，而非优化一个小型覆盖集。

需要强调的是，在参与比较的算法中，只有 CluSO、MOCOBO 和 SoM-EMOA 是专为解决 F4M优化问题而设计的，并且能够识别特定的目标函数 $G_{\text{ws}}(X_k)$。相比之下，其他进化多目标优化算法（NSGA-II、PREA、NRV-MOEA、MOEA/D-UR 和 HEA）并非针对 F4M 场景设计，也未包含任何可直接优化 $G_{\text{ws}}(X_k)$的机制。因此，它们无法有意构建一个小型且互补的解集，以共同覆盖所有目标。

3) 种群与终止条件：对于合成测试问题，种群规模设为 600，函数评估的最大次数设为 60，000。对于两个现实世界问题，种群规模设为 100，函数评估的最大次数分别设为 1000 和 10000。最终输出集的规模设为 $k \in \{5, 10\}$，以强调F4M优化：用少量解处理多个目标。对于那些传统的进化多目标优化算法，其最终输出是最终种群。为进行公平比较，我们使用算法 4 中描述的贪心子集选择算法，从最终种群中选择 k 个解，作为这些传统进化多目标优化算法的最终输出。

算法 4 贪心 F4M 子集选择

输入： 最终种群 P。

1: Initialize $S = \emptyset$

。2：**对于** $i \in \{1, ..., k\}$ **执行**3：识别 $\boldsymbol{x}^* = \text{argmin}_{\boldsymbol{x} \in P} G_{\text{ws}}(\{\boldsymbol{x}\} \cup S)$ 4：更新 $S = S \cup \{\boldsymbol{x}^*\}$ 和 $P = P \setminus \{\boldsymbol{x}^*\}$5：**结束循环 输出：** 一个子集 $S \subset P$，其具有 $|S| = k$。

4）变异算子：所有算法均采用模拟二进制交叉（SBX） [50] ，分布指数为 $\eta_c = 20$，以及多项式变异 [51]，其指数为 $\eta_m = 20$，与 [13]中的设置一致。交叉概率设为 $p_c = 1.0$，变异概率为 $p_m = 1/d$，其中 d 为决策变量的数量。

5）性能指标：根据 [13]，使用$G_{\text{ws}}(X_k)$中定义的 SoM函数评估每种算法的性能。较小的 $G_{\text{ws}}(X_k)$值表示更好的总体目标覆盖度。

每种算法均独立运行 30 次，并报告性能指标的均值和标准差，以确保统计可靠性。所有结果均采用显著性水平为 0.05 的 Wilcoxon秩和检验进行检验，以验证性能差异是否具有统计显著性。

B. 合成问题上的结果

表 I 总结了 SoM-EMOA 和六种基线算法在 5 个现有基准问题上针对 $k = 5$ 情况获得的 G_{ws}（X_k）均值（$k = 10$ 的结果见补充材料）。如表 I 所示，SoM-EMOA 在每个测试问题上均显著优于所有对比算法。具体而言，它在全部 5 个问题上都取得了最低的 G_{ws}（X_k）值，且相较于每种基线算法的改进均具有统计显著性。值得注意的是，即使是 CluSO——目前最具竞争力的 F4M求解器——也未能在任何情况下胜过 SoM-EMOA。

图3展示了所有算法在现有基准问题上，采用 $m = 50$ 和 $k = 5$时的收敛行为。对于所有实例，与基线算法相比，SoM-EMOA 在降低 G_{ws}（X_k）方面表现得更快、更稳定。这凸显了其在进化过程中发现高质量解的效率。

表II展示了针对所提出的基于 R2 的 F4M基准测试套件，在 $k = 5$ 情况下的相应结果（ $k = 10$ 的结果见

[1]https://github.com/nataliemaus/mocobo

表 I SoM-EMOA 与所有基线算法在现有基准测试问题上获得的均值 $G_{\mathrm{WS}}\left(X_k\right)$ 。最终解集的规模 k 为 5。“+”、“—”和“≈”分别表示比较算法“显著优于”、“显著劣于”和“在统计上与”SoM-EMOA“相似”。

问题	m	d	NSGA-II	PREA	NRV-MOEA	MOEA/D-UR	HEA	CluSO	SoM-EMOA
DC-MaTS1	25	25	-1.6042e+1 −	-1.6020e+1 −	-1.0927e+1 −	-1.4026e+1 −	-4.4803e+0 −	-1.7131e+1 −	-
	50	50	-2.8409e+1 −	-3.0261e+1 −	-2.1476e+1 −	-2.5645e+1 −	-2.2579e+0 −	-3.3120e+1 −	-
	75	75	-3.9786e+1 −	-3.8102e+1 −	-3.0181e+1 −	-3.6454e+1 −	-2.6148e+1 −	-4.8496e+1 −	-
	100	100	-5.0166e+1 −	-4.3673e+1 −	-3.9080e+1 −	-4.6837e+1 −	-3.5476e+1 −	-6.3067e+1 −	-
DC-MaTS2	25	25	-1.6050e+1 −	-1.6233e+1 −	-1.2472e+1 −	-1.5493e+1 −	-7.0205e+0 −	-1.7522e+1 −	-
	50	50	-2.9340e+1 −	-3.1975e+1 −	-2.5706e+1 −	-2.6801e+1 −	-1.8118e+1 −	-3.1545e+1 −	-
	75	75	-4.1662e+1 −	-3.9802e+1 −	-3.5373e+1 −	-3.7344e+1 −	-3.1877e+1 −	-4.5411e+1 −	-
	100	100	-5.2844e+1 −	-4.5429e+1 −	-4.5627e+1 −	-4.7871e+1 −	-4.2089e+1 −	-5.8587e+1 −	-
DC-MaTS3	25	25	-1.5735e+1 −	-1.4719e+1 −	-1.0936e+1 −	-1.2712e+1 −	-3.8945e+0 −	-1.6430e+1 −	-
	50	50	-2.7522e+1 −	-2.7469e+1 −	-1.7230e+1 −	-1.8153e+1 −	-3.0204e+0 −	-3.0680e+1 −	-
	75	75	-3.7624e+1 −	-3.6953e+1 −	-2.2956e+1 −	-2.7893e+1 −	-1.7042e+1 −	-4.3976e+1 −	-
	100	100	-4.6606e+1 −	-4.4198e+1 −	-2.9136e+1 −	-3.6269e+1 −	-2.5452e+1 −	-5.9525e+1 −	-
DC-MaTS4	25	25	-1.6315e+1 −	-1.7080e+1 −	-1.1722e+1 −	-1.3852e+1 −	-5.8713e+0 −	-1.6733e+1 −	-
	50	50	-2.8999e+1 −	-2.9934e+1 −	-2.0039e+1 −	-2.6205e+1 −	-2.0596e+0 −	-3.0800e+1 −	-
	75	75	-4.0553e+1 −	-3.5309e+1 −	-2.8575e+1 −	-3.9063e+1 −	-2.6617e+1 −	-4.4203e+1 −	-
	100	100	-5.1148e+1 −	-3.9836e+1 −	-3.6679e+1 −	-5.0318e+1 −	-3.3677e+1 −	-6.2014e+1 −	-
NMLR	25	10	1.8637e+0 −	9.4321e-1 −	1.2388e+0 −	1.1468e+0 −	1.2590e+0 −	6.0168e-1 −	2.0673e-1
	50	10	5.8519e+0 −	3.6759e+0 −	4.1019e+0 −	4.2455e+0 −	6.1741e+0 −	2.0714e+0 −	7.9513e-1
	75	10	1.3432e+1 −	8.9401e+0 −	1.0944e+1 −	1.0582e+1 −	1.4440e+1 −	5.1172e+0 −	2.1562e+0
	100	10	1.7722e+1 −	1.1892e+1 −	1.4738e+1 −	1.7212e+1 −	1.9174e+1 −	1.1357e+1 −	3.7980e+0
$+/-/\approx$			0/20/0	0/20/0	0/20/0	0/20/0	0/20/0	0/20/0	-

补充材料）。再次证明，SoM-EMOA 在几乎所有情况下都取得了最佳性能。在 32 个测试实例中，它在 31 个实例上排名第一。它在各种基准测试问题上始终保持优势，验证了所提出方法的强大泛化能力和可扩展性。

图4绘制了所提出的基准测试套件在 $m = 50$ 和 $k = 5$时的收敛曲线。在大多数情况下，SoM-EMOA 在相同的函数评估次数内取得了显著更低的 $G_{\mathrm{ws}}(X_k)$ 值，并且收敛速度更快。相比之下，其他算法要么过早停滞，要么收敛到较差的解集，表明它们解决高维 F4M 问题的能力有限。我们可以观察到，尽管 CluSO 专为解决 F4M优化问题而设计，但它在所有测试问题上的表现都不佳。这表明我们提出的测试套件更具挑战性，有望推动 F4M优化算法的进一步发展。

图4和表II中的另一项观察结果是，一些传统的进化多目标优化算法（如 NSGA-II 和 HEA）在若干 F4M基准问题上表现出竞争力。尽管这些算法并非专为 F4M优化而设计，但它们取得相对较好结果的原因，可以从所构造的 F4M测试问题的特性得到解释。

在我们的基准测试设计中，F4M-DTLZ 和 F4M-WFG 等问题是通过第IV节所述的基于R2的变换，由标准的三目标DTLZ和WFG问题推导而来的。在这一变换中，每个标量化F4M目标都对应于原始三目标空间中的一个加权方向，并且这些标量化目标的每个最优解都位于底层多目标优化问题的原始帕累托前沿上。因此，NSGA-II等算法——众所周知，这类算法能够有效近似三目标问题的帕累托前沿——仍然可以生成沿基础问题真实前沿分布的种群。当随后根据这些解构建F4M覆盖集时，所得覆盖质量可能显得很高，即使该算法本身并未显式优化F4M目标。

对于由低维基础问题（例如 $q = 3$）衍生、带有 m 的 F4M-DTLZ 和 F4M-WFG 实例，这一现象尤为明显，因为变换后的目标仍具有很强的相关性。在这种情况下，原始帕累托前沿本身就包含能够很好地覆盖各个派生标量化方向的点。然而，当基础目标的数量或相关程度降低时，SoM-EMOA 和 CluSO 等专用于F4M的算法的优势便愈发明显，因为它们会显式优化集合级覆盖目标。

C. 现实世界问题上的结果

表III报告了 SoM-EMOA 和七种基线算法在两个真实世界多目标优化问题 DDMOP1 和 DDMOP4 上获得的 $G_{\mathrm{ws}}(X_k)$ 均值。结果分别在两种不同的评估预算下给出，即 1000 和 10000

表II 所提出的基准测试问题上，SOM-EMOA 与所有基线算法得到的 $G_{\mathrm{WS}}(X_k)$ 均值。最终解集的大小 k 为 5。“十”、“{v9”和“≈”分别表示所比较的算法“显著优于”、“显著劣于”和“与”SOM-EMOA“统计上相近”。

问题	m	d	NSGA-II	PREA	NRV-MOEA	MOEA/D-UR	HEA	CluSO	SoM-EMOA
F4M-DTLZ1	25	7	1.0045e+0 −	1.0143e+0 −	1.0111e+0 −	1.0249e+0 −	1.0160e+0 −	1.3197e+2 −	9.4407e-1
	50	7	2.1148e+0 −	2.1359e+0 −	2.1264e+0 −	2.3078e+0 −	2.1348e+0 −	2.6285e+2 −	2.0601e+0
	75	7	3.2388e+0 −	3.2555e+0 −	3.2596e+0 −	3.5789e+0 −	3.2910e+0 −	3.8055e+2 −	3.1594e+0
	100	7	4.3598e+0 −	4.3772e+0 −	4.3861e+0 −	4.9325e+0 −	4.4170e+0 −	5.2429e+2 −	4.2456e+0
F4M-DTLZ2	25	12	2.3841e+0 −	2.3819e+0 −	2.4047e+0 −	2.4079e+0 −	2.4027e+0 −	4.2317e+0 −	2.3770e+0
	50	12	5.0278e+0 −	5.0280e+0 −	5.0814e+0 −	5.0631e+0 −	5.0565e+0 −	9.0217e+0 −	5.0055e+0
	75	12	7.6872e+0 −	7.6869e+0 −	7.7445e+0 −	7.9240e+0 −	7.7149e+0 −	1.4346e+1 −	7.6692e+0
	100	12	1.0354e+1 −	1.0347e+1 −	1.0454e+1 −	1.0665e+1 −	1.0371e+1 −	1.8970e+1 −	1.0316e+1
F4M-DTLZ3	25	12	2.9496e+0 −	1.0418e+1 −	4.8099e+0 −	2.5187e+0 −	2.5187e+0 −	1.0286e+3 −	2.3900e+0
	50	12	5.7855e+0 −	1.9300e+1 −	1.0721e+1 −	6.5438e+0 −	5.0825e+0 −	2.2005e+3 −	5.0234e+0
	75	12	8.8362e+0 −	2.7879e+1 −	1.5911e+1 −	1.2696e+1 −	7.7517e+0 −	3.2520e+3 −	7.6933e+0
	100	12	1.2081e+1 −	3.7592e+1 −	2.0053e+1 −	1.8583e+1 −	1.0424e+1 −	4.5370e+3 −	1.0352e+1
F4M-DTLZ4	25	12	2.3846e+0 −	2.3818e+0 −	2.4054e+0 −	2.3944e+0 −	2.4856e+0 −	3.4705e+0 −	2.3773e+0
	50	12	5.0281e+0 −	5.0265e+0 −	5.0894e+0 −	5.0563e+0 −	6.2525e+0 −	7.7132e+0 −	5.0043e+0
	75	12	7.6877e+0 −	7.6831e+0 −	7.7748e+0 −	7.9817e+0 −	9.5849e+0 −	1.2368e+1 −	7.6689e+0
	100	12	1.0353e+1 +	1.0345e+1	1.0463e+1 +	1.0924e+1 −	1.1163e+1 −	1.5875e+1 −	1.0476e+1
F4M-WFG1	25	12	5.1876e+0 −	6.1981e+0 −	5.5615e+0 −	4.5628e+0 −	4.5723e+0 −	1.8780e+1 −	4.3270e+0
	50	12	1.0594e+1 −	1.2581e+1 −	1.1296e+1 −	9.4794e+0 −	9.5165e+0 −	3.6270e+1 −	9.0636e+0
	75	12	1.5813e+1 −	1.8845e+1 −	1.7101e+1 −	1.4907e+1 −	1.4477e+1 −	5.5646e+1 −	1.3769e+1
	100	12	2.1259e+1 −	2.5031e+1 −	2.2606e+1 −	2.0061e+1 −	1.9507e+1 −	7.3400e+1 −	1.8475e+1
F4M-WFG2	25	12	4.6982e+0 −	4.6933e+0 −	4.7613e+0 −	4.6833e+0 −	4.6814e+0 −	7.9513e+0 −	4.5091e+0
	50	12	9.8864e+0 −	9.9091e+0 −	1.0024e+1 −	9.8948e+0 −	9.9535e+0 −	1.7959e+1 −	9.6015e+0
	75	12	1.4989e+1 −	1.5046e+1 −	1.5239e+1 −	1.5055e+1 −	1.5089e+1 −	2.7461e+1 −	1.4602e+1
	100	12	2.0098e+1 −	2.0171e+1 −	2.0405e+1 −	2.0175e+1 −	2.0189e+1 −	3.6695e+1 −	1.9637e+1
F4M-WFG3	25	12	1.0055e+1 −	1.0139e+1 −	1.0334e+1 −	1.0278e+1 −	1.0376e+1 −	1.2404e+1 −	9.8869e+0
	50	12	2.1681e+1 −	2.1863e+1 −	2.2180e+1 −	2.2644e+1 −	2.2201e+1 −	2.6348e+1 −	2.1094e+1
	75	12	3.2966e+1 −	3.3307e+1 −	3.3734e+1 −	3.5368e+1 −	3.3751e+1 −	4.0618e+1 −	3.1778e+1
	100	12	4.4319e+1 −	4.4562e+1 −	4.5289e+1 −	4.7456e+1 −	4.5413e+1 −	5.2939e+1 −	4.2688e+1
F4M-WFG4	25	12	7.7557e+0 −	7.7475e+0 −	7.8386e+0 −	7.8083e+0 −	7.7003e+0 −	1.0602e+1 −	7.6640e+0
	50	12	1.6446e+1 −	1.6409e+1 −	1.6609e+1 −	1.6540e+1 −	1.6638e+1 −	2.3528e+1 −	1.6294e+1
	75	12	2.5069e+1 −	2.5042e+1 −	2.5318e+1 −	2.5483e+1 −	2.5245e+1 −	3.7768e+1 −	2.4913e+1
	100	12	3.3648e+1 −	3.3633e+1 −	3.4006e+1 −	3.4392e+1 −	3.3972e+1 −	5.1221e+1 −	3.3445e+1
$+/-/\approx$			1/31/0	1/31/0	1/31/0	0/32/0	0/32/0	0/32/0	-

表III SOM-EMOA 和所有基线算法在真实世界测试问题上获得的 $G_{\mathrm{WS}}(X_k)$ 均值。最终解集的大小 k 为 5。“十”、“—”和“≈”分别表示所比较的算法“显著优于”、“显著劣于”和“在统计上与”SOM-EMOA“相似”。

问题	FEs	NSGA-II	PREA	NRV-MOEA	MOEA/D-UR	HEA	CluSO	MOCOBO	SoM-EMOA
DDMOP1	1000	-9.2519e+1 −	-8.6229e+1 −	-6.1651e+1 −	-1.0671e+2 −	-8.8777e+1 −	-1.0850e+2 −	-9.3182e+1 −	-
	10000	-1.2945e+2 −	-1.3024e+2 −	-1.0246e+2 −	-1.2763e+2 −	-1.0292e+2 −	-1.0786e+2 −	-	-
DDMOP4	1000	-4.4692e+3 −	-4.4662e+3 −	-4.4540e+3 −	-4.5150e+3 −	-4.4400e+3 −	-	-4.5238e+3 −	-4.5280e+3
	10000	-4.5350e+3 −	-4.5332e+3 −	-4.5070e+3 −	-4.5380e+3 −	-4.5115e+3 −	-4.5349e+3 −	-	-
$+/-/\approx$		0/4/0	0/4/0	0/4/0	0/4/0	0/4/0	1/3/0	0/2/0	-

次函数评估（FEs），最终解集大小固定为 $k=5$。

在所有问题和评估预算下，SoM-EMOA始终取得最佳或在统计上相当的 $G_{\mathrm{ws}}(X_k)$值。对于DDMOP1，在1000和10000次函数评估下，SoM-EMOA的覆盖度指标均为所有算法中最低（最佳）。这些结果表明即使评估次数非常有限，SoM-EMOA也能有效发现一组紧凑的互补解，共同覆盖多样化的目标。

对于更复杂的 DDMOP4 问题，结果呈现出类似的趋势。在 1000 次函数评估时，SoM-EMOA 的表现与 CluSO 相当；在 10000 次函数评估时则略胜一筹，

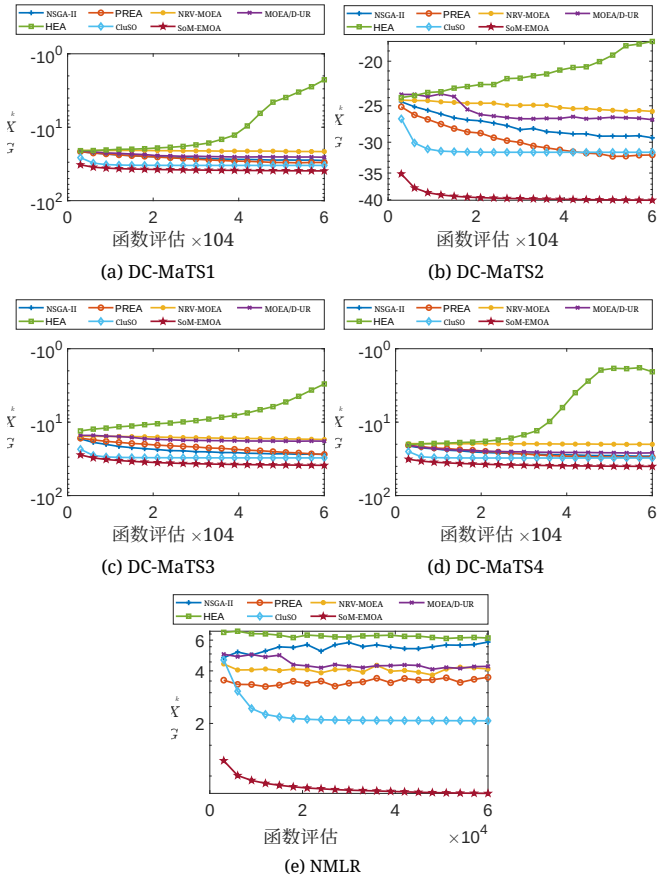


图3. 5个现有 F4M 优化基准问题上不同算法的收敛曲线。目标数量为 $m = 50$，解集大小为 $k = 5$。

取得了最佳均值 $G_{\text{ws}}(X_k)$值。值得注意的是，CluSO 在 1000 次函数评估时略占优势，这表明其基于聚类的机制可能在初期探索中带来小幅收益。然而，随着评估次数增加，SoM-EMOA 收敛至最佳性能，凸显了其在不同代数中的卓越搜索效率和鲁棒性。

对于 MOCOBO 算法，仅有 1000 次函数评估预算下的结果。由于其代理建模和信赖域贝叶斯优化设计，当函数评估次数增加到 10000 次时，MOCOBO 的计算开销变得难以承受，因此无法在合理时间内获得其结果。尽管如此，即使在 1000 次函数评估下，SoM-EMOA 在 DDMOP1 和 DDMOP4 上的性能也显著优于 MOCOBO，展现了所提出的进化方法在计算效率和评估次数受限的场景中的优势。

D. 运行时间比较

图5展示了所有算法在合成测试问题和真实世界测试问题上的运行时间对比。从这些结果中可以得出若干观察结论。

如图5(a)所示，NSGA-II 在所有算法中平均运行时间最短，HEA 和 CluSO 紧随其后。相比之下，PREA、NRV-MOEA 和 MOEA/D-UR 的计算成本较高，这是因为它们的分解和偏好学习组件更为复杂。与轻量级算法（如 NSGA-II 和 HEA ）相比，SoM-EMOA 的运行时间开销适中，这是预期之中的，因为它还执行额外的集合级覆盖度评估和档案管理操作。尽管如此，SoM-EMOA 仍明显快于大多数基于偏好的基线算法和基于分解的基线算法。由于 MOCOBO 的代理建模和信赖域更新需要大量计算，因此未将其纳入此次比较；当评估预算设为 60000 次函数评估时，MOCOBO 无法在合理时间内完成。

真实世界基准测试中的运行时间比较如图5(b)所示。当评估预算为 1000 次函数评估时，MOCOBO 虽能得出结果，但其运行时间比所有其他算法高出多个数量级，这反映了反复进行高斯过程训练和信赖域优化所带来的巨大计算开销。在 10000 次函数评估的情况下，由于运行时间过长，无法获得 MOCOBO 的结果。在其余算法中，NSGA-II 的执行速度仍然最快，其次是 HEA 和 CluSO，而 SoM-EMOA 的开销则保持在适中水平。这表明，SoM-EMOA 为基于覆盖的选择和档案更新引入的额外操作并未显著损害其计算效率。

E. 消融研究

1) 档案集的有效性 A：为研究档案集 A 在引导进化过程中的有效性，我们通过比较原始 SoM-EMOA 与移除档案集 A 的变体（记为 SoM-EMOA-$\not{A}$）进行消融研究。在该变体中，子代解仅由当前种群生成，不使用 A 中的解。

表 IV 展示了 SoM-EMOA 及其无档案变体 SoM-EMOA-$\not{A}$ 在 13 个基准问题上的 $G_{\text{ws}}(X_k)$ 均值。目标数量固定为 $m = 50$，最终解集的大小为 $k = 5$。结果清楚表明，档案集对于提升性能至关重要。具体而言，SoM-EMOA 在 13 个问题中的 12 个上优于其无档案变体，而在 1 个问题上的性能在统计上相近。

这些结果证实了将档案集 A 作为高质量特定目标代表解的记忆库加以维护的重要性。通过引导子代生成偏向这些档案解，该算法有效确保了所有目标都得到系统性处理，并提高了发现高质量解的可能性。

2）概率分布的有效性 $\boldsymbol{p}$：为了评估概率分布 $\boldsymbol{p}$ 在

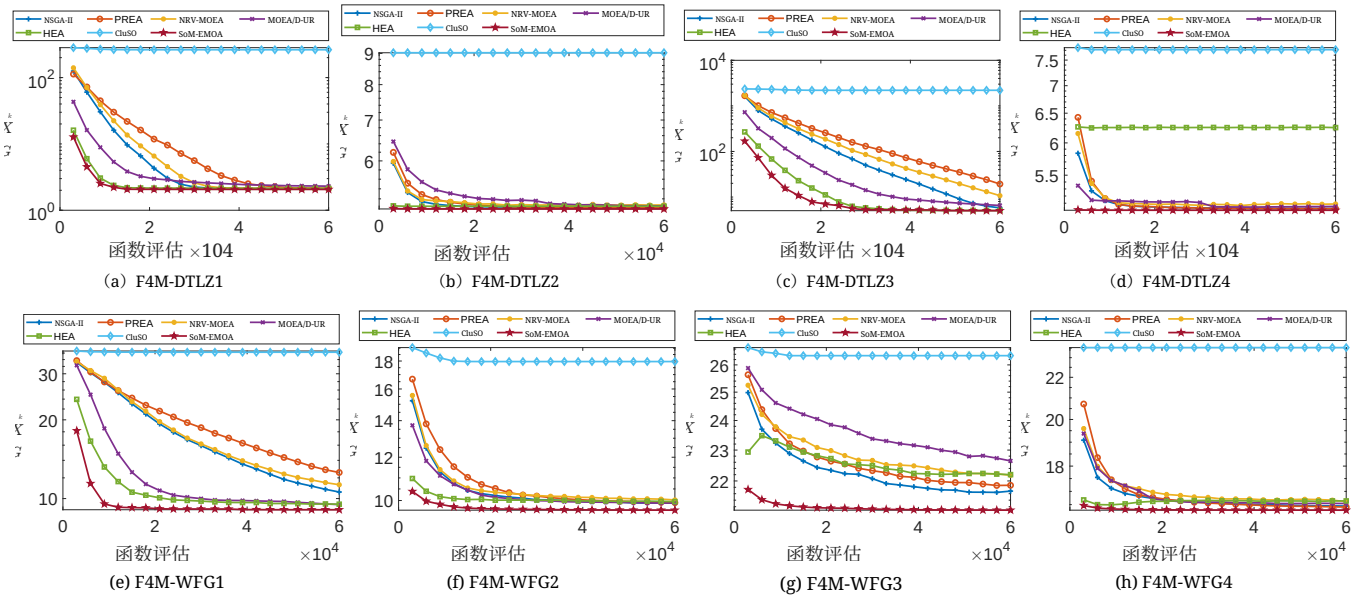


图4. 不同算法的收敛曲线以及解集的大小 $k = 5$。在所提出的 F4M 优化基准问题上的表现。目标数量为 $m = 50$

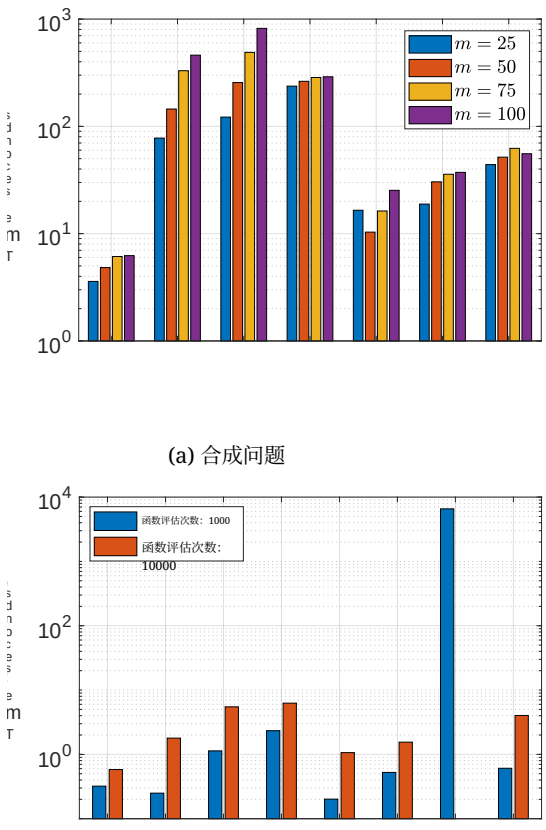


图5. 不同算法在合成问题和现实世界问题上的运行时间（秒）比较。每个柱形表示某一算法在所有问题上的平均运行时间，$k = 5$。

算法2中的交配选择过程中所发挥的作用，我们将原始 SoM-EMOA 与一个不使用 $\boldsymbol{p}$ 的变体进行比较（记为 SoM-EMOA-$\not{\boldsymbol{p}}$）。在该变体中，第二个父代解从档案集 A 中均匀随机选取，而不考虑当前种群未能充分处理哪些目标。

如表 IV 所示，在大多数基准问题上，SoM-EMOA 的性能始终优于其变体 SoM-EMOA-$\not{\boldsymbol{p}}$。具体而言，SoM-EMOA 在 13 个问题中的 5 个上优于 SoM-EMOA-$\not{\boldsymbol{p}}$，在 7 个问题上表现具有统计相似性，仅在 1 个问题上表现较差。这些结果表明，纳入 $\boldsymbol{p}$ 能够带来显著优势。

这一改进源于 $\boldsymbol{p}$的自适应特性，它对每个目标的覆盖不足进行编码。通过概率性地优先选择针对覆盖最不足目标的档案解，交配过程变得更有依据且目标更加明确。因此，使用 $\boldsymbol{p}$ 有助于发现更多互补解，最终提升最终解集中所有目标的覆盖度。

F. 局限性与讨论

尽管 SoM-EMOA 在多种 F4M基准问题上表现出有竞争力的性能，但它也继承了随机进化搜索固有的若干局限性。

初始种群的质量以及算法参数的选择（例如变异率、档案大小和种群规模 k）会显著影响最终的覆盖度表现。由于 SoM-EMOA 专为小规模种群（$k \ll m$）设计，不理想的初始化可能会限制其早期的探索能力

表 IV

G_{WS}（由 SoM-EMOA 及其两个变体获得（SoM-EMOA-$\cancel{A}$ 表示移除了档案集 A，SoM-EMOA-$\cancel{p}$ 表示未使用概率分布 $\boldsymbol{p}$）。目标数量 m 为 50，最终解集大小 k 为 5。“$+$”、“$-$”和“$\approx$”分别表示被比较算法“显著优于”、“显著劣于”和“与” SoM-EMOA“统计上相似”。

问题	SoM-EMOA-$\cancel{A}$	SoM-EMOA-$\cancel{p}$	SoM-EMOA
DC-MaTS1	-2.7397e+1 $-$	-3.7724e+1 $-$	-
DC-MaTS2	-3.0596e+1 $-$	-3.9094e+1 $-$	-
DC-MaTS3	-2.5320e+1 $-$	-3.6205e+1 $-$	-
DC-MaTS4	-2.8279e+1 $-$	-3.8391e+1 $-$	-
NMLR	8.8613e-1 $-$	8.4252e-1 $\approx$	7.9513e-1
F4M-DTLZ1	2.0644e+0 $-$	2.0627e+0 $\approx$	2.0601e+0
F4M-DTLZ2	5.0129e+0 $-$	5.0048e+0	5.0055e+0
F4M-DTLZ3	5.0461e+0 $-$	5.0249e+0 $\approx$	5.0234e+0
F4M-DTLZ4	6.7652e+0 $-$	8.3773e+0 $-$	5.0043e+0
F4M-WFG1	9.9371e+0 $-$	9.1178e+0 $\approx$	9.0636e+0
F4M-WFG2	9.6334e+0 $-$	9.6107e+0 $\approx$	9.6015e+0
F4M-WFG3	2.1577e+1 $-$	2.1159e+1 $\approx$	2.1094e+1
F4M-WFG4	1.6294e+1 $\approx$	1.6292e+1	1.6294e+1
$+/-/\approx$	0/12/1	1/5/7	

代数。两种策略有助于减轻随机变异性：

- **与基于梯度搜索的混合**。引入基于梯度的搜索（如 [14]，[52] 中所示）可以提高收敛速度，并减少随机评估次数。
- **自适应种群控制**。根据覆盖率改进情况动态调整种群或档案集规模，可以稳定收敛并避免过早停滞。

本质上，SoM-EMOA 的随机特性赋予了探索灵活性，但也引入了可能影响可重复性和一致性的随机性。认识到这些局限性，对于未来将进化搜索与自适应、基于梯度或确定性组件相结合，以实现 F4M 优化的鲁棒性和效率，至关重要。

VI. 结论

本文提出了一种用于 F4M 优化的高效进化算法，旨在找到一组规模较小、能够共同处理大量目标的解。不同于寻求整个帕累托前沿多样化近似的传统 EMO 算法，我们的方法针对 F4M 的目标进行了设计，即以最少的冗余实现覆盖。

我们还为F4M优化引入了一套新的基于R2的基准测试套件。通过利用F4M优化目标与R2 指标之间的结构相似性，我们构建了一套灵活的测试套件，可将标准多目标优化问题转化为具有挑战性的F4M优化实例。这克服了现有基准测试的局限性例如DTLZ、WFG以及过于简单的DC-MaTS套件。

在多种包含多达 100 个目标的 F4M优化实例上的实验结果表明，与 CluSO、MOCOBO 以及若干最先进的 EMO 算法相比，我们的算法性能更优。该算法在各类测试问题上始终表现更佳。

未来的工作可以将所提出的 SoM-EMOA 框架扩展到在线优化和零阶优化场景。例如，在线变体可以引入事件触发更新和时变档案，以处理流式或非平稳目标；这一思路受到具有时变约束的分布式多臂老虎机凸优化领域相关进展的启发 [53],[54]。此外，SoM-EMOA 天然适用于零阶（老虎机）反馈，因为它仅依赖函数评估；集成单点或随机扰动技术，可以在存在噪声或成本高昂的环境中实现更具样本效率的探索。这些扩展将进一步拓宽 F4M优化在实时和数据受限场景中的适用性。

参考文献

[1] K. Deb、A. Pratap、S. Agarwal 和 T. Meyarivan，“一种快速且精英化的多目标遗传算法：NSGA-II”，*IEEE*进化计算汇刊，第 6 卷，第 2 期，第 182–197 页，2002 年。[2] Q. Zhang 和 H. Li，“MOEA/D：一种基于分解的多目标进化算法”，*IEEE*进化计算汇刊，第 11 卷，第 6 期，第 712–731 页，2007 年。[3] K. Deb 和 H. Jain，“采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：求解带箱约束的问题”，*IEEE*进化计算汇刊，第 18 卷，第 4 期，第 577–601 页，2013 年。[4] R. Cheng、Y. Jin、M. Olhofer 和 B. Sendhoff，“面向多目标优化的参考向量引导进化算法”，*IEEE*进化计算汇刊，第 20 卷，第 5 期，第 773–791 页，2016 年。[5] Y. Tian、X. Li、H. Ma、X. Zhang、K. C. Tan 和 Y. Jin，“基于深度强化学习的进化多目标优化自适应算子选择”，*IEEE*新兴计算智能主题汇刊，第 7 卷，第 4 期，第 1051–1064 页，2022 年。[6] T. Huang、S. Wang 和 K. Li，“基于直接偏好的对决赌博机进化多目标优化”，《神经信息处理系统进展》，第 37 卷，第 122 206–122 258 页，2024 年。[7] Y. Hua、Q. Liu、K. Hao 和 Y. Jin，“不规则帕累托前沿多目标优化问题的进化算法综述”，*IEEE/CAA*自动化学报，第 8 卷，第 2 期，第 303–318 页，2021 年。[8] S. Yang、M. Li、X. Liu 和 J. Zheng，“一种面向多目标优化的基于网格的进化算法”，*IEEE*进化计算汇刊，第 17 卷，第 5 期，第 721–736 页，2013 年。[9] H. Wang、Y. Jin 和 X. Yao，“多目标优化中的多样性评估”，*IEEE*控制论汇刊，第 47 卷，第 6 期，第 1510–1522 页，2016 年。[10] H. Ishibuchi、N. Tsukamoto 和 Y. Nojima，“进化多目标优化：简要综述”，载于*2008*年*IEEE*进化计算大会。IEEE，2008 年，第 2419–2426 页。[11] A. L´opez Jaimes 和 C. A. Coello Coello，“多目标问题：挑战与方法”，施普林格计算智能手册，第 1033–1046 页，2015 年。[12] H. Sato 和 H. Ishibuchi，“进化多目标优化：难点、方法与讨论”，*IEEJ*电气与电子工程汇刊，第 18 卷，第 7 期，第 1048–1058 页，2023 年。

[13] Y. Liu、C. Lu、X. Lin 和 Q. Zhang，“多目标覆盖问题：发现少量解以覆盖多个目标”，载于自然界并行问题求解国际会议。施普林格，2024 年，第 68–82 页。[14] X. Lin、Y. Liu、X. Zhang、F. Liu、Z. Wang 和 Q. Zhang，“以少覆盖多：面向多目标优化的切比雪夫集合标量化”，arXiv 预印本 arXiv:2405.19650，2024 年。[15] H. Ishibuchi、N. Akedo 和 Y. Nojima，“多目标进化算法在多目标背包问题上的表现”，IEEE进化计算汇刊，第 19 卷，第 2 期，第 264–283 页，2014 年。[16] Y. Yuan、Y.-S. Ong、A. Gupta 和 H. Xu，“多目标优化中的目标约简：进化多目标方法与综合分析”，IEEE进化计算汇刊，第 22 卷，第 2 期，第 189–210 页，2017 年。[17] Z. Li、T. Li、V. Smith、J. Bilmes 和 T. Zhou，“多目标多解传输”，arXiv 预印本 arXiv:2403.04099，2024 年。[18] J. S. Angelo、I. A. Guedes、H. J. Barbosa 和 L. E. Dardenne，“多目标与多目标优化：从头药物设计的现状与未来”，《化学前沿》，第 11 卷，1288626，2023 年，第 页。[19] S. C. Matz、M. Kosinski、G. Nave 和 D. J. Stillwell，“心理定向作为一种有效的数字化大规模说服方法”，《美国国家科学院院刊》，第 114 卷，第 48 期，第 12 714–12 719 页，2017 年。[20] D. Eckles、R. F. Kizilcec 和 E. Bakshy，“利用同伴鼓励设计估计网络中的同伴效应”，《美国国家科学院院刊》，第 113 卷，第 27 期，第 7316–7322 页，2016 年。[21] T. Standley、A. Zamir、D. Chen、L. Guibas、J. Malik 和 S. Savarese，“在多任务学习中，哪些任务应该一起学习？”，载于国际机器学习会议。PMLR，2020 年，第 9120–9132 页。[22] S. Wu、T. Li、Z. Charles、Y. Xiao、Z. Liu、Z. Xu 和 V. Smith，“Motley：联邦学习中异质性与个性化的基准评估”，arXiv 预印本 arXiv:2206.09262，2022 年。[23] N. Maus、K. Kim、Y. Zeng、H. T. Jones、F. Wan、M. D. T. Torres、C. de la Fuente-Nunez 和 J. R. Gardner，“多目标覆盖贝叶斯优化（MOCOBO）”，2025 年。 [在线]。可访问：https://arxiv.org/abs/2501 19342 。K. Miettinen，[24]《非线性多目标优化》。施普林格科学与商业媒体，1999 年，第 12 卷。[25] K. Deb、K. Sindhya 和 J. Hakanen，“多目标优化”，载于《决策科学》。CRC出版社，2016 年，第 161–200 页。[26] K. S. Bhattacharjee、H. K. Singh、M. Ryan 和 T. Ray，“弥合差距：多目标优化与知情决策”，IEEE进化计算汇刊，第 21 卷，第 5 期，第 813–820 页，2017 年。[27] B. Dy、N. Ibrahim、A. Poorthuis 和 S. Joyce，“改进可视化设计以实现有效的多目标决策”，IEEE可视化与计算机图形学汇刊，第 28 卷，第 10 期，第 3405–3416 页，2021 年。[28] C. Contardo 和 A. Hertz，“一类几何集合覆盖问题的精确算法”，《离散应用数学》，第 300 卷，第 25–35 页，2021 年。[29] Y. Zhang、X. Mu、X.-W. Liu、X. Wang、X. Zhang、K. Li、T. Wu、D. Zhao 和 C. Dong，“将量子近似优化算法应用于最小顶点覆盖问题”，《应用软计算》，第 118 卷，108554 页，2022 年。[30] L. Ding、Z. Chen、X. Wang 和 W. Yin，“最小值之和优化的高效算法”，载于第41届国际机器学习会议论文集，2024 年，第 10 927–10 959 页。[31] X. Lin、H.-L. Zhen、Z. Li、Q.-F. Zhang 和 S. Kwong，“帕累托多任务学习”，《神经信息处理系统进展》，第 32 卷，2019 年。[32] G. Qin 和 J. Eisner，“学习如何提问：使用软提示混合向语言模型发起查询”，arXiv 预印本 arXiv:2104.06599，2021 年。[33] I. Bajaj、A. Arora 和 M. F. Hasan，“黑箱优化：方法与应用”，载于《黑箱优化、机器学习与无免费午餐定理》。施普林格，2021 年，第 35–65 页。[34] B. H. Korte、J. Vygen、B. Korte 和 J. Vygen，《组合优化》。施普林格，2011 年，第 1 卷。[35] N. Beume、B. Naujoks 和 M. Emmerich，“SMS-EMOA：基于支配超体积的多目标选择”，《欧洲运筹学杂志》

[36] J. Knowles，《ParEGO：一种用于昂贵多目标优化问题的在线景观近似混合算法》，《IEEE进化计算汇刊》，第10卷，第1期，第50–66页，2006年。[37] K. Deb、L. Thiele、M. Laumanns和E. Zitzler，《用于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》。施普林格，2005年，第105–145页。[38] S. Huband、P. Hingston、L. Barone和L. While，《多目标测试问题综述与可扩展测试问题工具包》，《IEEE进化计算汇刊》，第10卷，第5期，第477–506页，2006年。[39] H. Ishibuchi、Y. Setoguchi、H. Masuda和Y. Nojima，《基于分解的多目标算法的性能高度依赖于帕累托前沿形状》，《IEEE进化计算汇刊》，第21卷，第2期，第169–190页，2016年。[40] A. Jaszkiewicz和P. Zielniewicz，《R2多目标质量指标的精确计算及其性质》，《IEEE进化计算汇刊》，2024年。[41] L. Sch¨apermeier和P. Kerschke，《重新研究R2指标：通过积分实现帕累托一致性》，载于《自然界并行问题求解国际会议》。施普林格，2024年，第202–216页。[42] I. Das和J. E. Dennis，《法向边界交点法：在非线性多准则优化问题中生成帕累托曲面的新方法》，《SIAM优化期刊》，第8卷，第3期，第631–657页，1998年。[43] B. S. Saini、H. K. Singh、B. Shavazipour和K. Miettinen，《均匀表示帕累托前沿的高效迭代方法》，载于《进化多准则优化国际会议》。施普林格，2025年，第241–256页。[44] Y. Tian、R. Cheng、X. Zhang和Y. Jin，《PlatEMO：用于进化多目标优化的MATLAB平台 [教育论坛],》，《IEEE计算智能杂志》，第12卷，第4期，第73–87页，2017年。[45] C. He、Y. Tian、H. Wang和Y. Jin，《面向数据驱动进化多目标优化的真实世界数据集库》，《复杂与智能系统》，第6卷，197，2020年1，189第。[46] J. Yuan、H.-L. Liu、F. Gu、Q. Zhang和Z. He，《研究指标的性质及一种使用潜力区域的进化多目标算法》，《IEEE进化计算汇刊》，第25卷，第1期，第75–86页，2020年。[47] Y. Hua、Q. Liu和K. Hao，《自适应法向量引导的进化多目标与多目标优化》，《复杂与智能系统》，第10卷，第3期，第3709–3726页，2024年。[48] L. R. De Farias和A. F. Ara´ujo，《一种在需要时更新权重的基于分解的多目标进化算法》，《群体与进化计算》，第68卷，100980，2022年 p.。[49] Z. Liu、F. Han、Q. Ling、H. Han和J. Jiang，《一种基于超支配度的多目标优化进化算法》，《群体与进化计算》，第83卷，101411，2023年 p.。[50] L. Pan、W. Xu、L. Li、C. He和R. Cheng，《用于旋转多目标优化的自适应模拟二进制交叉》，《群体与进化计算》，第60卷，第100759页，2021年。[51] J. L. Carles-Bou和S. F. Gal´an，《NSGA-II中的自适应多项式变异》，《软计算》，第27卷，17 727，2023年23，17 711第。[52] K. Zhang、L. Xu、X. Yi、Z. Ding、K. H. Johansson、T. Chai和T. Yang，《面向网络资源分配的预定义时间分布式多目标优化》，《中国科学：信息科学》，第66卷，第7期，第170204页，2023年。[53] K. Zhang、X. Yi、G. Wen、M. Cao、K. H. Johansson、T. Chai和T. Yang，《具有时变约束的分布式事件触发多臂老虎机凸优化》，《IEEE网络系统控制汇刊》，2025年。[54] K. Zhang、L. Xu、X. Yi、G. Wen、L. Xie、T. Chai和T. Yang，《具有时变约束的分布式多臂老虎机凸优化中的单点采样》，《IEEE网络系统控制汇刊》，2025年。

，第181卷，第3期，第1653–1669页，2007年。