

对一种基于自适应分组解决高维昂贵多目标优化问题的算法的学习和复现

摘要

本文学习并复现了一种基于自适应分组的算法 (AVG-SAEA)，用于解决高维昂贵多目标优化问题 (EMOPs)。高维 EMOPs 具有搜索空间稀疏、评估成本高昂和多目标冲突等挑战。AVG-SAEA 通过自适应变量分组降低问题复杂性，并结合代理模型减少评估成本，同时优化解集的质量和多样性。实验结果表明，AVG-SAEA 在 50-200 维的 DTLZ 和 WFG 测试问题上表现出色，并在与现有算法的比较中显示出优势。此外，算法在更高维度的 LSMOP 测试问题和真实问题 TREE 上也表现出良好的性能。本文所复现的算法为高维 EMOPs 的求解提供了新的思路和方法。

关键词：高维昂贵多目标优化问题；自适应变量分组；代理模型

1 引言

高维昂贵多目标问题 (Expensive Multi-objective Optimization Problems, EMOPs) 是优化领域的核心难题之一，其复杂性主要体现在三个方面：高维性、昂贵性和多目标性 [1]。

高维问题的第一个挑战在于搜索空间的稀疏性。随着维度的增加，变量之间的组合可能性呈指数级增长，高维空间中的解分布变得极为稀疏，算法在解空间中难以有效定位最优解。此外，变量之间通常存在复杂的非线性依赖关系，这种耦合性使得简单的优化方法难以捕捉变量之间的相互作用，导致优化效果不佳。在高维场景中，数据的稀疏性也进一步加剧了代理模型性能的下降，模型难以准确拟合目标函数，优化效率显著降低。针对这些问题，研究者提出了多种解决办法。通过维度降解技术，例如主成分分析和变量分组，将高维变量映射到低维空间，降低问题的复杂性。协同进化方法通过将高维变量划分为若干子集，并在子空间中分别优化，有效缓解了维度灾难。同时，采用动态调整策略的代理模型，例如集成高斯过程与神经网络，可以提高模型在高维场景中的适应性和准确性。

昂贵问题的主要挑战在于目标函数的评估成本过高，往往需要耗费巨大的计算资源。例如原油蒸馏装置设计 [2]、翼型设计、制造工程等，在工业仿真或物理实验中，每次评估可能需要数小时甚至数天，这直接限制了可用的样本数量。由于样本稀缺，优化算法在探索解空间时面临严峻的资源分配问题。此外，数据不足还会导致代理模型的不确定性增加，预测结果偏离真实目标函数，从而影响优化路径。为了解决这些问题，代理模型辅助优化方法成为关键 [3]。通过使用高斯过程回归或径向基函数等代理模型替代真实目标函数，可以显著降低评估成本。在此基础上，不确定性采样策略被广泛应用，通过优先选择代理模型预测不确定性

较高的区域进行评估，有效提高了优化效率。主动学习技术则进一步结合动态采样策略，优化过程中根据不同阶段调整真实评估和代理评估的频率分配，最大化资源利用率。

多目标问题的难点在于需要同时优化多个可能冲突的目标函数。目标冲突使得问题不再有单一的全局最优解，而是需要通过寻找帕累托前沿来描述目标之间的权衡关系。此外，优化算法需要在解的质量和解集的多样性之间保持平衡。既要保证解尽可能接近帕累托前沿，也要覆盖整个解空间，以满足实际应用需求。随着目标数量的增加，优化算法的复杂性呈指数级增长，这进一步提高了问题求解的难度。多目标进化算法，例如 NSGA-II 和 MOEA/D [4]，已经成为解决多目标问题的主流方法，通过种群搜索机制，可以同时探索多个目标函数并维护解集的多样性。目标降维技术也被广泛应用，通过目标聚类或分解减少目标之间的冲突，提高算法的效率。为了保证解集的均匀分布，帕累托前沿维护策略被设计用于动态调整解的存档，例如基于拥挤距离的排序机制，能够有效保持解的多样性和代表性。

这些特点相互交织，使得传统的优化算法难以直接适用，从而对问题求解提出了严峻的挑战。深入分析这些特点及其对应的解决方法，不仅能够揭示问题的本质，还为后续的算法设计提供重要的理论依据。

2 相关工作

高维昂贵多目标优化问题广泛存在于工程设计、制造和能源等领域。然而，在解决由高维特性引发的搜索空间爆炸、样本稀疏和代理模型预测不准确等问题时，现有方法主要分为两类。一类是从根源出发想方设法通过降维解决决策变量维度高的问题。例如在 ADSAPSO [5] 中，设计了一种基于性能良好解和性能较差解之间决策变量差异的维度选择方法。另一类是从缓解由高维带来的评估消耗出发，设计更加简单高效的代理模型。例如在 EDN-ARMOEA [6] 中，采用了一种高效的 dropout 神经网络作为代理模型，该网络可以在一次运行中同时逼近多个目标函数，并利用所采用的 dropout 机制进行不确定性估计。其中决策变量分组策略作为降维的重要手段，在现有研究中并没有很好的效果。决策变量分组策略通过将高维决策空间划分为多个低维子空间，可以有效减轻优化负担。然而，传统的分组方法在面对昂贵优化问题时往往面临着一些矛盾：一类无需额外评估，但由于忽略变量交互，分组质量较差；另一类通过增加额外评估提升分组精度，但显著增加了优化成本，例如在基于图的深度分解方法 [7] 中，将挖掘出交互关系的决策变量转换成图，并通过最小顶点分隔符进行分组。这显然与昂贵、多目标的问题需要节约评估预算的目标相冲突。这种矛盾限制了变量分组策略在高维昂贵多目标优化问题中的应用。

因此，如何在不增加评估成本的情况下，高效地进行变量分组，成为解决高维 EMOPs 的关键问题之一。已有研究多集中于低维或单目标问题，而针对高维多目标优化问题的变量分组研究尚属空白

3 动机

在优化问题的研究中有一个重要的发现，并非所有的决策变量对解的收敛性影响都是均等的，一些决策变量在优化过程中扮演了更为关键的角色，它们的变化对整个解的收敛性有着显著的影响 [8]。据此，文章进一步探究决策变量对收敛性影响的差异，并尝试利用这种差

异对决策变量进行分组。这样不仅考虑了决策变量之间的信息使得分组效果好，而且因为分组方法的简单有效而不会带来不可接受的时间消耗，符合优化问题需要节约时间预算的目标。

此外，考虑到到高维空间中的解更容易聚集在一起，具体来说，在高维空间中，整个空间的体积迅速膨胀，使得高维空间中的距离度量变得不那么直观，原本在低维空间中相隔很远的点在高维空间中可能看起来接近，这种现象被称为“维度的诅咒”。因此如何在高维空间中有效地保持解的多样性和分布是十分重要的 [9]。为此，文章设计了一个自适应的采样和选择策略，能够根据优化需求实时调整，满足高维 EMOPs 在不同阶段对解的收敛性、多样性和模型不确定性的不同要求。

结合以上两个基本工作，文章提出了一种自适应变量分组的代理模型辅助进化算法 AVG-SAEA [10]。该算法通过对变量分组的动态调整，结合自适应环境选择和采样策略，在不增加评估预算的前提下，显著提高了高维 EMOPs 的优化效果。

在这项工作中，文章考虑以下昂贵的高维多目标优化问题：

$$\begin{aligned} \min_x \quad & \mathbf{f}(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s.t.} \quad & x \in X \end{aligned}$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_d)$ 为包含 d 个决策变量的决策向量， X 表示决策空间，目标向量 \mathbf{f} 由 m ($m \geq 2$) 个目标组成。此外，目标函数是一个黑盒子，需要昂贵耗时的数值模拟或物理实验来评估，为了解决优化问题，一般根据实验数值模拟收集的数据构建代理模型来减少真实计算目标函数的次数。在代理辅助进化优化的上下文中，通常认为具有 $d > 30$ 的问题被称为高维问题，还应该注意的，对于 MOPs，存在一组权衡帕累托最优解，称为帕累托集 (PS)，它是一个解集而不是一个优化所有目标的单一解，PS 在目标空间中的投影称为帕累托前沿 (PF)。

4 本文方法

AVG-SAEA 是一种针对高维昂贵多目标优化问题设计的算法，其核心思想在于结合变量分组策略和高效的代理模型，以减少维度复杂性、降低评估成本，并同时兼顾解集质量和多样性。算法的整体框架如图 1 所示，该方法主要分为两个部分，分别称为组间优化和组内优化。首先，AVG-SAEA 使用一种基于变量分组的机制对高维变量进行分解。具体来说，它将高维变量划分为若干组，使得每一组变量在解空间中具有相对独立的优化空间。这种分组策略通过分析变量之间的关联性来减少维度之间的耦合，从而降低优化问题的复杂性。在分组完成后，每个变量组被作为独立的子问题进行处理，每个子问题的优化目标可以聚焦于较低维度的变量集。接下来，为了应对昂贵问题中的评估开销，算法引入了基于代理模型的辅助优化技术。在优化过程中，AVG-SAEA 不直接调用真实的目标函数进行频繁评估，而是通过训练代理模型来近似目标函数。代理模型的构建采用高斯过程回归方法，其能够高效地捕捉输入变量与目标值之间的关系，并对未评估区域的预测值进行不确定性量化。这种不确定性量化使得算法可以在探索与开发之间取得平衡，从而更加智能地分配有限的计算资源。在代理模型的辅助下，AVG-SAEA 通过多目标进化算法 (MOEA) 生成候选解集。这些候选解由代理模型进行初步筛选，选出最有潜力的解。在此基础上，算法采用了一种动态采样策略，仅对筛选出的关键解点调用真实目标函数进行高精度评估。这种动态采样机制结合了探索与开

发，通过优先评估代理模型预测不确定性较大的区域，有效提升了优化效率，同时确保解的质量能够向真实帕累托前沿逼近。AVG-SAEA 在优化过程中还特别注重帕累托前沿的多样性维护。通过设计拥挤距离和适应度评价指标，算法能够平衡解集的分布性与最优性，确保生成的解集不仅接近帕累托前沿，而且在目标空间中均匀分布。此外，解集的更新采用了基于归档的维护策略，动态调整帕累托前沿的表示范围，以适应不同阶段优化过程的需求。最后，AVG-SAEA 在每轮迭代中重复上述步骤，即变量分组、代理模型训练、候选解生成与筛选、动态采样评估，以及帕累托前沿更新。通过多轮循环迭代，算法能够逐步提升解集的质量，并最终找到能够覆盖目标空间的高质量帕累托解集。

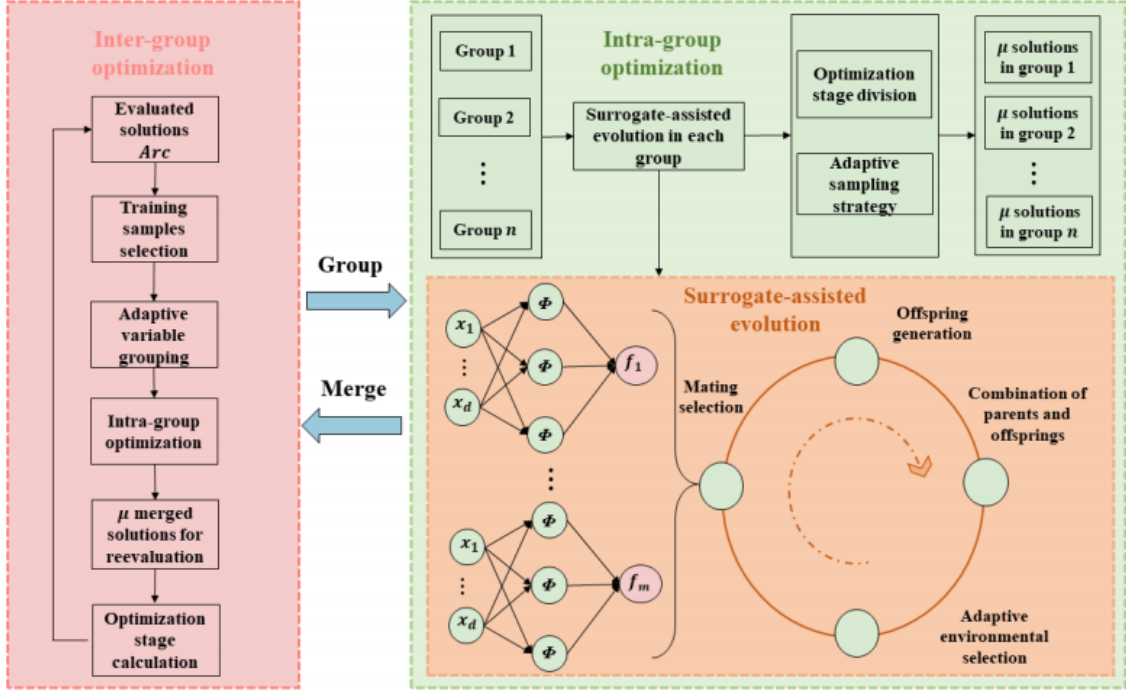


图 1. 算法框架图

5 复现细节

5.1 与已有开源代码对比

复现代码参考自安徽大学团队开发的 [Evolutionary multi-objective optimization platform](#) [11], 以下是关于代码的解释:

- CalFitness.m: 这个是从收敛性、多样性角度来计算一个比较各个解的适应度指标
- CalFlag.m: 通过计算 HV 和 the summation of normalized objective values (Con) 来决定下一代的 stage
- DimSelect.m: 通过非支配排序选择收敛好的和坏的解，用于计算分组策略
- DSelectNew.m: 环境选择: 先选择在各个目标上最好的 (收敛性)，然后再通过夹角去选 (多样性)

- DVsetEnvironmentalSelection.m: 环境选择: 先根据非支配排序去选 (收敛性), 然后再通过夹角去选 (多样性)
- EnvironmentalSelection.m: NSGA- 中的环境选择, 在本文是用于在组间进化时从 Arc 中选择个体进入组内进化
- MergeSubpop.m: 用于将组内进化得到的低纬度解拼接成高纬度解, 不过要记录归一化等数据处理的参数
- OperatorG.m: 通过给定的父本来生成子代
- SelectTrainData.m: 用来从真实评估的历史数据中选择一些数据去训练代理模型
- SubEnvironmentalSelection.m: 通过非支配排序进行选择
- AVGSAEA.m: 这个是主流程, 会调用其他的函数

5.2 复现过程

尽管 AVG-SAEA 是为了解决高维昂贵多目标优化问题而提出的, 但它本质上还是一个进化算法, 只不过为了解决由高维、昂贵、多目标等特性带来的挑战, 分别增加了一些处理对应问题的模块。从这个角度出发, 该算法的复现将基于最原始的进化算法尤其是遗传算法的代码, 一步一步加上处理高维、昂贵、多目标的代码块, 最终经调试得到整个 AVG-SAEA 的代码。实现过程如图 2 所示。

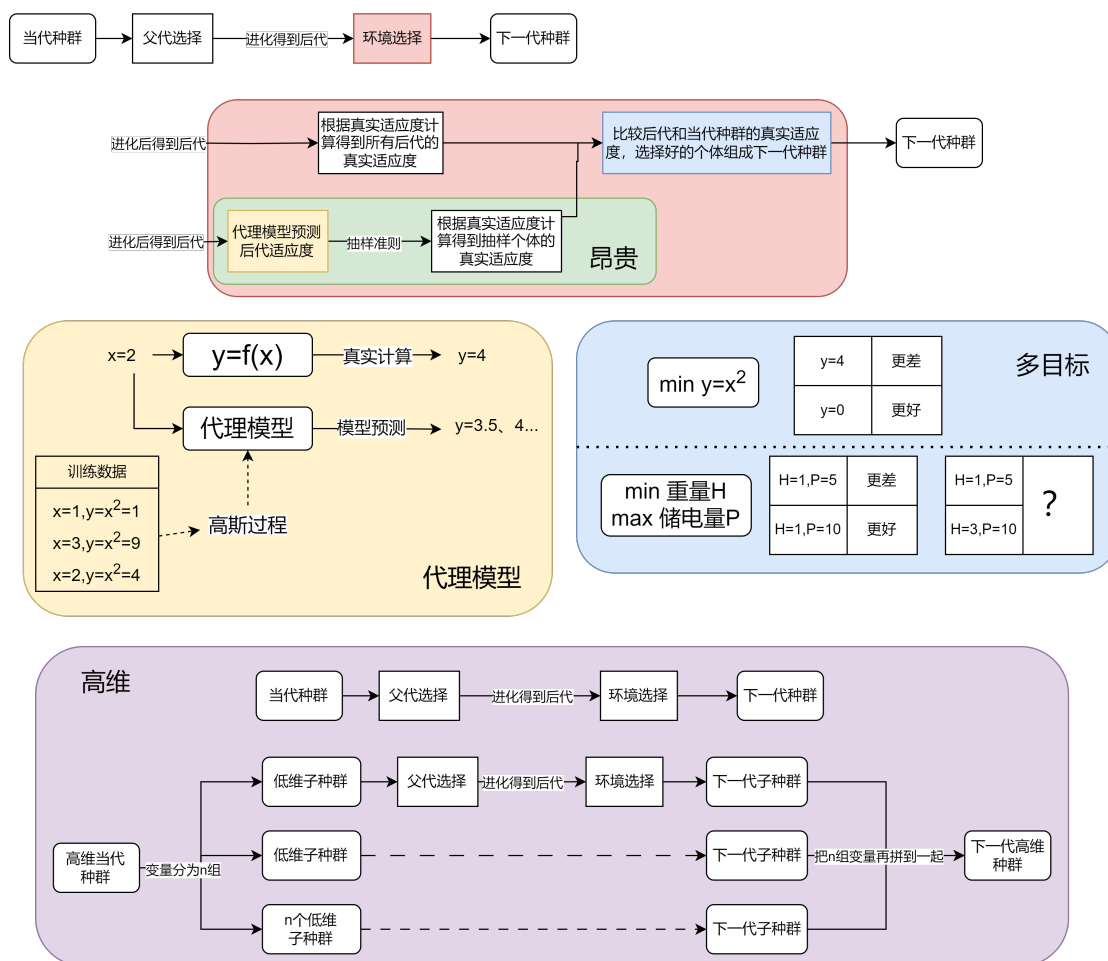


图 2. 复现过程图

基本的遗传算法进化一代的过程可以这样描述：从当代种群中根据适应度等指标进行父代选择，被选中的父代通过进化产生后代，环境选择是指在计算出后代的适应度等指标后，基于该指标比较后代和当代种群中个体的优劣，进而选择出下一代种群，以上过程重复执行直到达到终止条件，后面把基本的遗传算法称为 base algorithm (BA)。“昂贵”指的是真实获取个体适应度的过程十分昂贵，所以处理“昂贵”带来的挑战就是要在“计算适应度”这一步骤上做工作。适应度的计算属于环境选择中的一个步骤，如图 1 中红色部分所示，BA 中是直接根据真实适应度计算过程来得到个体的适应度，而在具有昂贵性质的问题中是通过代理模型来预测个体的适应度，不过代理模型往往结合抽样准则一起使用，如图 1 中绿色部分所示，这是为了矫正代理模型的预测准确性。代理模型的直观作用可以理解为图 2 中黄色部分所示，代理模型本质上就是一个预测模型，通常分为回归、分类和关系预测三个范式，它需要通过由具体训练数据的训练才能准确地进行预测。处理“多目标”带来的挑战则是要在“比较后代和当代种群的适应度”这一步骤上做工作。在 BA 中是可以简单地比较单个目标的适应度值来评价个体的优劣，而在多目标的场景下个体之间的比较变得复杂，这种复杂可以简单理解为如图 2 中蓝色部分所示，为此要增加非支配排序等处理多目标的模块。根据文章的描述，AVG-SAEA 是通过决策变量分组来实现降维从而解决高维性质带来的挑战，为此，在进行父代选择操作前就需要通过变量分组个体原始维度为 D 的种群分成 n 个个体维度为 D/n 的低维子种群，之后让各个子种群并行进化，最后再将低维的个体组合成原始维度 D 的个体作为

后代，直观表述如图 2 紫色部分所示。

此外在使用代理模型时需要考虑一些问题。首先是在构建代理模型前要先确定输入和输出是什么，用原始的输入输出还是要对数据做一些处理。其次要根据代理模型的输出选择合适的代理模型，这需要通过训练不同的模型并测试比较不同代理模型的预测性能来完成。最后要选择合适的填充抽样准则，它需要同时考虑解的质量和不确定性，不仅希望选择预测结果好的解进行真实评估，还希望探索解空间中未知或不确定的区域，因为如果预测结果差但是不确定性高，可能是由于预测错误而导致预测结果差。

5.3 实验环境搭建

- 处理器 (CPU): Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz。
- 内存 (RAM): 32GB。
- 显卡 (GPU): 无独立显卡。
- 操作系统: Windows 11 家庭中文版。
- 编程语言和环境: MATLAB R2023b。
- 实验软件和库: PlatEMO。
- 测试问题: DTLZ、WFG、LSMOP、TREE。
- 指标: IGD+、HV、Wilcoxon rank-sum test。
- 对比算法: K-RVEA、CSEA、EDN-ARMOEA，对比算法使用原文参数。

5.4 创新点

本文的两个主要创新点包括：首先，提出了一种自适应决策变量分组方法，该方法能够在不增加额外功能评估 (FEs) 的情况下，基于解的收敛性能差异自动调整分组策略，有效降低高维决策空间的维度；其次，设计了一种基于自适应环境选择的抽样策略，该策略根据优化过程中的不同阶段需求，动态选择和提供所需的解决方案进行重新评估，以提高优化效率。这些创新点在文中的图表中得到了直观展示，图 3 所示为自适应分组方法，图 4 所示为自适应的环境选择方法及抽样策略。

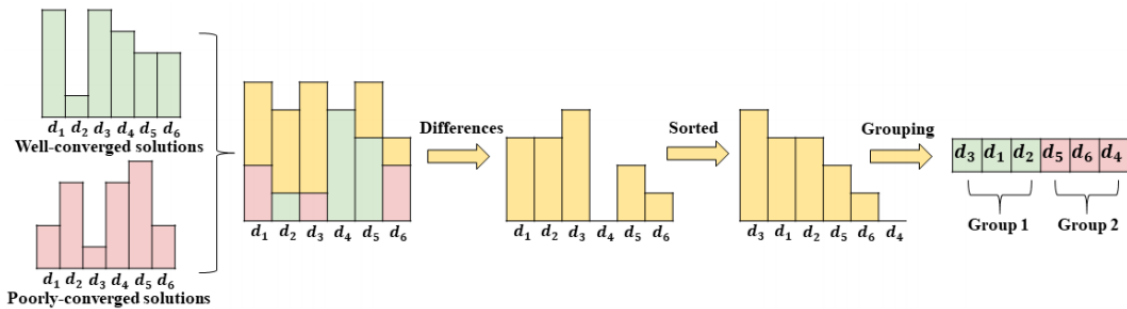


图 3. 自适应决策变量分组示意图

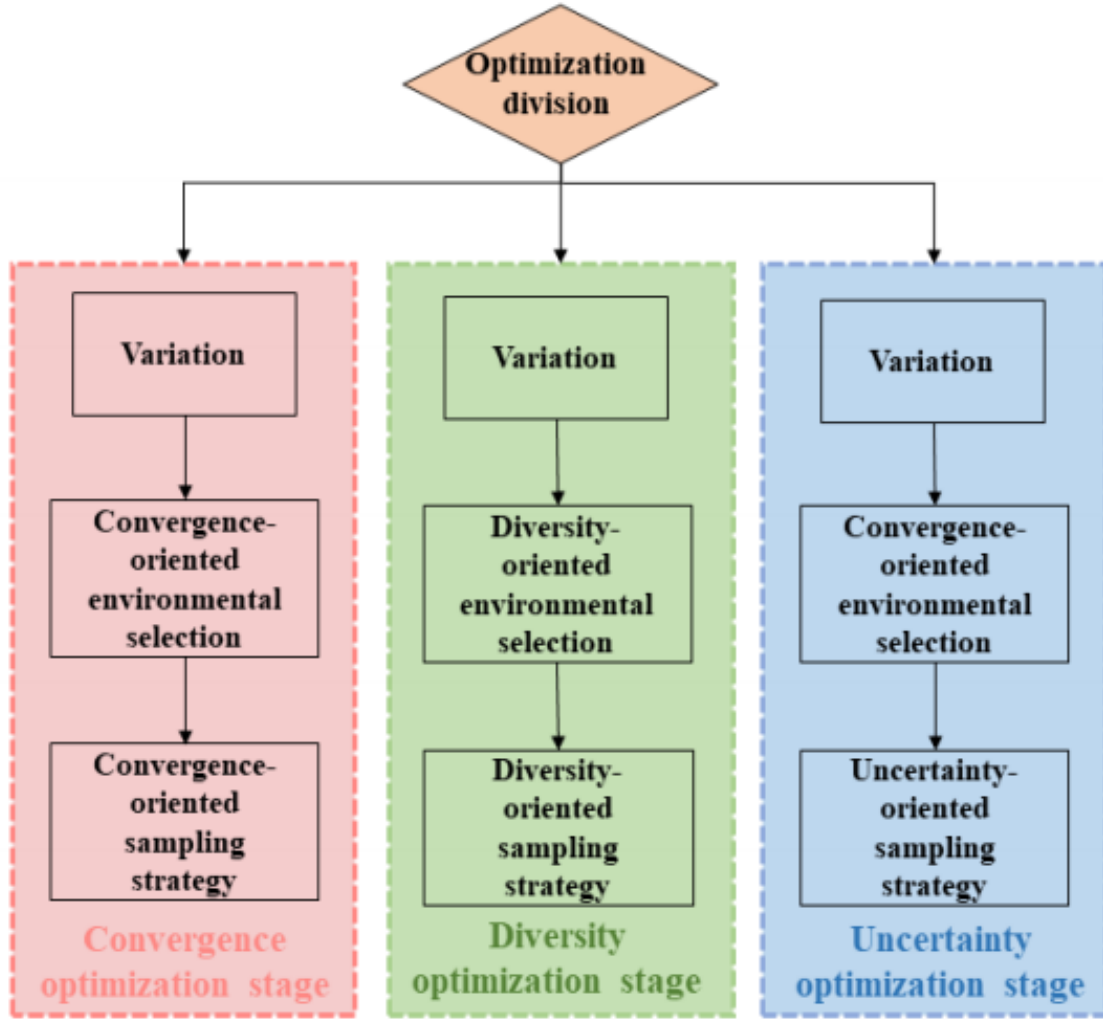


图 4. 基于自适应环境选择的抽样策略示意图

6 实验结果分析

6.1 在 50-200 维测试问题上验证文章方法的有效性

为了检验 AVG-SAEA 解决高维昂贵多目标优化问题的有效性，首先在 matlab 上让该算法去解决 DTLZ 和 WFG 测试问题集中的问题，具体来说，是从中选了三个具有不同特征（PF 有凸有凹，单峰多峰，可分离不可分离）的问题来测试 AVG-SAEA。在实验中设置决策变量维度为 50、100 和 200，以 IGD+ 作为评估指标，并将最终得到的近似解集可视化，用以直观将算法得到的近似帕累托前沿与问题的真实帕累托前沿进行比较。算法在不同问题上得到的 IGD+ 值如表 1 所示，从中选取 100 维的问题的结果进行可视化如图 5 所示。

表 1. AVG-SAEA 在 50-200 维测试问题上得到的 IGD+ 值表

| Problem | N | M | FE | D | AVG-SAEA |
|---------|-----|---|-----|-----|---------------------|
| DTLZ2 | 100 | 3 | 500 | 50 | 2.8039e+0 (1.54e-1) |
| | | | | 100 | 6.2916e+0 (2.56e-1) |
| | | | | 200 | 1.3908e+1 (1.93e-1) |
| WFG2 | 100 | 3 | 500 | 50 | 5.8498e+1 (4.87e-2) |
| | | | | 100 | 5.6226e+1 (8.24e-2) |
| | | | | 200 | 7.9325e+1 (3.45e-2) |
| WFG9 | 100 | 3 | 500 | 50 | 7.1359e+1 (9.98e-2) |
| | | | | 100 | 5.9006e+1 (2.13e-2) |
| | | | | 200 | 9.7108e+1 (4.45e-2) |

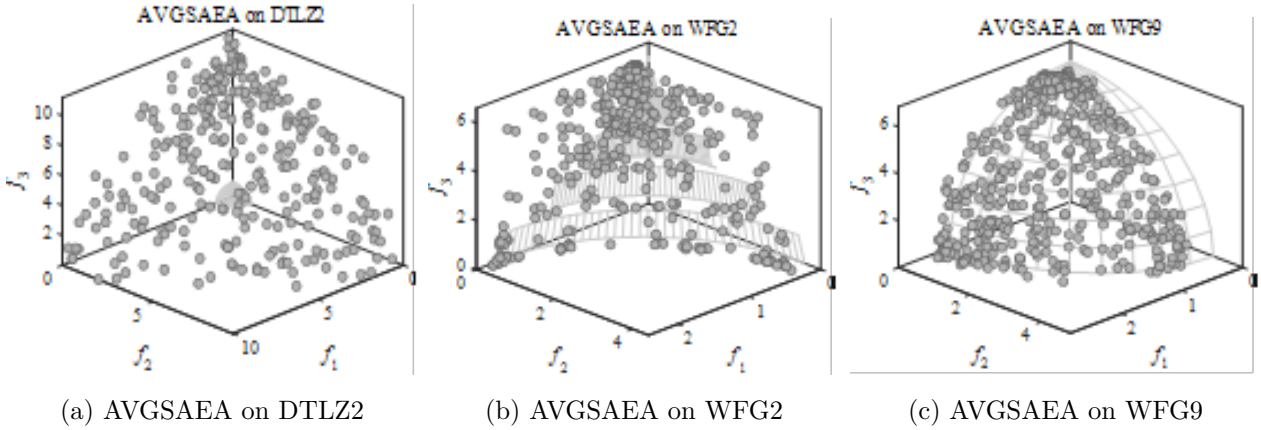


图 5. AVG-SAEA 在 100 维测试问题上得到的近似 PF 的可视化图

6.2 在 50-200 维测试问题上比较文章方法和对比算法的效果

将 AVG-SAEA 与现有的先进方法进行比较，凸显出 AVG-SAEA 在解决高维 EMOP 上的优势。具体做法是，在 DTLZ 测试套件和 WFG 测试套件下分别将决策变量维度设置为 50、100 和 200，与实验设置中的 3 个算法进行 IGD+ 性能比较。本地实验得到结果如表 2 所示，然而算法效果却并不和文章结果一致，目前还没有找到确切原因。

表 2. AVG-SAEA 与对比算法在 50-200 维测试问题上得到的 IGD+ 值表

| Problem | D | KRVEA | CSEA | EDNARMOEA | AVGSAEA |
|---------|-----|---------------------|---------------------|---------------------|--------------------|
| DTLZ2 | 50 | 2.7195e+0(1.35e-1)+ | 1.3870e+0(2.03e-1)+ | 2.7027e+0(1.35e-1)+ | 2.8039e+0(1.54e-1) |
| | 100 | 6.0493e+0(2.10e-1)+ | 3.9889e+0(3.81e-1)+ | 6.1120e+0(1.50e-1)+ | 6.2961e+0(2.56e-1) |
| | 200 | 1.3109e+1(3.53e-1)+ | 1.0666e+1(6.08e-1)+ | 1.3259e+1(3.48e-1)+ | 1.3908e+1(1.93e-1) |
| WFG2 | 50 | 8.2978e-1(2.40e-2)- | 5.7788e-1(4.71e-2)= | 8.2802e-1(2.14e-2)- | 5.8498e-1(4.87e-2) |
| | 100 | 8.0268e-1(1.51e-2)- | 6.5521e-1(3.62e-2)- | 8.0051e-1(1.26e-2)- | 5.6226e-1(8.24e-2) |
| | 200 | 7.8970e-1(6.84e-3)= | 7.1381e-1(2.25e-2)+ | 7.9361e-1(6.73e-3)= | 7.9325e-1(3.45e-2) |
| WFG9 | 50 | 9.4614e-1(2.26e-2)- | 8.0319e-1(7.11e-2)- | 9.4875e-1(2.20e-2) | 7.1359e-1(7.60e-2) |
| | 100 | 9.5743e-1(9.66e-3)- | 8.7927e-1(6.25e-2)- | 9.5062e-1(1.27e-2)- | 5.9006e-1(1.44e-1) |
| | 200 | 9.5301e-1(9.30e-3)+ | 9.2117e-1(4.27e-2)+ | 9.5335e-1(7.71e-3)= | 9.7108e-1(4.45e-2) |
| +/-/= | | 4/4/1 | | 5/3/1 | 3/4/2 |

6.3 在 500、1000 更高维测试问题上将文章算法与对比算法进行比较

为了更好地检验 AVG-SAEA 在高维优化问题中的进行, 实验进一步在具有更高决策空间维度的测试问题上进行, 分别设置为 500 和 1000 维, 选择使用 LSMOP 测试套件是因为该测试问题的性质使得它能更好地反映高维 MOP 带来的挑战。此外, 由于 K-RVEA 不是为解决高维优化问题设计的, 它在高维优化问题上的效果并不好, 因此在本次实验中没有比较的必要, 所以对比算法就剩下两个即 CSEA 和 EDNARMOEA。本地复现结果如表 3 所示。

表 3. AVG-SAEA 与对比算法在 500 和 1000 维测试问题上得到的 IGD+ 值表

| Problem | M | D | CSEA | EDNARMOEA | AVGSAEA |
|---------|---|------|---------------------|---------------------|---------------------|
| LSMOP8 | 3 | 500 | 5.6083e-1 (1.38e-4) | 5.6082e-1 (1.32e-4) | 5.6102e-1 (5.79e-5) |
| | | 1000 | 5.4347e-1 (1.98e-4) | 5.4302e-1 (1.74e-4) | 5.4370e-1 (1.13e-4) |
| LSMOP9 | 3 | 500 | 1.0002e+2 (2.00e+0) | 1.2358e+2 (2.24e+0) | 1.1301e+2 (8.09e+0) |
| | | 1000 | 1.1115e+2 (4.73e+0) | 1.3329e+2 (1.70e+0) | 1.2461e+2 (2.39e+0) |
| + /=/= | | | 0/0/4 | 0/0/4 | |

6.4 在真实问题上对文章算法和对比算法进行比较

真实问题和测试问题在研究中扮演着不同角色。测试问题, 也称为基准问题, 是人为设计的、具有已知特性的标准问题, 它们主要用于算法开发初期的快速验证和比较不同算法的性能。而真实问题则直接来源于现实世界的应用场景, 它们通常更加复杂、不确定, 并且包含噪声等实际数据问题。尽管算法在测试问题上表现出色, 但在真实问题上的实验是必要的, 因为这些实验能够验证算法在处理现实世界复杂性时的有效性和鲁棒性, 同时展示算法解决

实际问题的能力。此外，真实问题的实验结果对于推动算法的进一步发展和优化具有重要意义，因为它们提供了关于算法在实际应用中可能遇到的挑战和限制的宝贵信息。为此，实验进一步在真实问题 TREE 上对 AVG-SAEA 进行测试。前面的实验中的测试问题首先采用了 IGD+ 作为评估指标，这是因为 IGD+ 能够有效地衡量算法产生的解集与已知真实帕累托前沿之间的距离，这一指标在测试问题中特别有用，因为这些测试问题通常具有已知的帕累托前沿，使得 IGD+ 的计算成为可能。然而，在真实问题的实验中，由于难以获得真实的帕累托前沿数据，IGD+ 的计算变得不切实际。因此，实验转而使用 HV（超体积）指标，它不需要真实的帕累托前沿信息，而是通过计算解集在目标空间中覆盖的体积来评估算法性能，这使得 HV 成为评估真实问题中算法性能的一个实用且有效的指标。总结来说，IGD+ 和 HV 的不同在于它们适用的场景和所需信息的不同：IGD+ 适用于测试问题，需要真实帕累托前沿进行精确的收敛性评估；而 HV 适用于真实问题，不需要真实帕累托前沿，侧重于评估解集的多样性和覆盖范围。本地复现结果如表 4 所示。

表 4. AVG-SAEA 与对比算法在真实问题上得到的 IGD+ 值表

| Problem | D | K-RVEA | CSEA | EDNARMOEA | AVG-SAEA |
|---------|-----|--------------------|--------------------|--------------------|--------------------|
| TREE1 | 120 | 1.0129e-4(4.08e-4) | 1.6660e-2(1.85e-2) | 0.0000e+0(0.00e+0) | 5.1185e-1(1.28e-1) |
| TREE2 | 120 | 1.0229e-1(1.44e-2) | 1.2853e-1(1.67e-2) | 1.0458e-1(8.04e-3) | 5.1294e-1(1.48e-1) |
| +/-/= | | 0/6/0 | 0/6/0 | 0/6/0 | |

7 总结与展望

7.1 个人收获

首先是对高维昂贵多目标优化问题及其研究方向有了较为清晰的认识。昂贵就是计算适应度的过程复杂，所以就涉及到代理模型和填充准则，通过它们来减少昂贵的真实评估次数而又不影响效果。多目标会导致个体之间不便进行比较，所以就是解决办法是要寻找比较的指标，比如非支配排序等。高维会引起搜索空间爆炸，进而导致算法收敛速度慢、代理模型训练慢等问题，所以解决办法就是要么通过降维从根本上剔除，要么是设计更优质的代理模型减少评估解的时间消耗。其次是提高了自身的 matlab 编程、运行、调试以及收集数据、整理数据并将数据可视化为图像的能力。在复现文章算法的时候，基于自身的理解并结合基本的进化算法框架，不断将涉及高维昂贵多目标的代码加入框架中，这个过程中会出现很多报错，需要经常调试纠错。等到算法能正常运行后，需要将算法进行实现，去解决测试问题或者真实问题。为了体现算法的效果，需要经常收集算法优化后最终得到的种群，用以计算最终的种群的目标值集合与真实帕累托前沿的距离，通过这个距离衡量算法的效果。此外，还需要对收集的种群数据进行整理，通过图标的方式更加直观地加以展示。此外，通过课程实验了解了一般实验的过程，总体上就分为两步。首先是要验证所提方法整体的有效性，这通过将方法与现有的先进对比算法进行比较来体现。第二步是通过消融实验来验证算法中的创新点的有效性，说明算法整体性能好确实是因为创新工作的作用。当然不同工作解决问题时针对的点不一样，比如课程实验中 AVG-SAEA 主要是解决了高维引起的问题，那么在实验中还可以从决策变量维度以及时间消耗方面去更进步一说明算法的有效性。除了上述这些，其实还

有很多小的知识点，比如对 DTLZ、WFG、LSMOP、TREE 等特点有了了解，也知道了 IGD+、HV、Wilcoxon rank-sum test 等指标的作用和含义。总之，这次课程实验是我第一次完整地走完整个科研的流程，收获很多，对我今后的科研肯定有所帮助。

7.2 课程复现不足

对于课程实验中涉及的对比算法，我并没有去看原文章以及复现等，所以我对这些对比算法并不熟悉。这就导致了在对比实验中我并不能清楚地知道为什么在某些条件下 AVG-SAEA 的效果会不如对比算法。未来，我会逐步去看懂这些对比算法，但这是一个稍微较费时间的过程，因为这相当于好几个课程实验的工作量。此外，自己复现的算法在实验中表现的效果并没有达到和原文一样好的水平，这一点我也还没有找到具体原因，更加麻烦的是我对此还没有一个如何去找错去优化去修正的想法，这应该是由我自身经验不足导致的。

参考文献

- [1] Ye Tian, Langchun Si, Xingyi Zhang, Ran Cheng, Cheng He, Kay Chen Tan, and Yaochu Jin. Evolutionary large-scale multi-objective optimization: A survey. *ACM Comput. Surv.*, 54(8), October 2021.
- [2] Lluvia M Ochoa-Estopier, Víctor M Enríquez-Gutierrez, Lu Chen, José M Fernández-Ortiz, Lorenzo Herrero-Soriano, and Megan Jobson. Industrial application of surrogate models to optimize crude oil distillation units. *Chemical Engineering Transactions*, 69:289–294, 2018.
- [3] Yaochu Jin, Handing Wang, Tinkle Chugh, Dan Guo, and Kaisa Miettinen. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23(3):442–458, 2018.
- [4] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.
- [5] Jianqing Lin, Cheng He, and Ran Cheng. Adaptive dropout for high-dimensional expensive multiobjective optimization. *Complex & Intelligent Systems*, 8(1):271–285, 2022.
- [6] Dan Guo, Xilu Wang, Kailai Gao, Yaochu Jin, Jinliang Ding, and Tianyou Chai. Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network. *IEEE transactions on systems, man, and cybernetics: systems*, 52(4):2084–2097, 2021.
- [7] Xin Zhang, Bo-Wen Ding, Xin-Xin Xu, Jian-Yu Li, Zhi-Hui Zhan, Pengjiang Qian, Wei Fang, Kuei-Kuei Lai, and Jun Zhang. Graph-based deep decomposition for overlapping large-scale optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(4):2374–2386, 2022.

- [8] Cheng He, Ran Cheng, Lianghao Li, Kay Chen Tan, and Yaochu Jin. Large-scale multi-objective optimization via reformulated decision variable analysis. *IEEE transactions on evolutionary computation*, 28(1):47–61, 2022.
- [9] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, 10(3):263–282, 2002.
- [10] Yingwei Li, Xiang Feng, and Huiqun Yu. Solving high-dimensional expensive multiobjective optimization problems by adaptive decision variable grouping. *IEEE Transactions on Evolutionary Computation*, 2024.
- [11] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4):73–87, 2017.