高效应对百万维多目标问题：一种方向采样与微调方法

Haokai Hong、Min Jiang，*IEEE*高级会员，Qiuzhen Lin、*IEEE*高级会员，以及Kay ChenTan，*IEEE*会士

arXiv:2304.04067v2 [cs.NE] 8 Apr 2024

摘要—**我们将超大规模多目标优化问题定义为决策变量超过100,000个的多目标优化问题（VLSMOPs）。鉴于现实世界场景中普遍需要优化数十万乃至数百万个变量，这些问题具有重要意义。然而，VLSMOPs中更大的维度加剧了维数灾难，并给现有的大规模演化多目标优化算法带来了重大挑战，使得这些问题更难在实际计算资源的约束下求解。为克服这一问题，我们提出了一种名为超大规模多目标优化框架（VMOF）的新方法。该方法在超大规模空间中高效地采样一般但适用的演化方向，随后对这些方向进行微调，以定位帕累托最优解。为了针对不同的解采样最合适的演化方向，我们采用汤普森采样，因为它能够在有限的历史评估次数下，有效地从数量极其庞大的选项中进行推荐。此外，我们设计了一种针对帕累托最优解跟踪的方向微调技术。为了理解所设计的框架，我们首先对该框架进行了分析，然后使用广泛认可的基准测试和现实世界问题对VMOF进行了评估，这些问题的维度范围为100至1,000,000。实验结果表明，与现有算法相比，我们的方法不仅在大规模多目标优化问题（LSMOPs）上表现出优越性能，在超大规模多目标优化问题（VLSMOPs）上同样如此。**

索引词—**进化算法、多目标优化、大规模优化、**

I. 引言

多目标优化问题（多目标优化问题）具有多个相互冲突的目标，并在各种现实世界领域中得到广泛应用 [1], [2]。在科学和工程领域，存在着各种各样的多目标优化问题，其特点是包含大量决策变量。这些这类问题通常称为大规模多目标优化问题（LSMOPs） [3], [4]。“维数灾难”是传统多目标进化算法（MOEAs）在处理大规模多目标优化问题时面临的一项重大挑战。这一挑战源于决策空间的指数级增长，导致其收敛性比数学规划方法更慢 [3], [5]。

为缓解维数灾难，近年来人们付出了大量努力，专门针对大规模多目标优化问题设计大规模多目标进化算法 [5]–[9]。一个值得关注的例子是 Antonio 等人 [10],提出的 CCGDE3 算法，该算法通过维持多个等长子组来实现决策变量分组。Zille 等人开发了 WOF [11], ，采用分组机制和基于种群的优化器，分别处理原始变量和权重变量。另一种创新方法是 Tian 等人提出的 LMOCSO [12] ，该方法设计了一种利用竞争群优化器的新型搜索策略。

尽管对大规模多目标优化问题的研究已取得长足进展，但现有大多数方法主要针对维度少于100,000的问题而设计。然而，遇到决策变量超过十万的多目标优化问题并不罕见。例如，在当代电力输送系统中，电压互感器呈现出时变特性，从而导致一个涉及超过100,000个决策变量的时变比差估计问题 [13]。此类高维问题超-越了当前算法的能力范围。

鉴于上述讨论，我们提出了一类新的优化问题，称为超大规模多目标优化问题（VLSMOPs），其特征是拥有超过100,000个决策变量。解决超大规模多目标优化问题的主要挑战源于通常可用的有限计算资源与庞大搜索空间之间的矛盾，这使得定位帕累托最优前沿（PF）的任务变得复杂——这是维数灾难相较于传统大规模多目标优化问题加剧的体现 [14] 。因此，解决超大规模多目标优化问题的关键在于高效识别演化方向，并有效探索帕累托最优前沿。

为应对超大规模多目标优化问题带来的挑战，我们提出了一种专用框架，称为超大规模多目标优化框架（VMOF），该框架采用了进化方向采样和微调技术。我们的框架专为解决超大规模多目标优化问题中遇到的维数灾难而设计

© 2024 IEEE。允许将本材料用于个人用途。对于所有其他用途，包括在任何当前或未来的媒体中重印/再版本材料、将本材料用于广告或宣传目的、创作新的合集作品、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的任何受版权保护的组成部分，必须获得IEEE的许可。H. Hong 和 M. Jiang 隶属于福建省厦门市 361005 的厦门大学信息学院、文化和旅游部福建与台湾非物质文化遗产数字化保护与智能处理重点实验室人工智能系。 *(*通讯作者：*M. Jiang)*Q. Z. Lin 隶属于中国深圳大学计算机科学与软件工程学院。K. C. Tan 隶属于中国香港特别行政区香港理工大学计算学系。

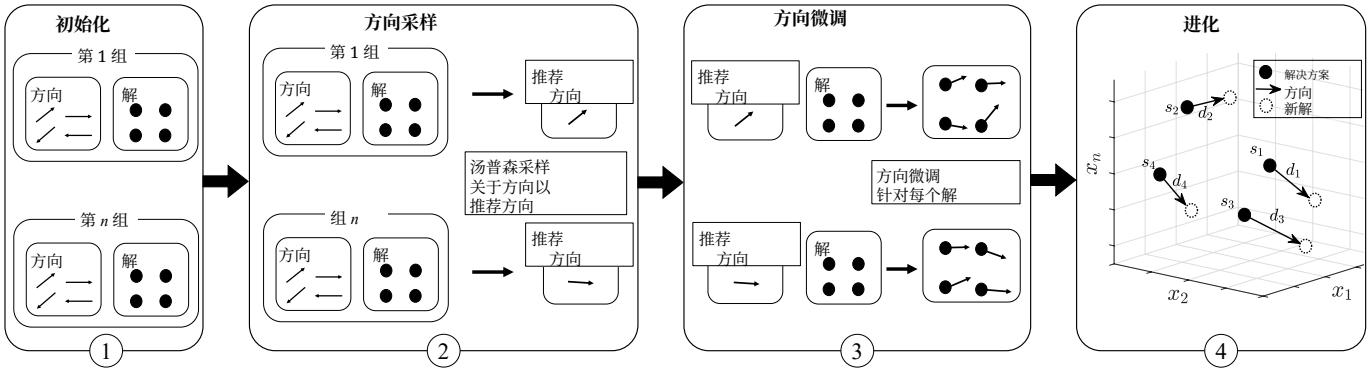


图 1. 所提出的超大规模多目标优化框架（VMOF）示意图。1. 初始化一组方向和解，并将它们随机划分为若干组。2. 为每组采样并推荐一个方向。3. 根据推荐方向，为所有解微调方向。4. 根据微调方向演化每个解。

在超大规模多目标优化问题中遇到的维度问题。

首先，进化方向采样技术从汤普森采样中汲取灵感。汤普森采样是推荐系统 (RS) [15] 中常用的一种方法，用于解决冷启动问题 [16]。汤普森采样能够根据有限的历史交互记录，从大量候选项中高效推荐最优选项，这与解决超大规模多目标优化问题的需求相吻合。因此，我们提出了一种方法，在超大规模搜索空间中建立进化方向分布，随后从该分布中采样，以推荐最合适的进化方向。

然而，仅有合适且通用的演化方向，还不足以在超大规模多目标优化问题的广阔决策空间中找到帕累托最优集（PS）。为确保准确、高效地搜索帕累托最优解，我们提出一种方向微调算法，该算法基于采样方向为每个解计算精细化方向，从而更好地逼近帕累托最优集。图 1 展示了所提出框架的示意图。

1）在本研究中，我们提出一种创新框架，专门用于应对超大规模多目标优化问题中固有的复杂性；这类问题在实际应用中十分常见。全面的实验结果证实，所提出的算法能够解决规模高达一百万的超大规模多目标优化问题。

2) 为解决超大规模多目标优化问题，我们提出一种采用汤普森采样对方向进行采样和推荐的新方法。该方法使我们提出的框架 VMOF 能够在函数评估次数有限的情况下，于超大规模多目标优化问题典型的广阔搜索空间中，推荐具有普遍性且适用的演化方向。

3) 此外，为了探索超大规模多目标优化问题所固有的广阔决策空间，我们提出一种方向微调方法。该方法利用采样方向引导每个解趋近帕累托最优解的近似解。

本文其余部分安排如下。在第三区中，我们定义超大规模多目标优化问题，并介绍相关研究。第三区介绍所提出方法的详细内容。第四区对所设计的框架进行分析。在第五区中，我们将所提出的方法与若干先进的大规模多目标进化算法进行一系列实验比较。最后，在第六区总结全文并讨论未来工作。

II. Preliminary Studies and Related Work

A. 超大规模多目标优化

大规模多目标优化问题的数学形式如下:

$$\begin{aligned} &\textbf{min } \boldsymbol{F}(\boldsymbol{x}) = < f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}) > \\ &\text{s.t. } \boldsymbol{x} \in \Omega \end{aligned} \tag{1}$$

其中，$\boldsymbol{x} = (x_1, x_2, \ldots\ldots, x_n)$ 是 n 维决策向量，$\boldsymbol{F} = (f_1, f_2, \ldots\ldots, f_m)$ 是 m 维目标向量。

假设 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 是某个多目标优化问题的两个解，当且仅当 $f_i(\boldsymbol{x}_1) \leqslant f_i(\boldsymbol{x}_2)(\forall i = 1, \ldots\ldots, m)$ 且至少存在一个目标 $f_j(j \in \{1, 2, \ldots, m\})$ 满足 $f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2)$ 时，称解 $\boldsymbol{x}_1$ 帕累托支配解 $\boldsymbol{x}_2$（记作 $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$）。决策空间中所有帕累托最优解的集合称为帕累托最优集（PS），PS 在目标空间中的投影称为帕累托最优前沿（PF）。

需要指出的是，传统的大规模优化问题通常将决策变量数量n 定义为超过 100 [12]。然而，在许多现实世界问题中，例如神经架构搜索、关键节点检测和车辆路径问题，决策变量很容易超过 100，000。因此，我们引入了超大规模多目标优化问题（VLSMOPs）的概念，其中 n 超过 100，000。

B. 相关工作

大规模多目标进化算法分为三类：基于决策变量分组的方法、基于决策空间约简的方法，以及基于新型搜索策略的方法 [3]。

1) 基于决策变量分组：第一类方法基于决策变量分组。Antonio等人 [10],提出的 CCGDE3 是首个用于求解大规模多目标优化问题的多目标进化算法。该算法维护多个不同的子组，并以分而治之的方式优化大规模多目标优化问题。此后，研究者引入了随机分组 [17] 和差分分组 [18] 技术来求解大规模多目标优化问题。

基于决策变量分组的方法中，另一类是基于变量分析的算法。Ma等于 [20] 提出了MOEA/DVA，通过分析决策变量的控制特性来划分决策变量。为更通用地划分决策变量，Zhang等 [21] 提出了一种基于决策变量聚类方法的大规模进化算法。

对决策变量进行分组是解决大规模多目标优化问题的有效方法，但大规模决策变量之间的关联可能会被错误识别 [21],，从而导致局部最优。对于超大规模决策变量，这一缺陷可能会进一步降低性能。

2）基于决策空间降维的方法：第二类方法基于降维。Liu等于[23],提出采用主成分分析解决大规模多目标优化问题，即PCA-MOEA。此外，Tian等通过学习帕累托最优子空间 [5]，利用无监督神经网络解决大规模问题。

在这一类中，研究者提出了基于问题转换的算法来求解大规模多目标优化问题。Zille等 [11]提出了一种加权优化框架（WOF），旨在作为一种通用方法，可与任何基于种群的元启发式算法配合使用。此外，He等[24] 提出了一种通用的大规模多目标优化框架，将原始的大规模多目标优化问题重新表述为低维单目标优化问题。

基于决策空间约简的方法在约简后可能会丢失原始搜索空间中的重要信息。在求解超大规模多目标优化问题时，降维会丢失更多搜索空间，因此无法充分探索原始搜索空间 [25]，导致性能不佳。

3) 基于新型搜索策略：第三类研究主要围绕新型搜索策略展开 [26], [27]。Tian 等。 提出了一种竞争性群体优化器，以高效搜索大规模多目标优化问题的解。与此同时，Hong 等。开发了一种具有改进多样化机制的可扩展算法 [28]。最近，Wang 等。 [29]提出了一种基于生成对抗网络的流形插值框架，用于学习流形并生成高质量解。Hong 等。 提出了一种利用趋势预测模型和生成-筛选策略来解决大规模多目标优化问题的模型。在另一种方法中，Qin等。 [30]提出了一种使用直接采样的大规模多目标进化算法。与此相反，我们采用汤普森采样来选择最优方向，并从理论和实证两方面证明了我们的方法对于具有超大规模决策变量的问题更加有效。

多年来，人们提出了众多算法来求解大规模多目标优化问题。更大规模的多目标优化问题也引起了一些关注。例如，Tian 等人。 [31] 提出了一种用于求解超大规模稀疏多目标优化问题。然而，所设计的算法要求问题的决策变量具有稀疏性，即问题的决策变量中存在大量的 0，这限制了该算法。Li 等。 [32],的工作探索了更高维度的多目标优化问题，主要致力于提高求解大规模多目标优化问题的计算速度。然而，对于其所提出方法的全面分析和实验仍然十分缺乏。值得注意的是，超大规模多目标优化问题已在这一研究领域引起关注。然而，该主题仍相对缺乏研究。

C. 汤普森采样

汤普森采样（TS） [33] 是推荐系统中一种常用的策略。汤普森采样最初由 1933 [34], [35]提出，用于分配临床试验中双臂老虎机问题的实验资源。汤普森采样兼顾了统计效率与算法复杂度 [36], ，广泛应用于互联网广告 [37], 、推荐系统 [38] 和超参数调优 [39]。在进化计算领域，Sun 等人[40] 提出了一种基于动态汤普森采样、用于多目标进化算法的新型自适应算子选择机制。

为解决超大规模多目标优化问题，关键在于在极大的搜索空间中平衡探索与利用。上置信界（UCB）是一种实现这种平衡的经典方法，兼顾统计效率与计算效率。然而，这会导致 UCB 算法牺牲统计效率 [36]。相比之下，汤普森采样已被证明能有效解决具有高维特征的老虎机问题 [41]。因此，我们的框架采用汤普森采样，使算法无需在高维决策空间中直接搜索，而是获取能够有效引导种群进化的方向。

III. 基于采样和微调的超大规模多目标优化框架

A. 概述

所提出的框架包含两个主要组成部分：方向采样和方向微调，如图1 所示。这两个组成部分旨在采样通用且合适的方向，并对其进行微调，以协同求解超大规模多目标优化问题。

首先，我们的方案采用汤普森采样，在广阔的搜索空间内对演化方向进行采样和推荐。汤普森采样为大规模空间中的每个进化方向维护一个概率分布，并从这些分布中选择样本，从而在有限的函数评估次数内提出推荐。

其次，为了缓解在超大规模决策空间中难以准确逼近帕累托最优解的问题，尽可能精细地搜索至关重要。为此，我们提出了一种算法，为每个解微调一个进化方向。该算法根据每个解对应的推荐方向生成一组方向。

B. 进化方向采样

对于维度为 d的解 $\mathcal{S}=\{\boldsymbol{s}_i\}$，由于整个搜索空间规模极大，因此无法在其中找到 $\mathcal{D}=\{\boldsymbol{d}_j\}$ 朝向帕累托最优集的方向。因此，我们将方向优化转化为一个冷启动问题，并采用汤普森采样来推荐最佳方向。

对于一个解，其进化方向定义为 $\boldsymbol{d}$ ($\boldsymbol{d}$ =$(d_1,d_2,\ldots,d_d)$). 算法将为解推荐方向，然后为每个解微调方向。

冷启动问题考虑在大规模项目中分配一组固定的有限资源 $\mathcal{I}=\{i_1,\ldots,i_k\}$，以最大化用户 $\mathcal{U}$ =$\{u_1,\ldots,u_l\}$ 的预期回报。在所提出的框架中，每个解 $\boldsymbol{s}_i$ 被视为一个用户 u_i，而一个项目 i_j 则是用于引导解进化的方向 $\boldsymbol{d}_j$。用户 u_i对项目 i_j的响应，或称为奖励 r_{ij}，是解 $\boldsymbol{s}_i$ 沿着方向 $\boldsymbol{d}_j$进化后的质量。

汤普森采样是解决冷启动问题的一种高效算法。在汤普森采样中，每个项目都被赋予一个独立的先验奖励分布 $\theta_j, j\in\{1,\ldots,k\}$。具体而言，使用参数为 $\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_k)$ 和 $\boldsymbol{\beta}=(\beta_1,\ldots,\beta_k)$ 的 Beta 分布来表示该先验分布：

$$P^{Beta}(\theta_j)=\frac{\Gamma(\alpha_j+\beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)}\theta_j^{\alpha_j-1}(1-\theta_j)^{\beta_j-1} \tag{2}$$

其中，α_j 和 β_j 是与第 j 项 i_j 相关的参数，$\Gamma(\cdot)$ 是伽马函数。对于用户 u_i 的给定项目 i_j，如果其对用户 u_i 的奖励为 $r_{ij}=1$，则 α_j 和 β_j 更新为：

$$\alpha_j=\alpha_j+1, \beta_j=\beta_j \tag{3}$$

否则，如果 $r_{ij}=0$，那么我们有：

$$\alpha_j=\alpha_j, \beta_j=\beta_j+1. \tag{4}$$

算法 1： 参数更新

输入： $\boldsymbol{\alpha}$(参数集 α), $\boldsymbol{\beta}$(参数集 β), $\mathcal{F}$ (非支配解集).
输出： $\boldsymbol{\alpha}$(更新后的参数集合 α), $\boldsymbol{\beta}$(集合更新的参数 β）。

对于 $\boldsymbol{f}_i$ 在 $\mathcal{F}$ **执行**
  如果 $f_i=1$ **那么** $r_i\leftarrow 1$ **否则** $r_i\leftarrow 0$;
  $\alpha_i\leftarrow\alpha_i+r_i$;
  $\beta_i\leftarrow\beta_i+1-r_i$;
结束

在我们的方法中，我们采用汤普森采样算法对方向进行采样并推荐最优方向。项 i_j表示方向 $\boldsymbol{d}_j$。随后，根据沿方向 $\boldsymbol{d}_j$应用进化算法后所得解 $\boldsymbol{s}_i$ 的质量，确定方向 d_j 的奖励 r_{ij}。利用解的支配关系评估解的质量：

$$r_{ij}=\begin{cases}1, & s_i \text{ evolved by } \boldsymbol{d}_j \text{ Pareto-dominates } \boldsymbol{s}_i\\ 0, & \text{Otherwise.}\end{cases} \tag{5}$$

算法 2： 进化方向采样

输入： $\mathcal{D}=\{\boldsymbol{d}\}$ (一组方向), $\mathcal{S}=\{s_i\}$(一组个解），n_d （推荐数量方向), e (函数评估用于汤普森采样), $\mathcal{E}$ (一种进化优化器)。
输出： $\mathcal{D}=\{\boldsymbol{d}\}$ (一组采样方向), $\{\mathcal{S}_i\}$ （分组解）。

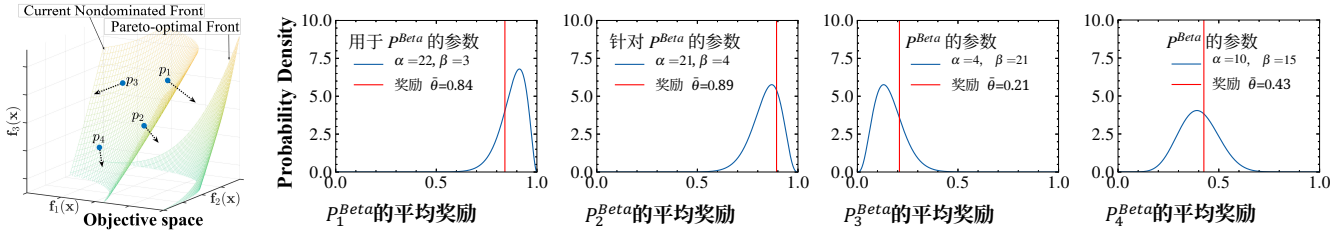
1 $\hat{\mathcal{D}}\leftarrow\emptyset$, $\hat{\mathcal{S}}\leftarrow\emptyset$, $e'\leftarrow 0$;
2 $\{\mathcal{D}_i\}\leftarrow$ 将集合 $\mathcal{D}$ 随机分成 n_d 组;
3 $\{\mathcal{S}_i\}\leftarrow$ 将集合 $\mathcal{S}$ 随机划分为 n_d 组;
4 **对于** $\mathcal{D}_i$ 在 $\{\mathcal{D}_i\}$ 和 $\mathcal{S}_i$ 在 $\{\mathcal{S}_i\}$ **执行**
5 $n\leftarrow$ $\mathcal{D}_i$ 的大小;
6 $\boldsymbol{\alpha}\leftarrow\mathbf{1}\in\mathbb{R}^{n\times 1}$; $\boldsymbol{\beta}\leftarrow\mathbf{1}\in\mathbb{R}^{n\times 1}$;
7 **当** $e'\le e$ **时执行**
8 **对于** $d_j\in\mathcal{D}_i$ 和 $\boldsymbol{s}_j\in\mathcal{S}_d$ **执行** $\boldsymbol{s}_j\leftarrow\boldsymbol{s}_j+\boldsymbol{d}_j$;
9 $\mathcal{F}\leftarrow$ **快速非支配排序**($\mathcal{S}_i$);
10 $e'+|\mathcal{S}_i|$; // 记录函数评估的数量在集合 $\mathcal{S}_i$ 上使用的函数评估。
11 **参数更新**($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathcal{F}$);
12 $S_i\leftarrow\mathcal{E}(S_i,\ \mathcal{F})$ // 使用进化算法在 S_i 上进行 $\mathcal{F}$ 的评估，以获得新的 S_i;
13 **end**
14 **对于** $i\leftarrow 1$ 到 n **执行**
15 $\bar{\ }$从 Beta 分布中抽取 θ_i 估计的平均奖励 Beta 分布 $P^{Beta}(\alpha_i,\ \beta_i)$;
16 **end**
17 $k\leftarrow\arg\max^{i\in\{1,\ldots,n\}}\bar{\theta}_i$;
18 $\hat{\ }_{\mathcal{D}\leftarrow}\hat{\mathcal{D}}\cup\{\boldsymbol{d}_k\}$;
19 **结束**

由于在方向进化过程中采用了进化算法，式（2）中的固定奖励分布无法适应可能解的非平稳特性。为解决这一问题，我们采用动态汤普森采样 [42] 策略，具体见算法 1。每个分布的 α 和 β 均初始化为 1，且 $P^{Beta}(\theta_i)$ 初始化为 $[0,1]$上的均匀分布。

汤普森采样算法推荐方向 $\boldsymbol{d}$。使用方向 $\boldsymbol{d}_j$ 为解$\boldsymbol{s}_i$ 所获得的奖励 r_{ij}，通过评估使用 $\boldsymbol{d}_j$演化得到的新解$\hat{\ }\boldsymbol{s}_i$ 来计算。对于每个解组 $\mathcal{S}_i$，我们将从方向组 $\mathcal{D}_i$中为该组推荐最佳方向 $\boldsymbol{d}_k$。图2给出了在方向方法上进行汤普森采样的示例，采样方向的算法如算法2所述。第2行和第3行将方向和解划分为 n_d 组。**快速-非支配排序** 是源自NSGA-II的过程[1]。第11行记录已消耗的函数评估。

C. 进化方向微调

要在超大规模决策空间中搜索一组帕累托最优解，需要解决的一个关键问题是进化方向是否充足。基于汤普森采样的方向采样为我们提供了少量潜在的进化方向。为了解决



(a) (b)

图2。该插图描述了基于汤普森采样的进化方向采样过程。最初，每个个体在决策空间内都拥有一个独特的方向（$\boldsymbol{d}_i$），引导个体完成一次进化。随后，每个个体（$\boldsymbol{p}_i$）在目标空间内进行进化和评估。根据评估结果，算法更新每个方向对应的 α 和 β 参数。值得注意的是，α 和 β 分别表示个体沿方向 $\boldsymbol{d}_i$ 进化和退化的次数。然后，算法计算每个方向的 $P^{Beta}(\theta)$ 分布（在 P_i^{Beta} 中以蓝线表示），并从 $P^{Beta}(\theta)$ 分布中采样平均奖励 $\bar{\theta}_i$（在 P_i^{Beta} 中以红线表示）。选择对应于平均奖励 $\bar{\theta}_i$ 最高的分布的方向。在此示例中，根据 P_1^{Beta}，采样的方向为 $\boldsymbol{d}_1$。

进化方向不足的问题，本节提出了一种针对每个解进行方向微调的方法。

进化方向微调的本质是对推荐得到的潜在方向进行微调，从而为每个解生成一个独特且合适的进化方向。

对于成对解组 $\mathcal{S}_i$ 和方向组 $\mathcal{D}_d$，我们已采样并推荐了方向 $\boldsymbol{d}_k$。为了对每个解进行微调，我们基于推荐方向 $\boldsymbol{d}_k$ 生成一个方向种群，并从解组 $\mathcal{S}_i$ 中选出一组代表性解来评估相应的种群。

算法从分组后的解 $\mathcal{S}_i$ 中，选取第一非支配前沿上拥挤距离 [1]最大的代表性解，其中包括边界解。对于每个代表性解 $\boldsymbol{s}_d$，算法首先基于推荐方向 $\boldsymbol{d}_k$ 初始化一组随机方向种群 $\mathcal{D}_p = \{\boldsymbol{d}_i + \boldsymbol{d}_k\}$，其规模为 $n_{di} = N/n_d$。其次，使用方向种群对解 $\boldsymbol{s}_d$ 进行进化并评估。随后，我们对方向种群应用进化算法 $\mathcal{E}$ 并重复上述过程。

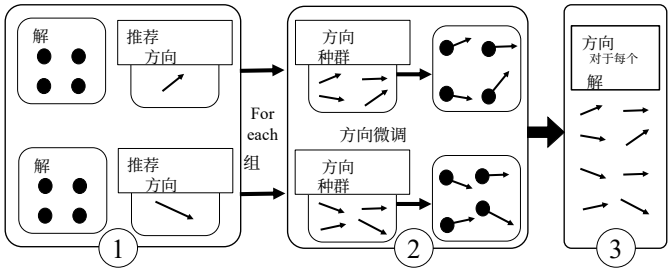


图 3. 进化方向微调示意图。1. 对于成对解组 $\mathcal{S}_i$ 和方向组 $\mathcal{D}_d$。2. 基于推荐方向初始化一组方向种群，然后使用方向种群对每个解进行进化并评估；4. 为每个解获得一组方向。

进化方向的微调在每个分组解 $\mathcal{S}_i$ 和推荐方向 $\boldsymbol{d}_k$ 上进行。最后，根据推荐方向为每个解获得一组方向。微调的可视化方向如图 3 所示，流程见算法3。

算法 3： 进化方向微调

输入： $\hat{\mathcal{D}} = \{\boldsymbol{d}_k\}$ (推荐方向), $\{\mathcal{S}_i\}(\mathcal{S}_i = \{\boldsymbol{s}_i\})$(分组解), e (用于微调的函数评估), $\mathcal{E}$ (一个进化优化器)。

输出： $\mathcal{D} = \{\boldsymbol{d}_i\}$ (微调方向), $\mathcal{S}$ (微调期间产生的解)

$N \leftarrow \sum_{i=0}^{i<|\{\mathcal{S}_i\}|} |\mathcal{S}_i|$; // 整个种群的大小。
$n_d \leftarrow |\hat{\mathcal{D}}|$; // 推荐方向的大小。
$n_{di} \leftarrow N/n_d$; // 方向种群的大小。
$\mathcal{D} \leftarrow \emptyset$, $e' \leftarrow 0$;
对于 $\boldsymbol{d}_k \in \hat{\mathcal{D}}$ 和 $\mathcal{S}_i$ 在 $\{\mathcal{S}_i\}$ **执行**
  $\boldsymbol{s}_d \leftarrow$ **选择多样性解** $(\mathcal{S}_i, n_{di})$;
  $\mathcal{D}_p \leftarrow$ 使用大小 n_{di} 初始化方向种群基于推荐方向 $\boldsymbol{d}_k$;
  while $e' \leq e$ **do**
    对于 $\boldsymbol{d}_i \in \mathcal{D}_p$ **执行**
      $\boldsymbol{s}_i \leftarrow \boldsymbol{s}_d + \boldsymbol{d}_i$; $\mathcal{S}_p \leftarrow \mathcal{S}_p \cup \{\boldsymbol{s}_i\}$;
    end
    $\mathcal{F} \leftarrow$ **快速非支配排序**$(\mathcal{S}_p)$;
    $e' + |\mathcal{S}_p|$; // 记录函数的数量在集合 $\mathcal{S}$ 上使用的评估次数 p.
    $D_p \leftarrow \mathcal{E}(D_p, \mathcal{F})$ // 使用进化算法 *on* $\mathcal{D}_p$ 通过对 $\mathcal{F}$ 进行评估，获得新的 $\mathcal{D}$ p ;
  end
  对于 $i \leftarrow 1$ 至 n_{di}，**执行**
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{d}_i\}$; $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_p$;
  end
结束

D. 一种超大规模多目标优化框架

算法的整体框架如算法 4 所示，以及种群和方向的初始化

以随机方式进行。所提出的VMOF的关键在于通过汤普森采样执行全局搜索，并对方向进行微调，以实现更细粒度的搜索。在第 6行所描述的优化步骤中，采用粒子群算法，这是因为该算法与采样方向和微调方向具有内在的兼容性。该算法使用方向集初始化粒子群，记为 $\mathcal{D}$。在粒子群优化之后，记录与每个解对应的方向，并将其用作下一轮进化方向采样的初始方向。

算法 4： VMOF

输入： N (种群规模), n_d (推荐方向大小), e (函数评估用于采样、微调和演化， $\mathcal{E}$ (An 进化优化器）。)

输出： S （由 VMOF 优化得到的一组解）。

1 $\mathcal{S} \leftarrow$随机初始化 N 解;
2 $\mathcal{D} \leftarrow$ 随机初始化 N 方向;
3 **当**终止条件未满足**时**
4 $(\hat{\mathcal{D}}, \{\mathcal{S}_i\}) \leftarrow$ **进化方向采样** $(\mathcal{D}, \mathcal{S}, n_d, e, \mathcal{E})$;
5 $(\mathcal{D}, \mathcal{S}) \leftarrow$ **进化方向微调** $(\hat{\mathcal{D}}, \{\mathcal{S}_i\}, e, \mathcal{E})$;
6 $(\mathcal{D}, \mathcal{S}) \leftarrow$ **优化器** $(\mathcal{D}, \mathcal{S}, e)$;
7 **结束**

IV. 分析研究

本节讨论支持方向采样和方向微调设计的一些潜在理论基础，并分析所提出框架的时间复杂度。

A. 超大规模搜索空间中的方向采样

首先，方向 i 的奖励通过旧解与沿该方向演化得到的新解之间的帕累托支配关系来刻画。因此，奖励，即支配关系，很容易表示为二元形式。 consequently，Beta 分布天然适用于伯努利奖励，因此被采用来在方向上建立分布。

其次，已证明，在伯努利奖励的情况下，汤普森采样能够达到累积遗憾的渐近速率 [43], 这为使用汤普森采样推荐最佳解提供了基础。这里，我们使用贝塔先验给出一个与问题无关的汤普森采样遗憾界：

$$\mathbb{E}[r(T)] \leq \mathcal{O}(\sqrt{ST \ln T}) \tag{6}$$

其中， T 是时间， S 是可用候选项的数量。给定种群规模 N、建议方向的数量 n_d以及用于汤普森采样的函数评估次数 e， T 是更新汤普森采样的迭代次数（ $T = e/|S_i| = e/(N/n_d)$，算法 2 第 11 行），且S 为 n_d，因此方向采样的遗憾界为$\mathbb{E}[r] \leq \mathcal{O}(\sqrt{(en_d^2)/N \ln (en_d/N)})$。

B. 超大规模搜索空间中的方向微调

当最优解位于大规模搜索空间中时，就会出现维度灾难，这体现了在乘积空间 [14] 上通过穷举进行优化所面临的挑战。对于多目标优化问题，这可以粗略地描述为：在优化 m 个目标、包含 d 个决策变量 [14] 的情况下，需要进行 $n(1/\epsilon)^d$ 次函数评估，才能获得 n 个解；每个解与真实帕累托最优集（PS）的误差均不超过 ϵ。

术语 n 表示超大规模多目标优化问题需要一组解，这是由其多目标特性决定的。术语 $(1/\epsilon)^d$ 与问题决策变量的维度直接相关。表达式$n(1/\epsilon)^d$ 强调了在广阔的搜索空间中寻找一组帕累托最优集的难度，因此凸显了对每个解的探索方向进行微调、以便充分探查搜索空间的必要性。

C. 时间复杂度分析

给定一个决策变量维度为 d、目标数为m 的多目标优化问题，对于所提出的 VMOF，种群规模为 N，采样方向的数量为 n_d，函数评估总次数为 E，方向采样、方向微调和优化算法的函数评估次数为 e。算法 2 本质上是对 N 个方向进行 en_d/N 次 Beta 分布更新。每次更新都会沿该方向演化一个 d维解，然后对其进行评估和排序，因此算法 2 的时间复杂度为 $\mathcal{O}(Nd(en_d/N)MN^2) = \mathcal{O}(den_dMN^2)$。算法 3 对 N d维方向进行微调，每次微调重复 $e/|S_p| = en_d/N$ 次，因此算法 3 的时间复杂度为 $\mathcal{O}(den_dMN^2)$；所采用的粒子群优化的时间复杂度为 $\mathcal{O}(MN^2de)$，因此 VMOF 的总时间复杂度为 $E/e[\mathcal{O}(den_dMN^2) + \mathcal{O}(den_dMN^2) + \mathcal{O}(MN^2de)] = \mathcal{O}(dMN^2E)$。

V. 实验研究

A. 比较和测试问题中使用的算法

将所提出的 VMOF与若干最先进的多目标进化算法进行比较，包括 NSGA-II [1], CCGDE3 [10],WOF [11], LSMOF [24], LMOCSO [12], GMOEA [44] 和DGEA [27]。所有对比算法均在 PlatEMO [45]上实现。作为基线, NSGA-II 是一种用于求解多目标优化问题的多目标进化算法。CCGDE3 是一种通过决策变量分组求解大规模多目标优化问题的典型方法。WOF 和 LSMOF 是两种基于决策空间缩减的代表性框架。LMOCSO和 LMOEA-DS 分别代表两种很有前景的方法，它们采用了新颖的搜索策略，分别利用竞争群体优化和直接采样。尽管DGEA同样利用了采样和方向向量等概念，但其与我们的方法存在显著差异。DGEA选择一个均衡的父代种群，并利用这些选择结果在决策空间中构建方向向量，从而促进生成具有潜力的子代解。

基准测试大规模多目标优化问题 [46] 广泛用于比较大规模多目标进化算法的性能。我们还纳入了

TREE [13] 作为一个真实世界问题纳入实验。该问题的维度高达 900,000，属于超大规模多目标优化问题。在实验中，采用逆世代距离 [47]和超体积 [48]指标评估对比算法的收敛性和多样性。

为检验指标值差异的统计显著性，每种算法均独立运行20次，并采用 Wilcoxon 秩和检验 [49] 比较统计结果，显著性水平设为0.05。符号“+”、“-”和“=”分别表示该算法的结果显著优于、显著劣于或与所提出的 VMOF 算法在统计上无显著差异。

B. 参数设置

1) 决策变量：超大规模多目标优化问题的决策变量数量为100,000、500,000和1,000,000，大规模多目标优化问题的决策变量数量为100至20,000。

2) 种群规模：按照 WOF [11],LSMOF [24], 和 DGEA [27], 中双目标和三目标问题的种群规模设置，对于包含两个目标的测试基准，种群规模 N 设为100；对于包含三个目标的测试基准，种群规模设为105。

3) 终止条件：最大评估次数记为 E，设定为 100,000。尽管这一函数评估次数可能看起来较少，但对于实际应用而言是可行的 [24]。这主要是因为经济和/或计算成本不可避免地限制了函数评估次数，在大规模优化问题中尤为如此。

对于需要大量时间来求解 VLSMOP或大规模多目标优化问题的算法，函数评估将无法作为公平的终止条件。我们在表 I 中记录了解决1,000 维大规模多目标优化问题 LSMOP1 所耗费的时间，作为算法选择的初步实验结果。耗时超过 1,000 秒（比典型方法长五倍）的算法以粗体突出显示。为便于比较，我们选择 MOEA/DVA 作为代表性示例。

表 I 用于求解 1000 维双目标大规模多目标优化问题 1 的大规模多目标进化算法运行时间初步实验（100,000 次适应度评估，单位：秒）

CCGDE3	MOEA/DVA	LMEA	WOF	LSMOF
16.61	**1709**	**1582**	34.26	27.58
MOEA/PSL	LMOCSO	LMOEA-DS	DGEA	VMOF
4993	28.15	14.51	28.27	25.52

*4)*算法的参数设置： 根据 [45]，对比算法的参数均按照其原始论文进行设置。每种算法的具体参数见补充材料[1]中的表 S-1。

对于所提出的算法，采样方向的数量 n_d设为 $N/4 = 25$。方向采样的函数评估次数 e 方向采样、方向微调和优化算法的参数均设为 $0.05E$。算法 2 和 3 中使用的进化优化器 $\mathcal{E}$ 为 NSGA-II [1]。

[1]可通过以下链接下载补充材料：补充材料链接

C. 超大规模多目标优化问题上的性能

为了测试我们所提出算法求解超大规模多目标优化问题的性能，我们将决策变量设置为1,000,000，并在表II和表III中给出 IGD 值和超体积值。随后，我们在表IV和表V中分别给出包含100,000和500,000 个决策变量时的 IGD和超体积结果。在每个表的最后一列中，我们给出了我们所提出的算法与其他算法中的最佳结果之间的变化率（ROC）指标。

平均而言，如表 II 和表IV 中的 IGD 值所示，所提出的方法相较于次优算法分别提升了 12% 和 19%。这一改进在大规模多目标优化问题8和大规模多目标优化问题 9 上尤为明显。鉴于这两个问题的独特特征，我们推测，随着决策变量数量增加到 100,000或更多，Pareto 前沿的复杂度也会相应增加，而具有混合适应度景观的VLSMOP 尤其复杂。尽管面临这些挑战，我们的算法仍能在如此大规模的搜索空间中，通过合理数量的函数评估次数有效识别最优方向。这一能力对于求解此类问题具有重要的实际意义。

从表 II 可以观察到，我们的算法在求解 1,000,000 维超大规模多目标优化问题时取得了显著提升。在 LSMOP2 和 LSMOP4 上，我们算法的 IGD 值略低，但与最优结果相比并无显著差异。

NSGA-II 是一种经典算法，但在处理超大规模搜索空间时效果较差。WOF 在三目标 LSMOP7 上的表现仅略优于 VMOF。LMOEA-DS 和 DGEA 同样采用了采样机制，能够处理大规模多目标优化问题，但与我们提出的算法相比，其整体表现仍有差距。LMOCSO 确实是一种用于解决大规模多目标优化问题的有效粒子群算法，因为它通过竞争粒子搜索方向。然而，在更高维度下，若不采用方向采样和微调，这些方向往往效率不高。因此，LMOCSO 的表现仍不够理想。

我们进一步使用 VLSMOP* 基准测试 [50], 评估了所提出算法的性能。该基准测试是原始大规模多目标优化问题基准测试的一个变体，其中引入了平移变换。补充材料 1 中表 S-II 和 S-III 所列的实证结果表明，我们提出的 VMOF 始终优于现有的多目标进化算法。

D. 在大规模多目标优化问题上的性能

本节中，我们针对双目标和三目标大规模多目标优化问题进行了实验，决策变量数量从 100 到 20,000 不等。所得的逆世代距离和超体积值由

表 II 维度为 1,000,000 的双目标 VLSMOP 问题上各对比算法获得的 IGD 指标。每行中的最佳结果以粗体标出。

问题	D	NSGA-II	CCGDE3	MOEA/DVA	WOF	LSMOF	LMOCSO	LMOEA-DS	DGEA	VMOF	
LSMOP1	1,000,000	1.17e+01-	6.39e+00-	1.18e+01-	3.11e-01-	3.48e-01-	1.80e+00-	3.33e-01-	3.34e-01-	**1.59e-01**	**48.83%**
LSMOP2	1,000,000	4.97e-03=	5.00e-03=	7.13e-03=	3.72e-03=	4.54e-03=	3.82e-03=	4.11e-03=	**3.59e-03=**	4.30E-03	-19.91%
LSMOP3	1,000,000	3.96e+01-	3.42e+01-	4.14e+02-	6.36e+02-	1.58e+00=	2.20e+03-	1.58e+00=	1.36e+01-	**1.57e+00**	**0.31%**
LSMOP4	1,000,000	7.41e-03=	7.04e-03=	9.97e-03=	3.73e-03=	4.55e-03=	5.32e-03=	3.82e-03=	**3.61e-03=**	4.41E-03	-22.23%
LSMOP5	1,000,000	2.50e+01-	1.40e+01-	2.49e+01-	7.42e-01-	7.42e-01-	3.70e+00-	7.42e-01-	7.46e-01-	**6.84e-01**	**7.85%**
LSMOP6	1,000,000	7.42e-01-	7.42e-01-	2.49e+03-	3.69e-01-	3.06e-01-	1.12e+03-	3.06e-01-	7.42e-01-	**1.48e-01**	**51.56%**
LSMOP7	1,000,000	9.21e+04-	3.74e+04-	9.57e+04-	1.52e+00=	1.52e+00=	2.29e+03-	1.52e+00=	1.52e+00=	**1.52e+00**	**0.01%**
LSMOP8	1,000,000	2.08e+01-	1.19e+01-	2.14e+01-	2.93e-01-	7.42e-01-	3.21e+00-	7.42e-01-	7.02e-01-	**1.55e-01**	**46.99%**
LSMOP9	1,000,000	6.07e+01-	3.05e+01-	6.40e+01-	6.40e+01-	8.02e-01-	1.68e+01-	5.29e-01=	2.19e+01-	**5.24e-01**	**0.95%**
(+/-/=)		0/7/2	0/7/2	0/7/2	0/6/3	0/5/4	0/7/2	0/4/5	0/6/3		7

表 III 在维度为 1,000,000 的双目标 VLSMOP 问题上，对比算法获得的 HV指标。每行中的最佳结果以粗体标出。

问题	D	NSGA-II	CCGDE3	MOEA/DVA	WOF	LSMOF	LMOCSO	LMOEA-DS	DGEA	VMOF	
LSMOP1	1,000,000	0.00e+00-	0.00e+00-	0.00e+00-	2.49e-01-	2.29e-01-	0.00e+00-	2.27e-01-	2.34e-01-	**4.04e-01**	**61.94%**
LSMOP2	1,000,000	5.80e-01=	5.80e-01=	5.75e-01=	5.82e-01=	5.81e-01=	5.82e-01=	5.81e-01=	**5.82e-01=**	5.82E-01	-0.13%
LSMOP3	1,000,000	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00	0.00%
LSMOP4	1,000,000	5.76e-01=	5.76e-01=	5.71e-01=	5.82e-01=	5.81e-01=	5.79e-01=	5.81e-01=	**5.82e-01=**	5.81E-01	-0.17%
LSMOP5	1,000,000	0.00e+00-	0.00e+00-	0.00e+00-	**9.09e-02=**	**9.09e-02=**	0.00e+00-	9.09e-02=	8.13e-02=	**9.09e-02**	0.00%
LSMOP6	1,000,000	9.08e-02-	9.08e-02-	0.00e+00-	9.62e-02-	1.08e-01-	0.00e+00-	1.08e-01-	9.09e-02-	**1.68e-01**	**55.96%**
LSMOP7	1,000,000	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00=	0.00e+00	0.00%
LSMOP8	1,000,000	0.00e+00-	0.00e+00-	0.00e+00-	1.19e-01-	9.09e-02-	0.00e+00-	9.09e-02-	4.91e-03-	**2.11e-01**	**76.97%**
LSMOP9	1,000,000	0.00e+00-	0.00e+00-	0.00e+00-	0.00e+00-	9.19e-02-	0.00e+00-	**1.53e-01=**	0.00e+00-	1.51E-01	-0.95%
(+/-/=)		0/5/4	0/5/4	0/5/4	0/4/5	0/4/5	0/5/4	0/3/6	0/4/5		6

表 IV 维度为100,000和 500,000的双目标 VLSMOP 问题上对比算法获得的 IGD 指标。每行中的最佳结果以粗体突出显示。

问题	D	NSGA-II	CCGDE3	MOEA/DVA	WOF	LSMOF	LMOCSO	LMOEA-DS	DGEA	VMOF	ROC（%）
LSMOP1	100,000 500,000	1.28e+01- 1.55e+01-	1.07e+01- 1.36e+01-	1.17e+01- 1.17e+01-	4.37e-01- 4.83e-01-	7.49e-01- 1.10e+00-	1.67e+00- 2.43e+00-	3.20e-01- 3.41e-01-	4.33e-01- 9.20e-01-	**2.00e-01** **1.82e-01**	**53.81%** **62.32%**
LSMOP2	100,000 500,000	6.22e-02= 8.89e-02=	4.98e-02= 7.60e-02=	7.24e-03= 7.09e-03=	**3.82e-02=** 5.83e-02=	4.99e-02= 7.51e-02=	3.88e-02= 5.88e-02=	4.26e-03= 6.75e-03=	3.85e-02= 5.80e-02=	5.02E-02 **5.05e-02**	-31.41% **12.93%**
LSMOP3	100,000 500,000	3.88e+01- 3.31e+01-	1.84e+01- 2.90e+01-	3.21e+02- 1.94e+03-	**8.61e-01=** 1.29e+00-	**8.61e-01=** 1.29e+00-	3.90e+01- 2.55e+01-	1.58e+00= 2.76e+03-	8.62e-01= 1.29e+00-	**8.61e-01** **8.61e-01**	0.00% **33.26%**
LSMOP4	100,000 500,000	5.62e-02= 8.33e-02=	5.99e-02= 7.85e-02=	9.63e-03= 1.07e-02=	5.82e-02= 6.17e-02=	5.24e-02= 7.85e-02=	5.83e-02= 6.20e-02=	4.22e-03= 1.06e-02=	5.82e-02= 6.04e-02=	**3.30e-02** **5.72e-02**	**37.02%** **5.30%**
LSMOP5	100,000 500,000	2.13e+01- 3.13e+01-	1.71e+01- 2.52e+01-	2.50e+01- 2.49e+01-	5.89e-01= 6.71e-01-	9.46e-01- 1.42e+00-	3.44e+00- 5.34e+00-	7.42e-01- 2.49e+01-	5.57e-01- 2.73e+00-	**5.01e-01** **3.60e-01**	**10.05%** **46.35%**
LSMOP6	100,000 500,000	6.75e+04- 5.75e+04-	4.88e+04- 2.98e+04-	1.59e+03- 5.56e+03-	1.71e+00- 1.34e+00-	7.81e-01- 8.47e-01-	1.39e+03- 4.44e+02-	3.06e-01- 2.73e+03-	5.07e+02- 8.36e-01-	**7.05e-01** **7.05e-01**	**9.73%** **15.67%**
LSMOP7	100,000 500,000	9.47e-01- 9.46e-01-	9.47e-01- 9.46e-01-	9.63e+04- 9.62e+04-	**4.57e-01+** 6.69e-01-	9.13e-01- 9.13e-01-	9.47e-01- 9.46e-01-	1.52e+00= 9.55e+04-	9.13e-01- 1.32e+00-	8.34E-01 **6.33e-01**	-82.49% **5.38%**
LSMOP8	100,000 500,000	9.49e-01- 9.49e-01-	9.49e-01- 9.49e-01-	2.13e+01- 2.14e+01-	1.73e-01- 9.75e-02=	2.32e-01- 2.16e-01-	9.48e-01- 9.49e-01-	7.42e-01- 2.14e+01-	9.46e-01- 1.35e+00-	**8.01e-02** **7.46e-02**	**53.70%** **23.49%**
LSMOP9	100,000 500,000	1.24e+02- 1.44e+02-	9.05e+01- 8.92e+01-	6.38e+01- 6.39e+01-	1.15e+00- 1.27e+00-	1.49e+00- 1.14e+00-	6.88e+01- 6.63e+01-	6.08e-01+ 6.40e+01-	6.17e+01- 6.89e+01-	**5.87e-01** **5.86e-01**	**48.96%** **48.60%**
(+/-/=)		0/14/4	0/14/4	0/14/4	1/10/7	0/13/5	0/14/4	1/11/6	0/13/5		

表V 对比算法在维度为100,000和500,000的双目标 VLSMOP 问题上获得的HV指标。每行中的最佳结果以粗体突出显示。

问题	D	NSGA-II	CCGDE3	MOEA/DVA	WOF	LSMOF	LMOCSO	LMOEA-DS	DGEA	VMOF	ROC (%)
LSMOP1	100,000	0.00e+00-	0.00e+00-	0.00e+00-	2.73e-01-	9.09e-02-	0.00e+00-	2.42e-01-	3.07e-01-	**5.83e-01**	**89.90%**
	500,000	0.00e+00-	0.00e+00-	0.00e+00-	3.97e-01-	9.09e-02-	0.00e+00-	2.16e-01-	1.84e-01-	**6.26e-01**	**57.68%**
LSMOP2	100,000	8.21e-01=	8.28e-01=	5.76e-01=	**8.43e-01=**	8.26e-01=	8.41e-01=	5.81e-01=	8.42e-01=	8.26E-01	-2.02%
	500,000	8.18e-01=	8.20e-01=	5.74e-01=	8.40e-01=	8.30e-01=	8.40e-01=	5.77e-01=	**8.41e-01=**	8.27E-01	-1.66%
LSMOP3	100,000	0.00e+00-	0.00e+00-	0.00e+00-	**9.09e-02=**	**9.09e-02=**	0.00e+00-	0.00e+00-	8.98e-02=	**9.09e-02**	0.00%
	500,000	0.00e+00-	0.00e+00-	0.00e+00-	**9.09e-02=**	**9.09e-02=**	0.00e+00-	0.00e+00-	**9.09e-02=**	**9.09e-02**	0.00%
LSMOP4	100,000	8.20e-01=	8.30e-01=	5.70e-01=	**8.43e-01=**	8.31e-01=	8.43e-01=	5.81e-01=	8.43e-01=	8.26E-01	-2.02%
	500,000	8.11e-01=	8.20e-01=	5.69e-01=	8.35e-01=	8.24e-01=	8.35e-01=	5.71e-01=	**8.37e-01=**	8.17E-01	-2.39%
LSMOP5	100,000	0.00e+00-	0.00e+00-	0.00e+00-	3.36e-01=	9.09e-02-	0.00e+00-	9.09e-02=	2.49e-01-	**3.44e-01**	**2.38%**
	500,000	0.00e+00-	0.00e+00-	0.00e+00-	3.40e-01=	9.09e-02-	0.00e+00-	0.00e+00-	0.00e+00-	**3.44e-01**	**1.18%**
LSMOP6	100,000	0.00e+00=	0.00e+00=	0.00e+00-	0.00e+00=	**1.20e-02=**	0.00e+00=	1.08e-01=	0.00e+00=	0.00E+00	-100.00%
	500,000	0.00e+00-	0.00e+00-	0.00e+00-	0.00e+00-	3.12e-03-	0.00e+00-	0.00e+00-	7.00e-03-	**1.18e-01**	**1585.71%**
LSMOP7	100,000	8.85e-02=	8.86e-02=	0.00e+00-	**1.58e-01+**	8.93e-02=	8.88e-02=	0.00e+00-	9.01e-02=	9.01E-02	-42.97%
	500,000	9.04e-02=	9.04e-02=	0.00e+00-	**9.08e-02=**	9.05e-02=	9.05e-02=	0.00e+00-	8.14e-02=	9.07E-02	-0.11%
LSMOP8	100,000	8.44e-02-	8.50e-02-	0.00e+00-	4.12e-01-	3.83e-01-	8.68e-02-	9.09e-02-	9.04e-02-	**4.95e-01**	**20.15%**
	500,000	8.44e-02-	8.53e-02-	0.00e+00-	4.60e-01=	4.00e-01-	8.58e-02-	0.00e+00-	8.90e-02-	**5.00e-01**	**8.70%**
LSMOP9	100,000	0.00e+00-	0.00e+00-	0.00e+00-	1.47e-01=	9.59e-02-	0.00e+00-	1.30e-01=	0.00e+00-	**1.92e-01**	**30.61%**
	500,000	0.00e+00-	0.00e+00-	0.00e+00-	1.29e-01-	1.48e-01=	0.00e+00-	0.00e+00-	0.00e+00-	**1.92e-01**	**29.73%**
(+/-/=)		0/11/7	0/15/3	0/10/8	1/5/12	0/8/10	0/11/7	0/11/7	0/9/9		

算法在补充材料的表 S-IV 至 S-XI 中有详细说明 [1]。

总体而言，VMOF 优于八种当代最先进的算法。例如，我们提出的 VMOF 在决策变量范围为 100 至 1,000 的 36 个测试实例中，有 19 个取得了最佳结果（参见补充材料中的表 S-IV）。与 LSMOP2、LSMOP5、LSMOP6 和 LSMOP8 相比，VMOF 展现出显著优势。即使维度扩展到 20,000，我们的算法仍显著优于所有对比算法。尽管在较低维度（100 至 1,000）下具有竞争力，我们的算法在更高维度下同样表现出色。在补充材料的表 S-VII 中，我们提出的 VMOF 在 36 个双目标大规模多目标优化问题测试实例中，有 21 个取得了最佳结果。只有 WOF 有两项结果显著优于我们提出的方法，而其他算法的最佳结果在 Wilcoxon 秩和检验下均未达到统计学上的显著优越性。

在比较维度从 100 增至 1,000，以及从 5,000 增至 20,000 时，WOF、LMOCSO 和 DGEA 的性能均有所下降。与 VMOF 相比，WOF 显著较差的结果更多（从 12 个增至 16 个），而 LMOCSO 和 DGEA 的较优结果则有所减少。不过，我们提出的算法 VMOF 获得的最佳结果更多（从 19 个增至 21 个）。随着决策变量数量的增加，我们的算法获得的最佳结果数量也随之增加；在求解三目标大规模多目标优化问题时，也观察到了这一规律（参见补充材料）。

我们提出的方法 VMOF 在大规模多目标优化问题的超体积指标方面也展现出优越的结果。具体而言，我们的方法分别在 16 个和 19 个双目标与三目标优化问题中取得了最佳结果（参见表 S-VI 和 S-VII）。总而言之，在求解大规模多目标优化问题时，我们的算法在收敛性和多样性方面也取得了令人满意的结果。因此，我们推断，采样策略并且为解推荐方向，在解决大规模和超大规模多目标优化问题方面是有效的。

E. 在真实世界超大规模多目标优化问题上的性能

本研究提出的一个动机是求解面向实际应用的超大规模多目标优化问题。为验证我们的算法在实际问题上的性能，我们在 TREE 数据集上进行了测试，实验结果如表 VI 所示。可以看出，所提出的算法在求解真实世界的超大规模多目标优化问题时仍具有显著优势。

TREE 问题主要描述在多个约束和超大规模决策变量下对变压器变比误差（RE）进行估计。从 TREE 1-5 的结果可以明显看出，在 100,000 至 900,000 维的范围内，所提出的算法始终优于对比算法。然而，对于 TREE6，所有对比算法均未能在规定的终止条件内得到可行解。这可归因于 TREE 6 中存在额外约束，从而增加了问题的复杂性。总体而言，结果证实了我们的算法在求解现实世界中的超大规模多目标优化问题方面的有效性，并凸显了其优越性。

此外，图4展示了所提出的 VMOF 及对比方法在包含 100,000 个决策变量的 TREE 问题上得到的非支配解。图中表明，所提出框架得到的解集具有更好的收敛性和多样性。

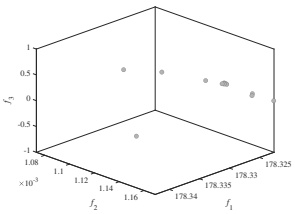
F. 收敛性分析

图 5(a) 和 (b) 分别展示了所有对比算法在具有 1,000 个和 100,000 个决策变量的双目标 LSMOP1 上取得的 IGD 值变化。由于部分算法的结果差异较大，为了更清楚地展示

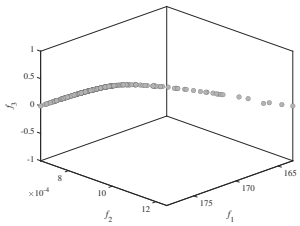
表VI 对比算法在 TREE 问题上获得的 IGD 指标。

问题	D	WOF	LSMOF	DGEA	VMOF	ROC
TREE1	100,000	9.71e+2-	9.55e+2-	9.62e+2-	**9.53e+2**	0.20%
TREE2	360,000	4.39e+3-	4.39e+3-	4.40e+3-	**4.36e+3**	0.80%
TREE3	96,000	9.60e+2-	8.72e+2-	9.60e+2-	**5.08e+2**	41.70%
TREE4	120,000	1.20e+3-	1.20e+3-	1.20e+3-	**1.05e+3**	12.81%
TREE5	900,000	9.28e+3-	8.90e+3-	9.20e+3-	**8.16e+3**	9.06%
TREE6	192,000	-	-	-	-	-
(+/-/=)		0/5/1	0/5/1	0/5/1		

每行中的最佳结果以粗体突出显示。耗时超过 50,000 秒的结果以「-」表示。

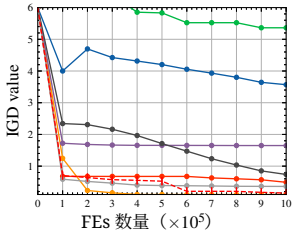


(a) DGEA 求解 TREE1。

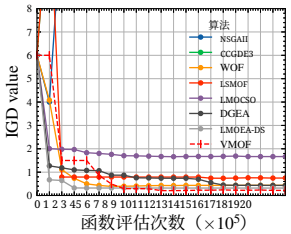


(b) VMOF 求解 TREE1。

图4.使用100,000个决策变量时，各比较方法在TREE上获得的非支配解。有关其他算法，请参阅补充材料。

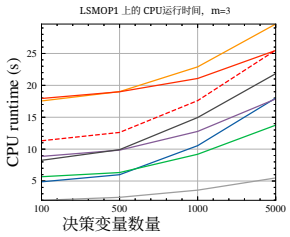


(a) 1,000 个决策变量。

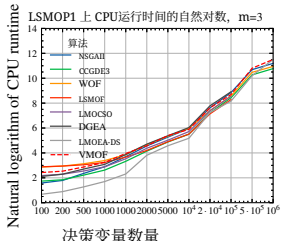


(b) 100,000 个决策变量。

图5。所有对比算法在分别具有1,000和100,000个决策变量的双目标 LSMOP1上的收敛曲线。



(a) 决策的维度变量变化范围为 100 到 20 ,000.



(b) The 维度 of the 决策 sion 变量 from 100 to 1,000,000。 变化范围为

图 6. 所有对比算法求解双目标 LSMOP1 时的运行时间，其中 FE=100,000。所有结果请参见补充材料中的图 S-1 至 S-9。

表现较好的算法之间的差距，对图进行了缩放。因此，部分图中省略了一些算法的结果。

从图中可以看出，大多数算法都能求解具有 1000个决策变量的 LSMOP1，但当问题扩展到 100,000 个维度时，许多算法无法求解。具体而言，NSGA-II和 CCGDE3 未能收敛，而 LSMOF、WOF和 DGEA能够在较少的函数评估次数内快速搜索到一个局部最优解，但随后未能收敛。只有我们提出的算法在给定的函数评估次数限制内输出了良好的结果。

G. 时间复杂度分析

作为一种用于求解超大规模多目标优化问题和大规模多目标优化问题的算法，时间复杂度是一项重要的评价指标。图6(a)和(b)展示了对比算法的时间复杂度。决策变量的维度范围分别为100至5,000和100至1,000,000。为了更好地展示结果，图6(b)绘制了决策变量维度从100至1,000,000时CPU运行时间的自然对数。

首先，从图 6.(a) 和 (b) 可以看出，尽管我们算法的时间复杂度并非最低，但其复杂度仍不高。因此，维度带来的影响相对有限，我们的算法属于所有运行算法中的第一梯队。此外，时间复杂度随决策变量维度增加的增长速率与大多数算法相近。其次，如图 6.(b) 所示，从 20,000 维增加到 100,000 维时，所有算法的时间复杂度都呈指数级增长，这充分说明处理如此高维的问题极具挑战性。

H. 消融研究

为验证方向采样和方向微调的有效性，我们进行了消融实验，结果见表VII。我们设计了VMOF算法的两个变体：一个不包含方向采样组件（W/O DS），另一个不包含方向微调组件（W/O DF）。实验在三目标大规模多目标优化问题上进行，决策变量的维度从10,000变化到500,000。

首先，从表VII可以看出，VMOF的两个变体在部分测试实例上都取得了最佳结果，这表明这两个组件均有助于提升VMOF算法的性能。其次，从最终结果可以看出，VMOF在36个测试案例中的22个上取得最佳结果，而消融变体在36个测试案例中的16个上取得最佳结果（其中两个测试案例并列）。在这16个最佳结果中，13个是在不使用方向采样的情况下取得的，5个是在不进行微调的情况下取得的（同样有两个测试案例并列）。因此，总体而言，尽管结果并非完全明确，但同时采用方向采样和微调的完整VMOF总体表现最佳。

I. Parameter Sensitivity Analysis

在所提出的 VMOF 中，参数 n_d 用于直接控制 Thomp- son 采样所推荐的方向数量，该参数还会间接控制

表VII 消融研究获得的三目标超大规模多目标优化问题和大规模多目标优化问题的IGD 指标值

问题	D	无 DR	不使用 DF	VMOF
LSMOP1	10,000	**1.96e-01+**	8.61e-01-	2.50e-01
	20,000	**1.37e-01=**	8.56e-01-	1.67e-01
	100,000	2.40e-01=	8.55e-01-	**2.00e-01**
	500,000	1.98e-01=	8.57e-01-	**1.82e-01**
LSMOP2	10,000	**8.12e-02=**	1.52e-01-	8.76e-02
	20,000	9.35e-02=	**5.89e-02=**	7.81e-02
	100,000	1.20e-01-	5.31e-02=	**5.02e-02**
	500,000	6.32e-02=	6.83e-02=	**5.05e-02**
LSMOP3	10,000	8.61e-01=	**8.61e-01=**	8.61e-01
	20,000	**8.61e-01=**	**8.61e-01=**	**8.61e-01**
	100,000	**8.61e-01=**	**8.61e-01=**	**8.61e-01**
	500,000	**8.58e-01=**	8.61e-01=	8.61e-01
LSMOP4	10,000	**5.91e-02=**	6.72e-02=	8.54e-02
	20,000	7.60e-02=	**5.45e-02=**	7.05e-02
	100,000	6.75e-02=	8.04e-02=	**3.30e-02**
	500,000	6.63e-02=	6.53e-02=	**5.72e-02**
LSMOP5	10,000	**4.98e-01=**	9.46e-01-	5.28e-01
	20,000	**5.01e-01=**	9.46e-01-	5.35e-01
	100,000	5.29e-01=	9.46e-01-	**5.01e-01**
	500,000	5.24e-01-	9.46e-01-	**3.60e-01**
LSMOP6	10,000	**7.33e-01+**	1.70e+00-	8.29e-01
	20,000	**7.63e-01+**	1.71e+00-	1.33e+00
	100,000	1.71e+00-	1.71e+00-	**7.05e-01**
	500,000	8.11e-01-	1.71e+00-	**7.05e-01**
LSMOP7	10,000	**8.35e-01=**	1.71e+00-	8.35e-01
	20,000	8.34e-01-	9.19e-01-	**7.63e-01**
	100,000	8.35e-01=	9.14e-01-	**8.34e-01**
	500,000	8.34e-01-	1.71e+00-	**6.33e-01**
LSMOP8	10,000	**8.03e-02=**	5.44e-01-	8.66e-02
	20,000	9.64e-02=	5.00e-01-	**7.19e-02**
	100,000	9.88e-02=	3.50e-01-	**8.01e-02**
	500,000	1.12e-01-	5.97e-01-	**7.46e-02**
LSMOP9	10,000	1.14e+00-	1.54e+00-	**5.88e-01**
	20,000	1.14e+00-	1.54e+00-	**5.87e-01**
	100,000	1.15e+00-	1.54e+00-	**5.87e-01**
	500,000	9.34e-01-	1.54e+00-	**5.86e-01**
(+/-/=)		3/11/22	0/25/11	

VMOF W/O DS 是不基于汤普森采样进行方向采样的方法，VMOFW/O VOis 是不采用方向微调方法的方法。每行中的最佳结果以粗体突出显示。

进化方向微调阶段的方向种群数量。为分析该参数对 VMOF 性能的影响，我们在一组包含 1,000 个决策变量的双目标 LSMOP 问题上进行了实验。参数 n_d 分别设为 5、10、20、25 和 50，对应的多样性解数量分别为 20、10、5、4 和 2。实验结果如图 7 所示。可以观察到，我们提出的方法的性能并未受到 n_d 不同设置的显著影响。这在问题 LSMOP2、LSMOP3 中尤为明显，

在 LSMOP4、LSMOP7 和 LSMOP9 上如此；对于其他问题，性能虽有一定波动，但结果仍优于其他最先进算法。

这一结果可归因于方向采样和方向微调的同时使用。当推荐方向数量不足时，微调过程中方向种群的规模会增大，从而弥补推荐方向数量不足的问题。相反，当方向种群的规模为不足时，推荐方向的数量会增加。因此，结合上一节的消融实验，我们提出的算法通过推荐方向与微调方向相互补充，提升了算法的整体性能。

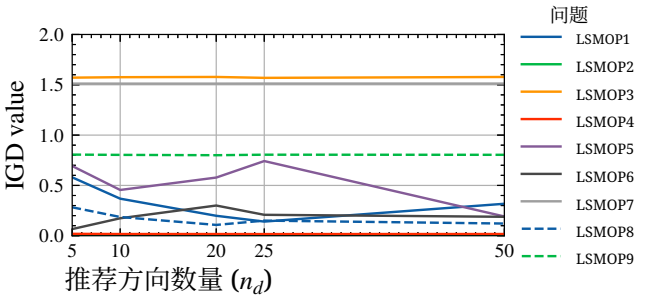


图 7. VMOF 在采样方向数量 n_d设置不同时所取得的 IGD结果统计。

J. 讨论

我们开展了一系列实验，以验证所提出算法的有效性。首先，我们选取了一组具有代表性的多目标进化算法，在超大规模多目标优化问题、VLSMOPs* 和大规模多目标优化问题上进行比较；结果表明，我们的算法在大多数测试实例上表现更优。随后，我们展示了 IGD变化和时间复杂度的可视化结果，以验证所提框架的效率。最后，我们针对方向采样和微调开展了消融实验和参数敏感性分析。实验结果证实，这两个组件都能提升搜索结果，而且我们的方法对这些超参数具有鲁棒性。

VI. 结论与未来工作

决策变量数量不断增加的多目标优化问题广泛存在于工业、经济和科学研究领域。在本研究中，我们将决策变量超过 100,000 的超大规模多目标优化问题进行了形式化定义，并指出这类问题在现实世界中十分普遍。因此，我们提出了一种基于方向采样和微调的框架，用于求解超大规模多目标优化问题。该算法采用汤普森采样对方向进行采样，并推荐能够引导种群在高维搜索空间中进化的方向；同时采用方向微调方法，对每个解的方向进行微调，从而有效求解超大规模问题。

为了全面理解并验证所提出的VMOF的有效性，我们对该框架进行了综合分析，并在广泛的实例上将该算法与最先进的算法进行了基准比较。实验结果证实，方向采样和方向微调策略在解决超大规模多目标优化问题方面尤为有效。未来的工作可以考虑引入用于求解多目标优化问题的先进技术 [51]–[54] ，这仍然是一个值得探索的方向。

此外，鉴于此类超大规模问题具有广泛的实际应用价值，这一重要问题值得进一步研究。

致谢

本研究得到中国国家自然科学基金（编号：62276222）的资助，并部分得到厦门市公共技术服务平台项目（编号：3502Z20231043）的资助。

参考文献

[1] K. Deb、A. Pratap、S. Agarwal 和 T. Meyarivan，“一种快速且精英的多目标遗传算法：NSGA-II”，*IEEE*《进化计算汇刊》，第6卷，第2期，第182–197页，2002年。[2] V. A. Shim、K. C. Tan 和 H. Tang，“用于进化多目标优化的自适应模因计算”，*IEEE*《控制论汇刊》，第45卷，第4期，第610–621页，2015年。[3] Y. Tian、L. Si、X. Zhang、R. Cheng、C. He、K. c. Tan 和 Y. Jin，“进化大规模多目标优化：综述”，*ACM*计算调查，第1卷，第1期，2021年。[4] W.-J. Hong、P. Yang 和 K. Tang，“大规模多目标优化的进化计算：十年进展”，国际自动化与计算期刊，第1–15页，2021年。[5] Y. Tian、C. Lu、X. Zhang、K. C. Tan 和 Y. Jin，“利用无监督神经网络求解具有稀疏最优解的大规模多目标优化问题”，*IEEE*《控制论汇刊》，第51卷，第6期，第3115–3128页，2021年。[6] Z. Yang、K. Tang 和 X. Yao，“利用协同协进化进行大规模进化优化”，信息科学，第178卷，第15期，第2985–2999页，2008年。[7] Y. Tian、X. Zhang、C. Wang 和 Y. Jin，“一种用于大规模稀疏多目标优化问题的进化算法”，*IEEE*《进化计算汇刊》，第24卷，第2期，第380–393页，2020年。[8] C. Qian，“面向大规模含噪子集选择的分布式帕累托优化”，*IEEE*《进化计算汇刊》，第24卷，第4期，第694–707页，2020年8月。[9] H. Hong、M. Jiang、L. Feng、Q. Lin 和 K. C. Tan，“通过注意力机制平衡探索与利用以求解大规模多目标优化问题”，载于*2022*年 *IEEE*进化计算大会（*CEC*），2022年，第1–8页。[10] L. M. Antonio 和 C. A. C. Coello，“利用协同协进化求解大规模多目标优化问题”，载于*2013*年 *IEEE*进化计算大会，2013年，第2758–2765页。[11] H. Zille、H. Ishibuchi、S. Mostaghim 和 Y. Nojima，“基于问题转换的大规模多目标优化框架”，*IEEE*《进化计算汇刊》，第22卷，第2期，第260–275页，2018年。[12] Y. Tian、X. Zheng、X. Zhang 和 Y. Jin，“基于竞争群体优化器的高效大规模多目标优化”，*IEEE*《控制论汇刊》，第50卷，第8期，第3696–3708页，2020年。[13] C. He、R. Cheng、C. Zhang、Y. Tian、Q. Chen 和 X. Yao，“用于电压互感器比差估计的进化大规模多目标优化”，*IEEE*《进化计算汇刊》，第24卷，第5期，第868–881页，2020年。[14] D. Donoho，“高维数据分析：维度灾难与福祉”，*AMS*数学挑战讲座，第1–32页，2000年1月。[15] J. Bobadilla、F. Ortega、A. Hernando 和 A. Guti´errez，“推荐系统综述”，知识库系统，第46卷，第109–132页，2013年。 [在线]。可访问：https://www.sciencedirect.com/science/ article/pii/S0950705113001044[16] S. Li、W. Lei、Q. Wu、X. He、P. Jiang 和 T.-S. Chua，“无缝统一属性与物品：面向冷启动用户的对话式推荐”，*ACM*信息系统汇刊，第39卷，第4期，2021年8月。[17] L. M. Antonio、C. A. C. Coello、S. G. Brambila、J. F. Gonz´alez 和 G. C. Tapia，“用于大规模多目标优化问题的运算分解”，载于遗传与进化计算会议论文集（*Companion*），丛书 GECCO ’19。美国纽约州纽约：计算机协会，2019年，第225–226页。[18] F. Sander、H. Zille 和 S. Mostaghim，“从单目标到多目标分组机制的迁移策略”，载于遗传与进化计算会议论文集，丛书 GECCO ’18。美国纽约州纽约：计算机协会，2018年，第729–736页。[19] H. Chen、X. Zhu、W. Pedrycz、S. Yin、G. Wu 和 H. Yan，“Pea算法：通过分离大规模多目标优化中的收敛性和多样性实现并行进化算法”，载于《2018年 IEEE 第38届分布式计算系统国际会议（ICDCS）》，2018年，第223–232页。[20] X. Ma、F. Liu、Y. Qi、X. Wang、L. Li、L. Jiao、M. Yin 和 M. Gong，“一种基于决策变量分析的大规模变量多目标优化问题进化算法”，IEEE《进化计算汇刊》，第20卷，第2期，第275–298页，2016年。[21] X. Zhang、Y. Tian、R. Cheng 和 Y. Jin，“一种基于决策变量聚类的大规模多目标优化进化算法”，IEEE《进化计算汇刊》，第22卷，第1期，第97–112页，2018年。[22] H. Qian 和 Y. Yu，“利用低有效维度求解高维多目标优化问题”，载于《第三十一届 AAAI 人工智能会议论文集》，系列 AAAI’17。AAAI Press，2017年，第875–881页。[23] R. Liu、R. Ren、J. Liu 和 J. Liu，“一种基于聚类和降维的大规模多目标问题进化算法”，《应用软计算》，第89卷，第106120页，2020年。[24] C. He、L. Li、Y. Tian、X. Zhang、R. Cheng、Y. Jin 和 X. Yao，“通过问题重构加速大规模多目标优化”，IEEE《进化计算汇刊》，第23卷，第6期，第949–961页，2019年。[25] S. Liu、J. Li、Q. Lin、Y. Tian 和 K. C. Tan，“学习加速大规模多目标优化的进化搜索”，IEEE《进化计算汇刊》，第1–1页，2022年。[26] J.-H. Yi、L.-N. Xing、G.-G. Wang、J. Dong、A. V. Vasilakos、A. H. Alavi 和 L. Wang，“NSGA-III 中交叉算子在大规模优化问题中的行为”，《信息科学》，第509卷，第470–487页，2020年。[27] C. He、R. Cheng 和 D. Yazdani，“用于进化式大规模多目标优化的自适应子代生成”，IEEE《系统、人和控制论汇刊：系统》，第1–13页，2020年。[28] W. Hong、K. Tang、A. Zhou、H. Ishibuchi 和 X. Yao，“一种可扩展的基于指标的大规模多目标优化进化算法”，IEEE《进化计算汇刊》，第23卷，第3期，第525–537页，2019年。[29] Z. Wang、H. Hong、K. Ye、G.-E. Zhang、M. Jiang 和 K. C. Tan，“通过生成对抗网络实现大规模多目标优化的流形插值”，IEEE《神经网络与学习系统汇刊》，第1–15页，2021年。[30] S. Qin、C. Sun、Y. Jin、Y. Tan 和 J. Fieldsend，“由定向采样辅助的大规模进化多目标优化”，IEEE《进化计算汇刊》，第25卷，738，2021年 4，724第。页 – 。[31] Y. Tian、Y. Feng、X. Zhang 和 C. Sun，“一种用于超大规模稀疏多目标优化的快速聚类进化算法”，《IEEE/CAA 自动化学报》，第1–16页，2022年。[32] L. Li、C. He、R. Cheng、H. Li、L. Pan 和 Y. Jin，“一种用于百万维多目标优化的快速采样进化算法”，《群体与进化计算》，第75卷，第101181页，2022年。[33] O. Chapelle 和 L. Li，“对汤普森采样的实证评估”，载于《神经信息处理系统进展》，J. Shawe-Taylor、R. Zemel、P. Bartlett、F. Pereira 和 K. Weinberger 编，第24卷。Curran Associates, Inc.，2011年。[34] W. R. Thompson，“根据两组样本的证据，一种未知概率大于另一种的可能性”，《生物计量学》，第25卷，第3/4期，第285–294页，1933年。[35] W. R. Thompson，“关于分配理论”，《美国数学杂志》，第57卷，第2期，第450–456页，1935年。[36] D. J. Russo、B. Van Roy、A. Kazerouni、I. Osband 和 Z. Wen，“汤普森采样教程”，《机器学习基础与趋势》，第11卷，第1期，第1–96页，2018年7月。[37] D. Agarwal、B. Long、J. Traupman、D. Xin 和 L. Zhang，“Laser：面向在线广告的可扩展响应预测平台”，载于《第7届 ACM 网络搜索与数据挖掘国际会议论文集》，系列 WSDM ’14。美国纽约州纽约市：计算机协会，2014年，第173–182页。[38] J. Kawale、H. Bui、B. Kveton、L. T. Thanh 和 S. Chawla，“用于在线矩阵分解推荐的高效汤普森采样”，载于《第28届神经信息处理系统国际会议论文集——第1卷》，系列 NIPS’15，2015年，第1297–1305页。[39] K. Kandasamy、A. Krishnamurthy、J. Schneider 和 B. Poczos，“通过汤普森采样实现并行贝叶斯优化”，载于《第二十一届人工智能国际会议论文集和

统计学，系列：机器学习研究论文集，A.Storkey和 F. Perez-Cruz编，卷84。PMLR，2018 年 4 月 9–11 日，第133–142页。[40] L. Sun和 K. Li，“基于动态汤普森采样的自适应算子选择方法”，载于自然界并行问题求解——*PPSN XVI*，T. B¨ack、M. Preuss、A. Deutz、H. Wang、C. Doerr、M. Emmerich 和 H. Trautmann 编。Cham：Springer 国际出版公司，2020年，第 271–284页。[41] S. Chakraborty、S.Roy 和A. Tewari，“高维稀疏线性上下文老虎机的汤普森采样”，载于第*40*届国际机器学习会议，系列：机器学习研究论文集，A. Krause、E.Brunskill、K.Cho、B. Engelhardt、S. Sabato 和J. Scarlett 编，卷 202。PMLR，2023 年 7月 23–29 日，第 3979–4008 页。 [在线]。可访问：https://proceedings.mlr.press/v202/chakraborty23b.html [42] N. Gupta、O.-C.Granmo和 A. Agrawala，“动态多臂老虎机的汤普森采样”，载于*2011*年第*10*届国际机器学习与应用会议及研讨会，第 1 卷，2011年，第 484–489页。[43] E.Kaufmann、N. Korda和 R. Munos，“汤普森采样：渐近最优有限时间分析”，载于算法学习理论，N. H. Bshouty、G. Stoltz、N. Vayatis 和T.Zeugmann 编。柏林、海德堡：Springer柏林·海德堡，2012年，第199–213页。[44] C. He、S. Huang、R. Cheng、K.C. Tan 和 Y. Jin，“生成对抗网络（GAN）驱动的进化多目标优化”，*IEEE*《控制论汇刊》，卷 51，第 6 期，第3129–3142页，2021年。[45] 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》，卷12，第 4期，第 73–87 页，2017年。[46] R. Cheng、Y. Jin、M. Olhofer 和 B. Sendhoff，“大规模多目标和多目标优化的测试问题”，*IEEE*《控制论汇刊》，卷47，第12 期，第 4108–4121 页，2017年。[47] E.Zitzler、L. Thiele、M. Laumanns、C.M. Fonseca和 V.G.d. Fonseca，“多目标优化器的性能评估：分析与综述”，*IEEE*《进化计算汇刊》，卷7，第 2 期，第117–132 页，2003年。[48] L.While、P. Hingston、L. Barone 和 S. Huband，“一种更快速的超体积计算算法”，

IEEE《进化计算汇刊》，第 10 卷，第 1 期，第 29–38 页，2006年。[49] W. Haynes，《威尔科克森秩和检验》。纽约州纽约市：施普林格纽约，2013年，第 2354–2355 页。[50] H. Gu、H. Wang 和 Y. Jin，“帕累托集对基于问题重构的大规模多目标优化算法性能的影响”，载于 *2023*年 *IEEE*进化计算大会（*CEC*），2023年，第 1–8 页。[51] M. Jiang、Z. Wang、L. Qiu、S. Guo、X. Gao 和 K. C. Tan，“一种基于流形迁移学习的快速动态进化多目标算法”，*IEEE*《控制论汇刊》，第 51 卷，第 7 期，第 3417–3428 页，2021年。[52] M. Jiang、Z. Wang、S. Guo、X. Gao 和 K. C. Tan，“用于动态多目标优化的个体级迁移学习”，*IEEE*《控制论汇刊》，第 51 卷，第 10 期，第 4968–4981 页，2021年。[53] J. Ang、K. Tan 和 A. Mamun，“一种用于规则提取的进化模因算法”，《应用专家系统》，第 37 卷，第 2 期，第 1302–1315 页，2010年。 [在线]。可访问：https: //www.sciencedirect.com/science/article/pii/S095741741409005545 [54] X. Qiu、J.-X. Xu、Y. Xu 和 K. C. Tan，“一种用于稳健设计中极小极大优化的新型差分进化算法”，*IEEE*《控制论汇刊》，第 48 卷，第 5 期，第 1355–1368 页，2018年。



Haokai Hong 分别于2020年和2023年获得中国厦门大学计算机科学学士和硕士学位。目前，他是中国香港特别行政区香港理工大学计算学系的博士候选人。

他的研究主要关注进化计算和机器学习领域。



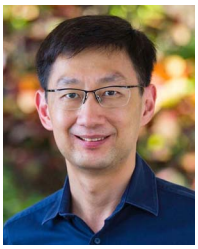
Min Jiang （M’11-SM’12）于2001年和2007年分别获得中国武汉大学计算机科学学士学位和博士学位。之后，他在厦门大学数学系从事博士后研究。目前，他是厦门大学人工智能系教授。他的主要研究兴趣包括机器学习、计算智能及其相互作用。Jiang博士目前担任*IEEE*《进化计算汇刊》、*IEEE*

神经网络与学习系统汇刊，*IEEE*《计算智能杂志》和认知与发展系统汇刊。他担任 IEEE CIS 厦门分会主席。



Qiuzhen Lin（IEEE会员）于2007年获得肇庆学院（中国肇庆）学士学位，2010年获得深圳大学（中国深圳）硕士学位，并于2014年获得香港城市大学电子工程学系（中国香港）博士学位。他现任深圳大学计算机科学与软件工程学院副教授。自2008年以来，他已发表或合著100多篇研究论文。他的研究兴趣包括人工免疫系统，

多目标优化和动态系统。他是*IEEE*《进化计算汇刊》和*IEEE*计算智能新兴主题汇刊。



Kay Chen Tan （IEEE会士）于1994年和1997年分别获得英国格拉斯哥大学工学学士学位（一等荣誉）和博士学位。他现任香港香港理工大学计算学系计算智能讲座教授。Tan教授已独立或合作撰写了300多篇经同行评审的论文和七本书籍。目前，他担任美国IEEE计算智能学会出版事务副主席。他曾任.....的主编。

IEEE《计算智能杂志》自2010年至2013年，以及*IEEE*《进化计算汇刊》自2015年至2020年，现任十多家期刊的编委会成员。他是施普林格《机器学习：基础、方法与应用》丛书的首席联合主编。