

中文图书分类号: TP18  
密 级: 公开  
UDC: 004  
学 校 代 码: 10005



# 硕 士 学 位 论 文

MASTERAL DISSERTATION

论 文 题 目: 面向大规模多目标优化的决策  
变量分组策略研究

论 文 作 者: 王佩科

学 科: 计算机科学与技术

指 导 教 师: 杨翠翠 副教授

论文提交日期: 2023 年 5 月



UDC: 004  
中文图书分类号: TP18

学校代码: 10005  
学 号: S202074008

# 北京工业大学工学硕士学位论文

题 目: 面向大规模多目标优化的决策变量分组策略研究

英文题目: RESEARCH ON LARGE-SCALE MULTI-  
OBJECTIVE EVOLUTIONARY ALGO-  
RITHM BASED ON DECISION VARIABLE  
GROUPING STRATEGY

论 文 作 者: 王佩科  
学 科 专 业: 计算机科学与技术  
研 究 方 向: 计算机应用技术  
申 请 学 位: 工学硕士  
指 导 教 师: 杨翠翠 副教授  
所 在 单 位: 计算机学院  
答 辩 日 期: 2023 年 5 月  
授 予 学 位 单 位: 北京工业大学



## 独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的  
研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他  
人已经发表或撰写过的研究成果，也不包含为获得北京工业大学或其它教育机构  
的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均  
已在论文中作了明确的说明并表示了谢意。

签 名：\_\_\_\_\_

日 期：\_\_\_\_\_ 年 月 日

## 关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有权  
保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部  
分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后应遵守此规定）

签 名：\_\_\_\_\_

日 期：\_\_\_\_\_ 年 月 日

导师签名：\_\_\_\_\_

日 期：\_\_\_\_\_ 年 月 日



## 摘 要

大规模多目标优化问题广泛存在于现实应用和科学研究中，是一类需要同时优化多个目标且具有大量决策变量的复杂优化问题。相较于一般的多目标优化问题，该类问题由于含有大规模的决策变量，导致搜索空间庞大且高度复杂，比一般的多目标优化问题更难求解。许多研究已经表明，当前的多数多目标进化算法在有限的计算资源下无法有效解决该类问题，而通过对决策变量进行分组，将大规模多目标优化问题转化为一系列较小规模的多目标优化问题来分而治之地求解是一种有效的解决方法。目前，常见的决策变量分组策略包括决策变量随机分组和决策变量分析分组两种。然而，现有的决策变量随机分组策略可能使相互关联的决策变量分入不同组，影响求解性能；而决策变量分析分组策略由于需要消耗大量的函数评价来分析决策变量之间的关联性，严重影响了计算效率。为在一定程度上克服分组策略的上述问题，本课题开展了如下两项研究工作：

(1) 针对决策变量随机分组策略会使相互关联的变量分入不同组从而影响求解性能的问题，本研究提出了一种面向大规模多目标优化问题的双分解分组策略 (Dual Decomposition Grouping Strategy for Large-scale Multi-objective Optimization Problems, DDS)。具体来说，首先，采用滑动窗口技术将决策变量划分为相互重叠的组以缓解相互关联的变量被划分入不同组的问题，同时将大规模多目标优化问题转换为一系列小规模多目标优化问题。然后，根据全局参考点对小规模多目标优化问题进一步分解，将小规模多目标优化问题转换为一组单目标优化问题以提高求解效率。最后，利用差分进化算法对每个单目标优化问题进行优化，并使用块坐标下降策略维护原始问题解的完整性。在基准测试问题集和实际应用问题上的实验结果表明：与多个先进算法对比，嵌入了 DDS 的算法在大多数问题上表现优越，能够得到性能更好的帕累托最优解集。

(2) 针对决策变量分析策略计算效率低下的问题，本研究提出了一种面向大规模多目标优化问题的保守序列分组策略 (Conserved Sequence Grouping Strategy for Large-scale Multi-objective Optimization Problems, CS)。首先，通过分析决策变量在进化过程中的历史行为，将决策变量划分为保守序列和非保守序列，其中保守序列被定义为连续两次更新变化差异较小的决策变量；非保守序列被定义为连续两次的更新差异较大的决策变量。这种划分方式无需进行函数评价，从而提高决策变量分析的效率。然后，对保守序列和非保守序列分别采取特定的进化

策略。保守序列根据其父代的历史更新行为进行优化，可以充分利用种群进化过程中的历史信息来提高算法性能。非保守序列则按照常规的进化过程进行优化。在基准测试集上的实验结果表明：与多个先进算法相比，嵌入了 CS 的算法能够快速地的得到性能更好的帕累托最优解。

上述两个策略可以嵌入到现有的多目标进化算法中，以提高其求解大规模多目标优化问题的性能。因此，本研究丰富了大规模多目标优化进化算法的理论框架，为大规模多目标优化问题求解提供了新的思路，具有重要的理论意义和应用价值。

**关键词：**大规模多目标优化；进化算法；决策变量分组；滑动窗口；保守序列



## Abstract

Large-scale multi-objective optimization problems are widely found in real-world applications and scientific research, which involve simultaneously optimizing multiple objectives and coping with a large number of decision variables. In comparison to general multi-objective optimization problems, this type of problem is more difficult to solve due to the vast and highly complex search space resulting from the large-scale decision variables. Many studies have shown that most existing multi-objective evolutionary algorithms struggle to effectively solve these problems under limited computational resources. An effective approach is to divide the large-scale multi-objective optimization problem into smaller ones by grouping decision variables.

There are two common strategies for decision variable grouping: random grouping and variable analysis grouping. However, existing random grouping strategies may separate correlated decision variables, affecting solution performance, while variable analysis grouping strategies consume a significant amount of function evaluations to analyze correlations between variables, severely impacting computational efficiency. To address these issues, this project conducts the following two research efforts:

(1) To tackle the issue that random grouping strategies may separate correlated variables, we propose a Dual Decomposition Grouping Strategy for Large-scale Multi-objective Optimization Problems (DDS). Firstly, sliding window technique is employed to partition decision variables into overlapping groups, mitigating the problem of correlated variables being separated and transforming the large-scale problem into several smaller ones. Then, the small-scale multi-objective optimization problems are further decomposed according to global reference points, converting them into single-objective problems to improve solution efficiency. Finally, differential evolution algorithm is used to optimize each single-objective problem, and block coordinate descent strategy is employed to maintain the integrity of the original problem's solutions. Experimental results on benchmark test sets and real-world applications demonstrate that algorithms embedded with DDS outperform several advanced algorithms, achieving better Pareto-optimal solution sets.

(2) To address the low computational efficiency issue of variable analysis strategies, we propose a Conserved Sequence Grouping Strategy for Large-scale Multi-objective Optimization Problems (CS). Firstly, by analyzing the historical behavior of decision variables during the evolutionary process, they are divided into conserved sequences and non-conserved sequences. Conserved sequences are defined as decision variables with small differences between consecutive updates, while non-conserved sequences have large differences between consecutive updates. This partitioning method does not

require function evaluations, thus improving the efficiency of variable analysis. Then, specific evolutionary strategies are applied to the conserved and non-conserved sequences separately. Non-conserved sequences are optimized according to their parents' historical update behavior, utilizing historical information from the population's evolution process to enhance algorithm performance. Conserved sequences are optimized following the standard evolutionary process. Experimental results on benchmark test sets show that algorithms embedded with CS can quickly obtain better Pareto-optimal solutions compared to several advanced algorithms.

These two strategies can be integrated into existing multi-objective evolutionary algorithms to improve their performance in solving large-scale multi-objective optimization problems. Therefore, this research enriches the theoretical framework of large-scale multi-objective optimization evolutionary algorithms and provides new insights for solving large-scale multi-objective optimization problems, carrying significant theoretical and practical value.

**Keywords:** Large-scale multi-objective optimization, evolutionary algorithms, decision variables decomposition, sliding window, conserved sequence

## 目 录

摘 要 .....	I
Abstract .....	III
第 1 章 绪论 .....	1
1.1 研究背景与意义 .....	1
1.2 国内外研究现状 .....	2
1.3 论文的主要研究内容 .....	3
1.4 论文的组织结构 .....	4
第 2 章 相关知识介绍 .....	7
2.1 大规模多目标优化问题简介 .....	7
2.1.1 大规模多目标优化的基本概念 .....	7
2.1.2 大规模多目标优化的评价指标 .....	8
2.2 决策变量分组策略简介 .....	8
2.2.1 决策变量随机分组策略 .....	9
2.2.2 决策变量分析分组策略 .....	10
2.3 本章小结 .....	13
第 3 章 面向大规模多目标优化问题的双分解分组策略 .....	15
3.1 双分解分组策略思想 .....	15
3.2 双分解分组策略描述 .....	17
3.2.1 外分解操作 .....	17
3.2.2 内分解操作 .....	20
3.2.3 基于块坐标下降的差分进化优化策略 .....	20
3.3 时间复杂度分析 .....	22
3.4 收敛性讨论 .....	22
3.5 实验结果及分析 .....	23
3.5.1 实验设置 .....	24
3.5.2 参数敏感性分析 .....	25
3.5.3 双分解分组策略的有效性验证 .....	27
3.5.4 双分解分组策略的效率验证 .....	28
3.5.5 与其它算法在基准测试问题上的比较 .....	29

3.5.6 与其它算法在实际应用问题上的比较 .....	32
3.6 本章小结 .....	35
第 4 章 面向大规模多目标优化问题的保守序列分组策略 .....	37
4.1 保守序列分组策略思想 .....	37
4.2 保守序列分组策略描述 .....	38
4.2.1 保守序列分析 .....	38
4.2.2 特异进化策略 .....	41
4.3 时间复杂度分析 .....	42
4.4 实验结果及分析 .....	43
4.4.1 实验设置 .....	43
4.4.2 参数敏感性分析 .....	44
4.4.3 保守序列分组策略的有效性验证 .....	45
4.4.4 保守序列分组策略的效率验证 .....	45
4.4.5 与其它算法在基准测试问题上的比较 .....	47
4.5 本章小结 .....	51
结 论 .....	53
攻读硕士学位期间索取得的主要科研成果 .....	55
参考文献 .....	55
致 谢 .....	57

## 第 1 章 绪论

### 1.1 研究背景与意义

多目标优化问题<sup>[2]</sup> (Multiobjective Optimization Problems, MOPs) 是一类广泛存在于实际应用和科学研究的复杂优化问题。该类问题不存在唯一的最优解, 其求解目标是找到一组具有良好收敛性和多样性的权衡解, 称为帕累托最优解 (Pareto Optimal Solutions) 或非支配解 (Non-dominated Solutions)。所有帕累托最优解对应的目标值组成的曲面被称为帕累托前沿面 (Pareto Front, PF)。

进化算法 (Evolutionary Algorithms, EAs) 由于运行一次可以同时找到多个帕累托最优解, 已经成为了当今求解多目标优化问题的主流方法。解决多目标问题的进化算法通常被称为多目标进化算法 (Multi-objective Evolutionary Algorithms, MOEAs)<sup>[2]</sup>。在过去数十年里, 研究人员提出了多种多样的 MOEAs, 这些 MOEAs 大致可以分为: 基于帕累托的 MOEAs<sup>[2]</sup>、基于指标的 MOEAs<sup>[2]</sup> 和基于分解的 MOEAs<sup>[2]</sup>。客观来讲, 目前多数 MOEAs 只适合求解决策变量数量较少的 MOPs。

然而, 随着生产场景日益复杂, 实际应用中的 MOPs 往往呈现出大规模的特点, 如大规模调度优化问题需要同时对成百上千个节点进行调控管理, 通过寻找最佳的调度方案, 来实现最小的成本或最大的利润<sup>[2]</sup>; 计算机视觉领域的许多应用需要处理具有高分辨率像素的图片数据, 以实现准确的图像分析和识别<sup>[2]</sup>; 大型神经网络的训练需要同时优化众多参数, 以实现准确和智能的决策<sup>[2]</sup>; 推荐系统为了保证推荐的准确性, 需要同时对高维特征进行分析处理<sup>[2]</sup>等等。许多研究已经表明, 目前已有的多数进化算法不能有效解决具有大规模决策变量的多目标优化问题。为此大规模多目标优化问题 (Large-scale Multi-objective Optimization Problems, LSMOPs) 近年来受到了研究人员的广泛关注。通常情况下, 将超过 100 个决策变量的 MOPs 被称为 LSMOPs。随着决策变量的增多, 决策空间呈指数级扩张并极具复杂性, 大规模多目标优化问题的求解难度会随之快速增大。

为了提高 MOEAs 在解决大规模多目标优化问题的性能, 研究人员近年来已经提出不少策略<sup>[2]</sup>。其中, 决策变量分组策略<sup>[2]</sup>通过对决策变量进行分组, 将原始问题转换为一系列较小规模的 MOPs, 从而降低问题复杂性并提升求解效率, 提高解的质量, 是一种求解 LSMOPs 的有效策略。决策变量随机分组和决策变量分析分组是常见的分组策略, 然而, 现有的决策变量随机分组策略会使相互关联的决策变量被分为不同的组, 影响求解性能; 而决策变量分析分组策略需要

消耗大量的函数评价进行分析,效率低下。为了在一定程度上克服这些问题,本课题针对决策变量分组中上述问题进行了深入研究,提出了两种新的分组策略,为大规模多目标优化问题求解及其应用提供了新的思路,具有一定的理论意义和应用价值。

## 1.2 国内外研究现状

由于现有的 MOEAs 不能有效地对巨大且复杂的决策空间进行探索,因而不能有效地求解大规模多目标优化问题。为了解决该问题,近年来研究人员提出了三种不同的策略:决策变量分组策略、决策空间约简策略和整体搜索策略,下面将分别对其进行概述。

### (1) 决策变量分组策略

该类策略的求解思路为:首先将所有的决策变量划分为若干较小规模的决策变量组,这些决策变量组对应着较小的决策空间和较小规模的多目标优化问题;然后利用多目标进化算法在这些较小的决策空间中搜索相应小规模多目标优化问题的帕累托最优解,将每个较小规模的多目标优化问题的帕累托最优解进行拼接得到原始问题的帕累托最优解,从而降低大规模多目标优化问题的复杂度,提高求解性能。例如,文献[?]在 2013 年首次提出了决策变量随机分组策略,它将决策变量随机分成若干大小相等的组,并利用 GDE3<sup>[2]</sup>算法依次进行优化。该研究初步证明了决策变量分组策略求解 LSMOPs 的潜力。文献[?]为了解决固定大小的分组导致的缺乏鲁棒性问题,在进化过程中加入了动态分组,进一步提高了进化算法求解 LSMOPs 的能力。客观而言,随机决策变量分组策略虽然效率较高,但现有的随机分组策略会把相互关联的决策变量被划分到不同组,这可能会使种群趋于决策空间的局部最优区域。为了应对这一问题,有研究者提出通过对决策变量进行分析来进行分组的方法。例如,文献[?]通过对决策变量添加扰动,根据扰动前后的支配关系,将决策变量分为位置变量、混合变量和距离变量。文献[?]通过对决策变量进行聚类分析,将决策变量分为收敛变量和多样性变量。文献[?]和文献[?]提出了可区分分析分组,它根据不同决策变量对应的目标变量的大小关系来识别决策变量之间的相互作用。

### (2) 决策空间约简策略

该类策略的求解思路为通过降低决策空间维度来降低 LSMOPs 的求解难度。决策空间约简后,进化算法无需在高维的决策空间中寻找原始问题的解,而是在降维后的空间中进行搜索,得到低维空间的帕累托最优解后再将其映射回原始

问题的帕累托最优解。例如,文献[?]提出了 WOF 算法,它使用了线性加权的空间约简策略,通过权重向量分配将原始问题中高维决策变量的优化转化为更低维度的权重向量的优化,并通过决策变量和权重向量的线性组合得到原始问题的解。文献[?]分配了两个不同方向的权重向量来降低决策空间的维度,并通过优化权重向量最大化超体积<sup>[2]</sup>指标得到原始问题的解。文献[?]借助了文献[?]的聚类方法对决策变量进行分组,然后通过主成分分析<sup>[2]</sup>对原始决策空间进行约减降维,算法在降维后的空间进行搜索,并将搜索结果映射为原始问题的解,降低了原始问题的求解难度。文献[?]将解表示为二进制向量和实值向量,并使用二进制向量训练受限 Boltzman 机<sup>[2]</sup>,使用实值向量训练自编码器<sup>[2]</sup>,分别对两种向量进行降维,以达到对决策空间约简,降低原始 LSMOPs 求解难度的目的。

### (3) 整体搜索策略

上述两类策略是基于不同的思想将难求解的原始 LSMOPs 转化为易求解的较小规模的问题来求解。而整体搜索策略的基本求解思路为通过设计高效的搜索算子来对原始 LSMOPs 的搜索空间直接进行探索。例如,文献[?]提出 LMOCSSO,它针对粒子群算法设计了一种基于竞争优化的复制算子,并提出一种同时考虑速度和加速度的更新策略来加快收敛速度,能够高效地搜索原始决策空间。文献[?]引入自适应后代生成策略,可以在不同种群中沿支配和非支配解之间的方向快速生成解,从而提高算法的搜索效率。

综上所述,尽管研究者们已经为进化算法求解大规模多目标优化问题提供了不少策略,但是,现有策略仍有很多不足,有待进一步深入研究。具体来说,(1)决策变量分组策略会使得相互关联的决策变量被划分为不同组,导致算法可能陷入局部最优区域。决策变量分析策略的分析过程会消耗大量计算资源,导致了求解效率极低,限制了实际应用场景。(2)决策空间约简策略的效果与具体的空间变化方法有关,随着决策变量数量的增加,找到可逆且信息损失少的空间变换方法非常困难。(3)整体优化策略虽然可以直接对 LSMOPs 进行求解,但是,用于整体优化的搜索算子对决策变量的数量比较敏感,很难发现适用于一切情况的搜索算子。

## 1.3 论文的主要研究内容

本课题主要针对决策变量分组策略存在的问题进行了一些深入的探索,进行了如下两个方面的研究:

第一, 针对随机决策变量分组会使相互关联的决策变量被划分为不同组的问题, 提出了面向大规模多目标优化问题的双分解分组策略 (DDS)。该策略使用滑动窗口将决策变量划分为具有重叠部分的组, 根据全局参考点和全局参考向量进一步对决策变量组对应的 MOPs 分解以提高解的多样性。具体而言, 首先, 使用滑动窗口对决策变量分组, 将决策变量划分为相互重叠的组, 每个决策变量组对应一个小规模的 MOPs, 相互重叠的划分可以缓解随机分组将相互关联的决策变量划分为不同组的问题, 然后, 根据全局参考点和全局参考向量对小规模 MOPs 进一步分解, 保证解的多样性。最后, 使用块坐标下降的思想维护完整的解, 进一步保证解的质量。实验结果显示, 嵌入了 DDS 的算法在与其他先进大规模多目标进化算法比较时展现出令人满意的性能。

第二, 针对决策变量分析分组策略效率低下的问题, 提出了面向大规模多目标优化问题的保守序列分组策略 (CS)。该策略将自然进化中的保守序列概念引入 MOEAs。具体而言, 首先, 该策略将进化过程分为保守序列分析阶段和特异进化阶段。在保守序列分析阶段, 算法根据种群朝着帕累托最优解更新的方向变化的差异大小, 将决策变量划分为保守序列和非保守序列, 这种划分方式不用消耗函数评价, 因此具有较高的效率。然后, 在特异进化阶段, 根据保守序列和非保守序列的不同特点, 针对性地采取不同的优化更新策略。实验表明, 与其他先进的大规模多目标 MOEAs 相比, 嵌入了 CS 的算法能够高效地得到性能比较好的帕累托最优解。

## 1.4 论文的组织结构

本文文共分为 4 章, 具体内容如下:

第 1 章: 绪论。首先, 介绍了大规模多目标优化问题的研究背景和意义; 然后, 概括了一般大规模多目标进化算法的研究现状; 最后, 总结了论文的主要工作。

第 2 章: 相关知识介绍。首先, 对大规模多目标优化的基本概念和评价指标进行了介绍; 然后, 介绍了一些求解 LSMOPs 的经典的决策变量分组策略。

第 3 章: 提出面向大规模多目标优化问题的双分解分组策略。首先, 介绍了该策略的整体框架; 然后, 依次给出了双分解操作和基于块坐标下降的差分进化优化策略的具体实现; 随后, 对算法的收敛性和时间复杂度进行了分析。最后, 将 DDS 嵌入 MOEA 中并与多种优秀大规模多目标进化算法的实验比较表明, 嵌入了 DDS 的算法获得的非支配解有更好的收敛性和多样性。



第 4 章：提出了面向大规模多目标优化问题的保守序列分组策略。首先，介绍了该策略的整体框架；然后，依次给出了保守序列分析、个体的进化更新策略的具体实现；随后，对算法的时间复杂度进行了分析。最后，将 CS 嵌入 MOEA 并与多个大规模多目标进化算法的实验比较说明了 CS 能够更快地对决策变量分析分组，并且嵌入了 CS 的算法能够获得具有较高质量的帕累托最优解。

结论：总结了本文的主要工作和创新点，并对后续的研究工作进行了展望。



## 第2章 相关知识介绍

### 2.1 大规模多目标优化问题简介

在复杂生产和科学研究领域中经常会出现决策变量数量常达到几百乃至上千个的大规模多目标优化问题。该类问题求解的核心目标在于高效地为决策者提供的解决方案，从而满足各种生产生活和科学研究需求。本节将概述大规模多目标优化问题的基本概念以及相关评价指标。

#### 2.1.1 大规模多目标优化的基本概念

不失一般性，一个具有  $n$  个决策变量， $m$  个目标函数的 MOP 数学模型可以表示为：

$$\mathbf{Min}_{\mathbf{x} \in \Omega} F(\mathbf{x}) = f_1(\mathbf{x}), f_2(\mathbf{x}) \cdots, f_M(\mathbf{x}) \quad (2-1)$$

其中， $\mathbf{x} = (x_1, x_2, \cdots, x_D)$  是决策空间  $\Omega$  中的  $D$  维决策向量； $f_1, f_2, \dots, f_M$  是  $M$  个优化目标， $F: \Omega^D \mapsto \mathbb{R}^M$  是从  $D$  维决策空间  $\Omega$  到  $M$  维目标空间  $\mathbb{R}^M$  的函数。特别地，当  $D > 100$  时，MOP 通常被称为 LSMOP。

**定义 2-1 (Pareto 支配)：**对任意  $\mathbf{x}, \mathbf{y} \in \Omega$ ，称  $\mathbf{x}$  Pareto 支配  $\mathbf{y}$ （也称为  $\mathbf{x}$  支配  $\mathbf{y}$ ），当且仅当：

$$\begin{aligned} \forall i = 1, 2, \dots, M, f_i(\mathbf{x}) &\leq f_i(\mathbf{y}) \wedge \\ \exists j = 1, 2, \dots, M, f_j(\mathbf{x}) &< f_j(\mathbf{y}) \end{aligned} \quad (2-2)$$

记作  $\mathbf{x} > \mathbf{y}$ 。

**定义 2-2 (Pareto 最优解)：** $\mathbf{x}$  被称为 Pareto 最优解（或非支配解），当且仅当决策空间中不存在能够支配  $\mathbf{x}$  的解  $\mathbf{y}$ ，即：

$$\neg \exists \mathbf{y} \in \Omega, \mathbf{y} > \mathbf{x} \quad (2-3)$$

**定义 2-3 (Pareto 最优解集)：**所有 Pareto 解组成的集合称为 Pareto 最优解集，即：

$$PS = \{\mathbf{x} | \neg \exists \mathbf{y} \forall \mathbf{x} \in \Omega, \mathbf{y} > \mathbf{x}\} \quad (2-4)$$

**定义 2-4 (Pareto 前沿)：**所有 Pareto 最优解所应的目标向量组成的集合称为

Pareto 前沿，即：

$$PF = \{F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) \mid \mathbf{x} \in PS\} \quad (2-5)$$

### 2.1.2 大规模多目标优化的评价指标

本节将介绍如下两种在 LSMOPs 中常用的评价指标。

#### (1) 反世代距离

反世代距离 (Inverted Generational Distance, IGD) 衡量算法在决策空间中所获得的近似帕累托最优解与真实帕累托最优解之间的平均欧氏距离，是一种同时评价算法收敛性和多样性的综合指标，其计算公式为：

$$IGD(S, S^*) = \frac{1}{|S^*|} \sum_{\mathbf{x}^* \in S^*} d(\mathbf{x}^*, S) \quad (2-6)$$

其中， $S$  表示算法求得的一组帕累托最优解； $S$  表示沿 PS 的一组均匀分布解； $d(\mathbf{x}^*, S)$  是  $\mathbf{x}^*$  与其在  $S$  中最近的点之间的欧几里得距离。通常，较小的  $IGD(S, S^*)$  表示所得到的解集  $S$  的收敛性和在解空间中分布的均匀性更好。

#### (2) HV 超体积

超体积指标 (Hypervolume, HV) 反映了算法求得的一组帕累托最优解集对应的目标向量与参考点在目标空间中所围成区域的体积，也是一个同时评价算法收敛性和多样性的指标。HV 值越大，说明算法搜索到的帕累托最优解具有更好的收敛性和多样性，其计算公式为：

$$HV = \delta \left( \bigcup_{i=1}^{|S|} v_i \right) \quad (2-7)$$

其中， $\delta$  表示勒贝格测度；用来测量目标向量和参考点在目标空间中的超体积； $|S|$  表示非支配解集的数目； $v_i$  表示参照点与解集中第  $i$  个解对应的目标向量构成的超立方体。

## 2.2 决策变量分组策略简介

决策变量分组策略是求解 LSMOPs 的有效手段。该策略通过分治原则，将决策变量划分为数组，并对各组决策变量交替进行优化。鉴于 LSMOPs 中决策变量间存在复杂的相互作用关系，分组的质量直接影响算法最终所获得的解的质量。因此，对决策变量合理进行分组关系着求解性能。目前常见的分组策略

**算法 2-1: 决策变量随机分组策略框架**

输入: 种群大小 ( $N$ ), 最大迭代次数 ( $G$ ), 决策变量维度 ( $D$ ), 分组大小 ( $R$ )

输出: 最终种群  $P$

```

1:  $P \leftarrow$  初始化种群 ( $N, D$ )
2:  $G_c \leftarrow 1$ 
3: while  $G_c \leq G$  do
4:    $Group \leftarrow$  将  $P$  中的决策变量按照分组大小  $R$  随机划分为互不相交的子集
5:   for  $Group$  中的每个决策变量组  $g$  do
6:     使用选定的进化算法 MOEA 求解  $g$ 
7:   end for
8:    $P \leftarrow$  使用非支配排序选择当前优秀个体
9:    $G_c \leftarrow G_c + 1$ 
10: end while
11: return  $P \leftarrow$  选择  $P$  中的非支配解

```

包括决策变量随机分组策略和决策变量分析分组策略。

**2.2.1 决策变量随机分组策略**

决策变量随机分组(Random Grouping)策略是一种简单易实现的求解 LSMOPs 的分组策略。该策略通常预先设定每组决策变量的数量,然后随机地为每个分组确定决策变量。杨振宇等人<sup>[2]</sup>最早提出了基于决策变量随机分组的协同进化框架。如算法2-1所示,首先进行初始化操作(第1-2行)。随后开始迭代过程(第3-10行),首先,将决策变量随机划分为预设长度的决策变量组(第4行)。然后,利用选定的进化算法将对这些决策变量组依次优化(第5-7行)。当所有决策变量组被优化完成后,使用非支配排序<sup>[2]</sup>选择当前的优秀个体作为新种群(第8行)。重复上述过程直到迭代完毕,最后输出优化后的种群  $P$ (第11行)。在随机分组策略中,分组大小对算法性能有着很大的影响。因而上述采用固定值作为分组大小很难适应各种类型的 LSMOPs。因此,文献[?]提出了动态分组策略,其过程为:预先设置一个包含不同分组大小的集合,在每一次迭代中,使用轮盘赌选择方法从集合中选择分组大小,根据所选择的分组大小将决策变量分组并优化。对算法性能具有更大提升的分组大小将会被以更高的概率选择。

尽管随机分组策略提供了一种简单可行的求解 LSMOPs 的思路,但随机分组策略没能考虑相互关联的决策变量被划分为不同组的问题。因此,随机分组策略难以决策变量有较多关联的 LSMOPs。

---

**算法 2-2: 控制性质分析策略**


---

输入: 决策变量维度 ( $D$ ), 采样数量  $NCA$ , 待分析决策向量  $\mathbf{x}$

输出: 控制性质分析结果

```

1:  $i, j \leftarrow 0$ 
2:  $S \leftarrow$  初始化采样集
3: while  $i \leq D$  do
4:   在可行域中生成随机向量  $r = (r_1, \dots, r_i, \dots, r_D)$ 
5:   while  $j \leq NCA$  do
6:      $x'_i \leftarrow x_i^L + \frac{j-1+\text{rand}}{NCA}(x_i^U - x_i^L)$ , 其中  $\text{rand}$  为区间  $[0, 1)$  中的随机数;  $x_i^L$  和  $x_i^U$  分别为  $\mathbf{x}_i$ 
       第  $i$  个决策变量的下界和上界
7:      $f \leftarrow$  计算  $(x_1, \dots, x_{i-1}, x'_i, x_{i+1}, \dots, x_n)$  的目标变量, 其中  $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \in \mathbf{x}$ 
8:      $j \leftarrow j + 1$ 
9:     将  $f$  添加至采样集
10:  end while
11:  对  $S$  使用非支配排序得到不同的非支配前沿
12:  if 第一个非支配前沿面中的个体数量  $\geq NCA$  then
13:     $x_i$  为位置变量
14:  else if 所有非支配前沿面的个体数量  $= 1$  then
15:     $x_i$  为距离变量
16:  else
17:     $x_i$  为混合变量
18:  end if
19:   $i \leftarrow i + 1$ 
20: end while

```

---

### 2.2.2 决策变量分析分组策略

决策变量分析 (Decision Variable Analysis) 分组策略通过分析决策变量与目标变量之间的关系或对目标变量产生的影响来确定相互关联的决策变量, 将相互关联的决策划分为同一组。该策略的基本思路为: 首先, 设定分析标准, 采取遍历或随机抽样的方式对决策变量施加扰动, 观察目标变量呈现的变化或特性。然后, 根据目标变量的变化确定相互关联的变量, 并将其分为一组。决策变量分析分组策略包括三种代表性的具体策略: 可区分分析、控制性质分析和角度聚类分析。

#### (1) 可区分分析策略

可区分分析策略<sup>[2]</sup>通过比较不同决策变量对应的目标变量之间的大小关系, 区分具有相互作用的决策变量并将其划分为同一组。该可以缓解随机分组策略中具有相互作用的决策变量被分配到不同组的问题, 同时也可避免随机分组策略需要预先设定分组大小的问题。可区分分析的关键点在于识别决策变量之间

的相互作用。对于该策略，需要首先给出决策变量相互作用的数学定义：对于可微函数  $f(\cdot)$  和两个决策变量  $x_i, x_j$ ，若  $x_i, x_j$  相互作用，当且仅当存在四个变量值  $a_1, a_2, b_1, b_2$  使下式成立：

$$\begin{cases} f(\mathbf{x})|_{x_i=a_1, x_j=b_1} > f(\mathbf{x})|_{x_i=a_2, x_j=b_1} \\ f(\mathbf{x})|_{x_i=a_1, x_j=b_2} < f(\mathbf{x})|_{x_i=a_2, x_j=b_2} \end{cases} \quad (2-8)$$

根据公式（2-8）可知， $f(\mathbf{x})$  关于  $x_i$  的增减变化和  $x_j$  相关联，即一阶偏导数  $\frac{\partial f(\mathbf{x})}{\partial x_i}$  的正负性受到  $x_j$  的影响。因此，要找到  $f(\mathbf{x})$  关于  $x_i$  的最优解，就不得不考虑  $x_j$ ，既  $x_i$  和  $x_j$  相互影响，面对这样相互影响的决策变量，在分组时应该将其分为一组同时优化以保证求解质量。在进行可区分分析时，需要遍历所有决策变量并检查两两决策变量之间是否存在满足公式（2-8）的四个变量值。

### (2) 控制性质分析策略

控制性质分析策略最先由马晓亮等人<sup>[2]</sup>提出。该策略分析哪些变量影响多样性，哪些变量在种群收敛方面起重要作用。具体地，该策略通过观察添加扰动前后解的支配关系将决策变量划分为三类：位置变量、混合变量和距离变量。其中，若扰动前后的解是相互非支配的，则对应的决策变量被分类为位置变量，位置变量主要影响解的多样性；若扰动前后解之间出现了支配关系，则对应的决策变量被分类为距离变量，距离变量主要影响解的收敛性质；既不是位置变量也不是距离变量的决策变量被分类为混合变量。其中，位置变量只需要进行微调即可保持解的多样性，而距离变量则需要进一步的优化才能获得收敛性较好的解。

控制性质分析的具体操作如算法2-2所示，该算法输入为决策变量维度  $D$ ，预设的采样数量  $NCA$ ，和需要进行控制性质分析的决策变量  $\mathbf{x}$ 。首先初始化采样集  $S$ （第2行），然后对  $\mathbf{x}$  中的每一个决策变量  $x_i$  添加扰动（第6行），并计算扰动后的目标变量值（第7行），将其添加至采样集  $S$ （第8行）。随后，对采样集  $S$  进行非支配排序（第11行），并根据非支配排序结果，得到对  $x_i$  的分析结果（第12-18）行。当所有的决策变量分析完毕时，分析过程结束（第20行）。

### (3) 角度聚类分析策略

该策略提供了一种基于角度聚类的决策变量划分思路，由张兴义等<sup>[2]</sup>首次提出。该策略与控制性质分析策略类似，在执行聚类分析时，首先对每个决策变量进行若干次扰动，然后对扰动后的决策变量进行拟合，计算决策变量拟合线和收敛方向的夹角。然后基于夹角角度，使用  $k\text{-means}$ <sup>[2]</sup>算法对决策变量进行聚类，将对应夹角角度较小的决策变量分类为收敛相关变量，而对应夹角角度较大的

---

**算法 2-3: 角度聚类策略**


---

输入: 扰动次数 ( $sel$ ), 聚类数量 ( $per$ )

输出: 多样性相关变量 (DV), 收敛相关变量 (CV)

```

1:  $i \leftarrow 1$ 
2: while  $i \leq D$  do
3:    $Ps \leftarrow$  随机选择  $per$  个解
4:    $j \leftarrow 1$ 
5:   while  $j \leq sel$  do
6:      $SP \leftarrow$  随机扰动  $Ps$  中第  $j$  个解的第  $i$  个决策变量  $sel$  次产生新种群
7:      $L \leftarrow$  在目标空间中对  $SP$  进行拟合操作, 得到拟合直线
8:      $Angle[i][j] \leftarrow$  计算  $L$  和超平面  $f_1 + \dots + f_M = 1$  的夹角, 其中  $f_1, \dots, f_M$  为第  $i$  个决策变量对应的目标变量
9:      $MSE[i][j] \leftarrow MSE(L)$ 
10:   end while
11: end while
12:  $CV \leftarrow$  筛选  $MSE$  中小于  $1e-2$  的决策变量
13:  $[S1, S2] \leftarrow$  基于  $Angle$  将决策变量聚类
14: if  $CV \cap S1 \neq \emptyset$  and  $CV \cap S2 \neq \emptyset$  then
15:    $CV \leftarrow CV \cap S$ , 其中  $S$  为  $S1$  和  $S2$  中聚类中心较小的一个
16: end if
17:  $DV \leftarrow \{j = 1, \dots, D | j \notin CV\};$ 

```

---

决策变量被分类为多样性相关决策变量。具体的划分流程如算法2-3所示, 该策略有预设两个关键参数: 扰动次数  $sel$  和聚类数量  $per$ , 在进行聚类时, 首先需要从种群中随机选择  $per$  个解参与, 这些被选中的解保存在  $Ps$  里 (第 3 行)。然后, 对于  $Ps$  中的每一个解, 对其进行  $sel$  次扰动, 生成新种群  $SP$  (第 6 行), 计算  $SP$  在目标空间中的拟合直线  $L$  (第 7 行) 及其于决策空间超平面的夹角 (第 8 行), 并计算拟合直线的均方误差 (第 9 行)。紧接着, 以均方误差为依据, 选择均方误差较小的决策变量为收敛相关变量  $CV$  (第 12 行), 然后使用 k-means 根据夹角将决策变量分成两组  $S1$  和  $S2$ 。如果  $CV$  与  $S1$  和  $S2$  都不为空, 则将  $CV$  与具有较小夹角的一组相交 (第 14-16 行)。最后, 选取  $CV$  的补集作为  $DV$  (第 17 行)。

实验证明, 决策变量分析策略能够合理地对决策变量进行分组, 处理复杂问题。然而, 在进行决策变量分析时, 通常需要算法进行大量遍历和函数评估, 消耗大量计算资源。因此, 现有的决策变量分析方法效率有待提高。



## 2.3 本章小结

本章首先介绍了大规模多目标优化问题的基本概念和评价指标; 然后介绍了一些经典的求解 LSMOPs 的分组策略。



## 第3章 面向大规模多目标优化问题的双分解分组策略

为了解决现有决策变量随机分组策略会将相互关联的决策变量分为不同组而影响求解性能,本章提出了一种面向大规模多目标优化问题的双分解分组策略 (A Dual Decomposition Grouping Strategy for Large-scale Multi-objective Optimization Problems, DDS)。

### 3.1 双分解分组策略思想

双分解分组策略的基本思想为:在外分解操作的过程中,使用滑动窗口的将决策变量划分为相互重叠的组,从而将原本有着巨大复杂的搜索空间的 LSMOP 转换为一系列具有小型搜索空间的 MOPs,不同组相互重叠的部分可以缓解将相互关联的决策变量划分为同一组的问题,使算法能够在一定程度上跳出局部帕累托最优。然后,在外分解的过程中根据全局参考点和全局参考向量将每个 MOP 分解为一系列 SOPs。内分解操作进一步对原始问题进行分解。全局参考向量包含了原始决策空间中的最优信息,使得算法在搜索过程中具有全局视野,保证了每个 SOPs 的求解质量。最后,使用差分进化变异策略求解所有的 SOPs。在这个过程中,利用块坐标下降算法<sup>[7]</sup>维护原始问题的完整解,以保证求解质量。

算法3-1提供了嵌入 DDS 的 MOEA 的基本框架。其中  $A$  表示嵌入的 MOEA;输入为:滑动步长 ( $s$ )、窗口大小 ( $s$ )、双分解迭代次数 ( $It$ )、开始标记 ( $s\_t$ )、算法终止条件 ( $MaxFEs$ ) 和种群大小 ( $N$ )。该框架使用当前函数评估次数 (Function Evaluations,  $FEs$ ) 和最大函数评估次数 (Maximum Function Evaluations,  $MaxFEs$ ) 控制算法的迭代过程。具体实现过程为:算法从种群  $P$  的随机初始化开始 (第 1 行)。当 DDS 开关状态为开时,即  $Switch == True$ ,且当前函数评价次数 ( $FEs$ ) 大于开始标记 ( $s\_t$ ) 时,DDS 被激活 (第 4 行)。首先生成大小为  $w$  的滑动窗口,开始进行外分解过程来对所有决策变量进行分组 (第 8-20 行)。滑动窗口每滑动  $s$  个决策变量 (第 8 行),就得到一个决策变量组,该决策变量组使用  $X_{partial}$  表示。每个决策变量组  $X_{partial}$  都对应着一个 MOP (用  $G$  表示) (第 12 行)。然后,执行内分解过程,将每个  $G$  分解都为一系列 SOPs (第 14 行)。随后,基于块坐标下降的思想,使用差分变异策略对每个 SOP 进行优化。当所有的决策变量被优化一次后,就完成了完整的双分解过程。在整个算法中,双分解过程需要迭代多次,迭代次数由参数  $It$  控制,当双分解过程迭代  $It$  次后,整个 DDS 执行完成,此时开关设置为关闭状态,即  $switch == False$  (第 21 行)。当 DDS 执行完

---

**算法 3-1: DDS-A**


---

输入： 滑动步长 ( $s$ )，窗口大小 ( $w$ )，双分解迭代次数 ( $It$ )，开始标记 ( $s\_t$ )，算法终止条件 ( $MaxFEs$ )、种群大小 ( $N$ )

输出： 输出种群  $P$

```

1:  $P \leftarrow$  初始化种群 ( $N$ )
2:  $FEs \leftarrow 0, Switch \leftarrow True$ 
3: while  $FEs < Max\_FEs$  do
4:   if  $Switch == True$  and  $FEs > s\_t$  then
5:     while  $It > 0$  do
6:        $It \leftarrow It - 1$ 
7:       初始化滑动窗口, 准备开始外分解
8:       /* 外分解过程开始 */
9:       选择滑动窗口覆盖的决策变量, 保存在新的决策变量集合  $X_{partial}$  中
10:      while  $X_{partial} \neq \emptyset$  do
11:        /* 内分解过程开始 */
12:        生成关于  $X_{partial}$  的小规模多目标优化子问题  $G$ 
13:        根据  $G$  做内分解, 生成一组单目标优化问题
14:        根据块坐标下降算法的思想, 使用差分进化算法优化每一个单目标优化问题
15:        更新  $FEs$ 
16:        滑动窗口滑动  $s$ , 更新  $X_{partial}$ 
17:        /* 内分解过程结束 */
18:      end while
19:    end while
20:    /* 外分解过程结束 */
21:     $switch \leftarrow False$ 
22:  else
23:     $P \leftarrow A(P)$ 
24:    更新  $FEs$ 
25:  end if
26: end while

```

---

毕时, 种群  $P$  在目标空间中的位置已经非常接近  $PF_{true}$ 。此时, 再利用算法  $A$  在接近  $PF_{true}$  的小的搜索范围内进行精细优化以进一步提升解的质量 (第 23 行)。

为了更清晰地阐述 DDS 思想, 图3-1提供了在一个具有三个决策变量  $x_1, x_2, x_3$  的双目标优化问题上的双分解策略示意图。其中, 滑动窗口大小设置为 2, 步长被设置为 1。在执行 DDS 时, 首先初始化滑动窗口, 对决策变量进行分组执行外分解操作, 开始时滑动窗口覆盖了  $x_1, x_2$ , 这时生成一个关于  $x_1$  和  $x_2$  的小规模 MOP。接着, 滑动窗口向前移动一步覆盖  $x_2, x_3$ , 这时生成一个关于  $x_2$  和  $x_3$  的小规模 MOP。滑动窗口每向前滑动一次, 就会生成一个对应的 MOP, 然后每个 MOP 再进一步被转换为多个 SOPs。最后基于 BCD 思想, 利用差分进化算法

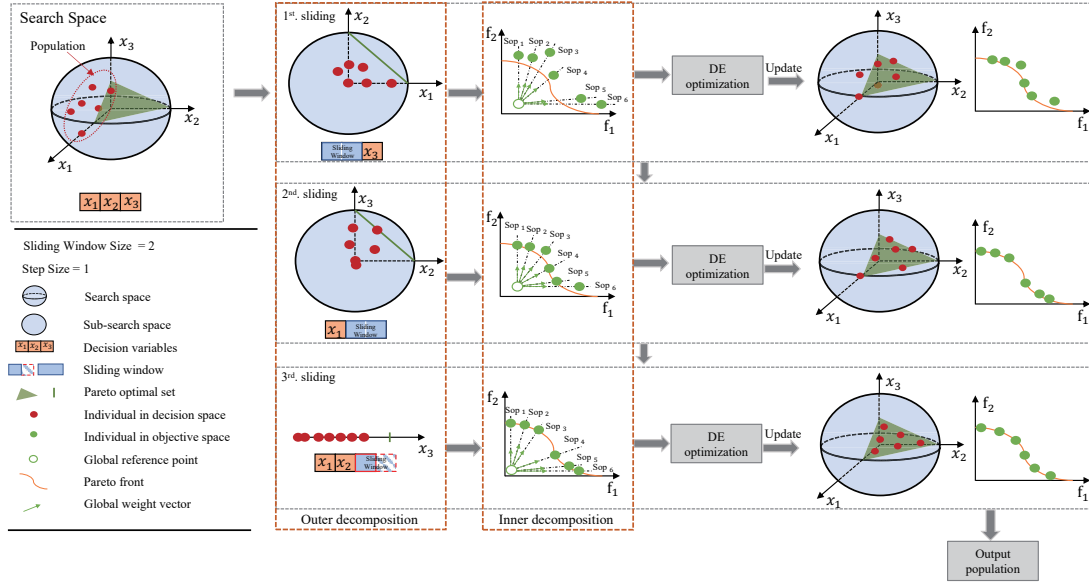


图 3-1 在具有三个决策变量的双目标优化问题上的 DDS 示意图

Fig. 3-1 DDS schematic for a double objective optimization problem with three decision variables

优化这些 SOPs。当滑动窗口滑动整个决策向量时，完成 DDS 的一次迭代。当迭代达到最大函数评价次数时，输出是优化后的种群。

## 3.2 双分解分组策略描述

双分解分组策略包括如下三个关键子操作：外分解操作、内分解操作和基于块坐标下降的差分进化优化策略。首先，外分解使用滑动窗口将大规模决策变量划分为具有重叠部分的小规模决策变量组。每次滑动窗口滑动时，都会产生一个决策变量组，每个决策变量组对应着一个小规模的 MOP，这个 MOP 的搜索空间是原始问题搜索空间的一部分。一旦生成小规模 MOP，内分解就立即创建一组全局方向向量，将每个小规模 MOP 分解为一组 SOPs。最后，使用差分变异算法对所有的 SOPs 进行优化，使用块坐标下降的思想维护原始问题的完整解。下面将详细介绍三个关键子操作。

### 3.2.1 外分解操作

滑动窗口技术是一种求解高维向量相关问题的经典方法。在目标检测<sup>[2]</sup>、区块链<sup>[2]</sup>技术<sup>[2]</sup>、异常检测<sup>[2]</sup>等领域有着广泛的应用。滑动窗口对特定大小的子向量而不是整个向量执行特定的操作，可见这与决策变量分组的目的非常相似，这也是本研究尝试在外分解中用滑动窗口划分决策变量的原因之一。

外分解使用滑动窗口将大规模决策变量划分为重叠的决策变量组。这样做

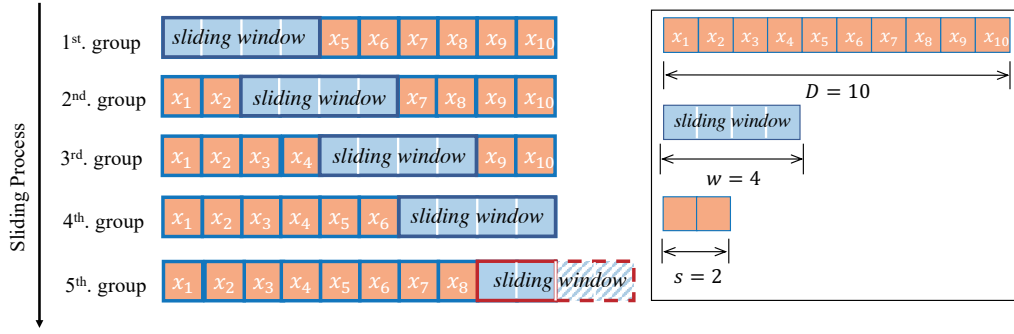


图 3-2 一个简单的滑动窗口分解过程的例子

Fig. 3-2 A simple example of the outer decomposition process by the sliding window

有两个好处：一方面，它可以降低直接求解大规模问题的难度。另一方面，相互重叠的决策变量组一定程度上缓解了相互影响的决策变量被分成不同组而对算法性能产生影响的问题。

在 DDS 中，滑动窗口有两个基本控制参数：窗口大小 ( $w$ ) 和滑动步长 ( $s$ )，且  $s < w$ 。窗口大小决定了每组中决策变量的数量，滑动步长与相邻两组间决策变量重叠的个数有关。规定滑动步长小于窗口大小，以确保相邻的两个决策变量组之间具有重叠的部分，重叠变量的数量为  $(w - s)$ 。每次滑动窗口滑动一步，就会得到一个新的决策变量组，从而在最新的决策变量组上生成一个新的多目标优化问题。显然，滑动步长越小，相邻的两组变量重叠越多，两个决策变量组所对应的 MOPs 的搜索空间重叠的也越多。

图3-2给出了基于滑动窗口的分解过程的一个示意图。在图3-2中，有 10 个决策变量，即决策向量维度  $D$  为 10。滑动窗口大小  $w$  为 4，滑动步长  $s$  为 2。首先，初始化滑动窗口，并将窗口左端对齐第一个决策变量，得到第一个决策变量组。然后，滑动窗口从左到右一步一步滑动。每滑动一步就得到一个决策变量组。直到滑动窗口不再覆盖任何决策变量，便结束外分解。由于窗口大小  $w$  和滑动步长  $s$  是固定的，当决策变量长度  $D$  不能被  $(s + w)$  整除时，在滑动窗口的最后几次滑动中，滑动窗口将超出决策变量。此时，规定只有滑动窗口覆盖的决策变量需要参与后续的优化过程。例如，如图3-2所示，在最后一次滑动时，滑动窗口将超出决策向量。只有滑动窗口覆盖的两个决策变量 (被红色实线包围的部分) 将参与后续的优化操作。

显然，相邻两个决策变量组之间的重叠部分会被反复多次优化，这可能会浪费 FESs，从而影响求解效率。而另一方面，重叠的部分也可能会缓解相互影响的

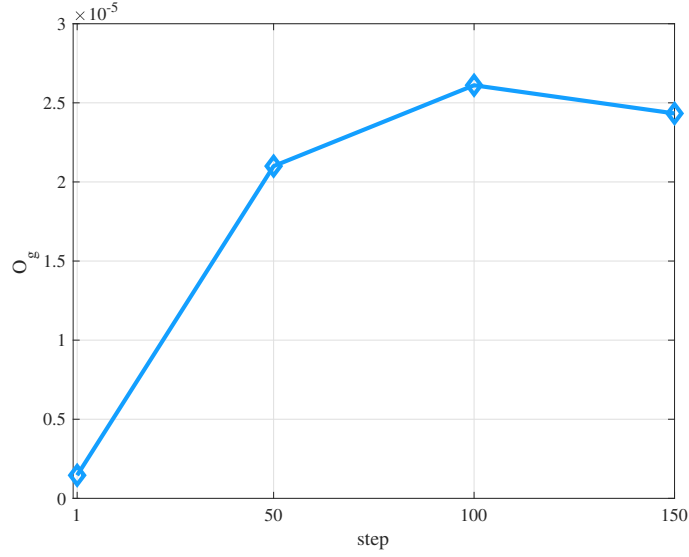


图 3-3 在  $w=150$ 、 $D=200$  的 LSMOP1 上将步长  $s$  设置为 1、50、100、150 时的  $O_g$  曲线

Fig. 3-3 The  $O_g$  curve when the step  $s$  is set to 1, 50, 100, 150 on LSMOP1 with  $w = 150$ ,  $D = 200$

决策变量被分成不同组而影响求解性能的问题。为了分析滑动窗口中重叠部分对求解质量和求解效率的影响，本研究定义了重叠增益 ( $O_g$ ) 的度量：

$$O_g = \frac{PM_1 - PM_2}{FE_2 - FE_1 + gap} \quad (3-1)$$

其中， $FE_1$ 、 $FE_2$  分别表示外分解执行前后的  $FEs$ ； $gap$  是固定的常数，用来提升  $O_g$  的鲁棒性； $PM_1$ 、 $PM_2$  分别表示  $FEs$  为  $FE_1$  时和  $FE_2 + gap$  时具体评价指标（如 IGD）的值。从等式（3-1）可以看出， $O_g$  与外分解前后的性能提升呈正相关，与外分解过程消耗的  $FEs$  呈负相关。因此， $O_g$  可用于反映两个决策变量组之间的重叠部分大小，即滑动步长的值对 DDS 的性能的影响。因此， $O_g$  可以在求解效率（较大的  $s$ ）和求解质量（较小的  $s$ ）之间找到折衷。图3-3展示了在 LSMOP1<sup>[7]</sup>上的  $O_g$  曲线， $s$  设置为 1、50、100 和 150，决策变量数为 200， $w$  设置为 150、性能指标为 IGD。图3-3清晰地反映出，当滑动步长太小 ( $s = 1$ ) 时，会有许多重叠的变量，导致此时  $O_g$  的值很小。这是因为过小的滑动步长导致在外分解过程中生成了太多的决策变量组，优化这些过多的决策变量组会消耗大量的  $FEs$ ，使得  $O_g$  的值显著减小。同时，在滑动步长较大的情况下，决策变量组之间重叠变量的数量减少， $O_g$  的值随之增加。然而，图3-3显示，在  $s = 150$  的情况下， $O_g$  的值低于在  $s = 100$  的情况下的值，这是因为过大的滑动步长会导致决策变量组之间重叠的部分变少，可能会忽略掉不同组之间的决策变量相互作用的情况，从而导致优化前后的 IGD 指标变化较小，从而导致  $O_g$  值的减小。因

---

**算法 3-2: 内分解**


---

输入： 种群大小 ( $N$ ), 种群 ( $P$ ), 决策变量组 ( $X_{partial}$ )

输出： 一组和  $X_{partial}$  相关的单目标优化问题

- 1: 根据公式 (3-2) 计算全局参考  $Z$
  - 2: 创建均匀分布的全局方向向量  $\{V_i\}$
  - 3: 将全局参考点  $Z$  和全局方向向量  $\{V_i\}$  投影到关于  $X_{partial}$  的子空间中
  - 4: 得到关于  $X_{partial}$  的局部参考点  $Z^*$  和一组局部方向向量  $\{\lambda_i\}$
  - 5: 使用切比雪夫方法得到  $N$  个单目标优化问题
- 

此, 这些结果表明, 选择适当的  $s$  对决策变量进行重叠分组对提高算法的性能至关重要。

### 3.2.2 内分解操作

虽然基于滑动窗口的分解可以通过对决策变量分组降低 LSMOP 的求解难度, 同时也缓解了相互作用变量被分成不同组的问题, 但在小规模子 MOP 的搜索空间中搜索可能会影响解的多样性。因此, 为了保证最终的求解质量, 一旦在分解中产生了决策变量组对应的子 MOPs, DDS 就会对其进行进一步的分解。首先, 计算全局参考点  $Z = (Z_1, Z_2, \dots, Z_l, \dots, Z_m)$ , 其中  $m$  是优化目标的个数,  $Z_l, l = 1, 2, \dots, m$  是目标  $l$  到目前为止在  $N$  个个体解中的最优值, 定义为:

$$\text{Min}(Z_l, f_l(\mathbf{x}^i)), \quad i = 1, 2, \dots, N \quad (3-2)$$

随后, 创建一组大小为  $N$  的均匀分布的全局方向向量, 记为  $\{V_i\}, i = 1, 2, \dots, N$ , 其中,  $V_i = v_{i1}v_{i2} \dots v_{ij} \dots v_{im}$  是原始目标空间中的  $m$  维向量, 并满足  $\forall j = 1, 2, \dots, m, v_{ij} \geq 0, \sum_{j=1}^m v_{ij} = 1$ 。然后, 将全局参考点  $Z$  和每个全局方向向量  $V_i$  投影到相应的子 MOP 的子空间中, 以获得局部参考点向量  $Z^*$  和一组局部方向向量  $\{\lambda_i\}$ 。最后, 利用切比雪夫分解方法<sup>[7]</sup>将相应的子 MOP 变换为一组大小为  $N$  的 SOPs。由于切比雪夫分解方法在基于分解的 MOEAs 中十分常见, 因此这里不再对其进行描述。算法3-2显示了内分解的过程。值得注意的是, 内分解是在全局参考点和全局方向向量所包含的一些全局信息的指导下进行的。因此, 从全局的角度来看, 全局信息在一定程度上有利于解的多样性。

### 3.2.3 基于块坐标下降的差分进化优化策略

为了进一步确保解的多样性, 本研究采用差分进化 (Differential Evolution, DE) 算法作为求解每个 SOP 的基本优化器。



DE 是一种利用交叉变异生成新候选解的高效 EA。具体地, 对于一个决策向量  $\mathbf{x} = \{x_1, x_2, \dots, x_D\}$ , 首先执行变异操作, 生成变异向量  $\mathbf{v}$ :

$$\mathbf{v} = x_{r_1} + F_1 (x_{r_2} - x_{r_3}) + F_2 (x_{r_4} - x_{r_5}) \quad (3-3)$$

其中,  $r_1, r_2, r_3, r_4, r_5 \in 1, 2, \dots, N$  是在种群中随机采样得到的五个不同的决策变量编号;  $F_1, F_2$  为缩放因子。随后, 对  $\mathbf{x}, \mathbf{v}$  执行交叉操作:

$$x_i^* = \begin{cases} \mathbf{v}_i & \text{rand}(0,1) > Cr \\ \mathbf{x}_i & \text{otherwise} \end{cases} \quad (3-4)$$

其中,  $Cr$  是预设的交叉率。

此外, 本研究在使用 DE 求解每个小规模 SOP 的优化过程中通过始终保留初始 LSMOP 的完整解来使得 DE 在分组后的子空间搜索的过程中具有全局视野。具体的方法是将块坐标下降 (Block coordinate decent, BCD) 的思想引入到 DE 的优化过程中。BCD 是一种经典的非梯度优化算法。它的基本思想可以概括如下: 每次 BCD 只搜索决策变量的一个子集, 同时保持其余决策变量固定。当获得当前决策变量子集的最优解时, 再搜索另一子集。即 BCD 在上次优化结果的基础上进行当前时刻的优化过程。一旦所有决策变量都被优化一次, 一轮迭代结束。在所提出的 DDS 中, 基于 BCD 的优化过程描述如下: 外部分解使用滑动窗口将所有决策变量划分到重叠的组中。每组决策变量包含  $w$  个决策变量。相邻的两组决策变量有  $w-s$  个重叠的决策变量。假设初始 LSMOP 的决策向量  $\mathbf{x}$  被分成  $k$  组, 那么  $\mathbf{x}$  可以表示为:

$$\mathbf{x} = \overbrace{x_1, x_2, \dots, x_{s+1}, \dots, x_w}^{\theta_1}, \underbrace{\dots, x_{s+w}, \dots, x_D}_{\theta_2}, \dots, \overbrace{\dots, x_D}^{\theta_k} \quad (3-5)$$

由于内分解没有进一步划分决策变量, 因此双分解后的每个 SOP 都与等式 (3-5) 中的一组决策变量  $\theta_i (i = 1, \dots, k)$  相关。因此, 基于 BCD 的基本思想, DE 在求解每个 SOP 时只优化对应的一组决策变量, 而其它决策变量保持不变。例如, 当 SOP 为等式 (3-5) 中的  $\theta_2$  时, 只有  $w$  个变量参与优化, 既  $x_{s+1}, \dots, x_{s+w}$  被优化, 而其它决策变量保持不变。此时, 对应的 SOP 可以表述为:

$$\theta_2^* = \underset{\theta_2}{\operatorname{argmin}} f(x_1, \dots, x_s, \theta_2, \dots, x_D) \quad (3-6)$$

每组决策变量  $\theta_i (i = 1, 2, \dots, k)$  所对应的 SOP 都经过 DE 求解后, 就完成了

一轮基于 BCD 的优化过程。显然，BCD 保证了 LSMOP 解的完整性，因为在整个优化过程中的整个决策向量  $\mathbf{x}$  总是使用了当前求解过程中的最新求解结果。根据 BCD 的思想，双分解优化后的  $X_{partial}$  会立即替换  $\mathbf{x}$  中对应的部分，使算法在求解过程中逐步逼近  $PF_{true}$ 。

### 3.3 时间复杂度分析

在这一节中，我们将分析 DDS 的时间复杂度。从 DDS 的总体框架来看（算法 3-1），DDS 中每一次迭代的主要计算代价来自三个主要部分：1) 外分解，2) 内分解，3) SOP 优化。令  $N$  表示种群规模， $D$  表示决策变量个数， $M$  表示目标个数， $It$  表示外分解的迭代次数。接下来，将分别讨论每部分的计算复杂性。

(1) 外分解主要包括生成滑动窗口（算法 3-1，第 7 行）、选择滑动窗口覆盖的决策变量（算法 3-1，第 9 行）以及向前滑动滑动窗口的过程（算法 3-1，第 16 行）。其中，前两项的时间复杂度为  $O(1)$ 。而滑动窗口滑动的时间复杂度主要有滑动次数决定，滑动次数由滑动步长  $s$  确定。平均滑动次数为  $\frac{\sum_{s=1}^D \lfloor D/s \rfloor}{D} = \sum_{s=1}^D \frac{1}{s}$ ，其中  $\sum_{s=1}^D \frac{1}{s} \approx \int_1^D \frac{1}{x} dx = \log D$ 。因此，滑动窗口滑动的时间复杂度为  $O(\log D)$ 。此外，外分解过程需要重复  $It$  次。因此，外分解部分的时间复杂度为  $O(It \log D)$ 。

(2) 内分解主要包括以  $O(M)$  的时间复杂度计算全局参考点（算法 3-2，第 1 行）、以  $O(1)$  的时间复杂度计算全局方向向量（算法 3-2，第 2 行）、以  $O(1)$  的时间复杂度投影全局向量（算法 3-2，第 3 行）以及用切比雪夫方法得到  $N$  个 SOPs（算法 3-2，第 5 行），该部分的时间复杂度为  $O(1)$ 。因此，内分解部分的时间复杂度为  $O(M)$ 。

(3) SOP 优化过程（算法 3-1，第 14 行）使用 DE 来优化  $N$  个 SOPs。每个 SOP 都有  $w$  个决策变量。DE 中的每个个体代表一个 SOP 的解。因此，求解与  $w$  个决策变量相关的  $N$  个 SOPs 的时间复杂度为  $O(Nw)$ 。

基于以上分析，对于每一次迭代，DDS 的计算复杂性为  $O(MNIt \log D + Itw \log D)$ 。

### 3.4 收敛性讨论

在这一节中，我们将讨论 DDS 的收敛性。为方便讨论，做如下符号约定： $N$  表示种群大小； $k$  表示滑动窗口的第  $k$  次滑动  $P^{(k)} = \{\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_N^k\}$  表示滑动窗口在第  $k$  次时的种群； $X_{partial,i}^{(j)}$  表示滑动窗口在第  $k$  次滑动时，在  $\mathbf{x}_i^{(k)}$  上所覆盖的决策变量组； $X_{partial,i}^{(k)}$  所对应的种群  $P_{partial}^{(k)}$  为  $\{X_{partial,1}^{(K)}, X_{partial,2}^{(K)}, \dots, X_{partial,N}^{(K)}\}$ 。

$F_i^{(k)}$  表示  $X_{patial,i}^{(k)}$  所对应的目标向量。

DDS 首先在  $P^{(0)}$  上使用外分解, 得到  $P_{partial}^{(0)}$ , 此时原本对  $F_i^{(0)}$  的优化被转换为对  $F_{patial,i}^{(0)}$  的分解。随后, 内分解进一步将  $F_{patial,i}^{(0)}$  分解转换为  $G^{(0)} = \{g_1^{(0)}, g_2^{(0)}, \dots, g_N^{(0)}\}$ , 其中  $g_i^{(0)}, i = 1, 2, \dots, N$  为  $X_{patial,i}^{(0)}$  对应的单目标优化问题。最后, 每个单目标优化问题依次通过 DE 优化, 得到  $\hat{G}^{(0)} = \{\hat{g}_1^{(0)}, \hat{g}_2^{(0)}, \dots, \hat{g}_N^{(0)}\}$ , 此时对应的解记为  $\hat{P}_{parital}^{(0)}$ , 对应的多目标问题的目标向量为  $\hat{F}_{partial}^{(0)}$ 。在使用 DE 优化时,  $g_i^{(0)}$  被优化得到  $\hat{g}_i^{(0)}$ , 当且仅当  $\hat{g}_i^{(0)} \leq g_i^{(0)}$  时,  $\hat{g}_i^{(0)}$  的解才会替换  $g_i^{(0)}$  的解, 即, 当单目标优化问题的解得到更新时, 下式一定成立:

$$\hat{g}_i^{(0)} \leq g_i^{(0)}, i = 1, 2, \dots, N \quad (3-7)$$

根据  $F_{parital,i}^{(0)}$  和  $G^{(0)}$  的对应关系, 当公式 (3-7) 成立时, 下式也同时成立:

$$\hat{F}_{partial,i}^{(0)} \geq F_{partial,i}^{(0)}, i = 1, 2, \dots, N \quad (3-8)$$

根据 BCD 的思想, 时用  $\hat{P}_{partial}^{(0)}$  替换  $P^{(0)}$  中对应部分的决策变量组, 得到  $P^{(1)}$ , 根据目标向量在外分解中的对应关系, 有:

$$F_i^{(1)} = F_{partial,i}^{(1)} \leq F_{partial,i}^{(0)} = F_i^{(0)}, i = 1, 2, \dots, N \quad (3-9)$$

紧接着, 滑动窗口向前滑动以执行  $P^{(1)}$  的外分解, 类似于上面的过程:

$$F_i^{(2)} = F_{partial,i}^{(2)} \leq F_{partial,i}^{(1)} = F_i^{(1)}, i = 1, 2, \dots, N \quad (3-10)$$

滑动窗口向前滑动时, 如下偏序关系成立:

$$\dots \leq F_i^{(2)} \leq F_i^{(1)} \leq F_i^{(0)} \quad (3-11)$$

也就是说,  $PF_{approx}$  将逐步接近  $PF_{true}$ 。

### 3.5 实验结果及分析

为了检验 DDS 的性能, 本节进行如下实验: 首先, 对 DDS 的关键参数进行参数敏感性分析; 随后, 验证了 DDS 的有效性; 接着在不同的  $FEs$  设置下验证 DDS 的求解效率; 然后在测试函数集上将 DDS 嵌入现有 MOEA 并与七个先进的大规模多目标优化算法和一个基于深度学习框架的算法进行实验对比, 综合验证 DDS 的有效性, 最后, 在两个实际应用问题上进一步验证了 DDS 的性能。

表 3-1 LSMOP 测试函数集问题特点及变量设置

Tab. 3-1 LSMOP test suit characteristics and variable settings

Problems	决策变量个数	目标变量个数	PF 特征	目标函数可分离性
LSMOP1	200-2 000	2	线性	可分离
LSMOP2	200-2 000	2	线性	部分可分离
LSMOP3	200-2 000	2	线性	混合可分离
LSMOP4	200-2 000	2	线性	混合可分离
LSMOP5	200-2 000	2	凹面	可分离
LSMOP6	200-2 000	2	凹面	部分可分离
LSMOP7	200-2 000	2	凹面	混合可分离
LSMOP8	200-2 000	2	凹面	可分离
LSMOP9	200-2 000	2	不连续	可分离

### 3.5.1 实验设置

#### (1) 基准测试问题

本实验采用 LSMOP1-9<sup>[21]</sup>作为测试函数集, 包括 9 个测试问题 (LSMOP1-LSMOP9)。测试问题的特点以及变量设置见表 3-1。其中, 目标函数可分离性指的是决策变量的相互用关系, 当目标函数可分离时, 决策变量之间无相互作用, 可独立优化, 即:

$$\operatorname{argmin}_{\mathbf{x}} f(\mathbf{x}) = \left( \operatorname{argmin}_{x_k} f(\mathbf{x}), \operatorname{argmin}_{\forall x_j, j \neq k} f(\mathbf{x}) \right) \quad (3-12)$$

其中,  $\mathbf{x}$  表示决策变量;  $f(\cdot)$  表示目标函数。当目标函数不可分离时, 则每个决策变量之间都存在相互作用, 无法独立优化。混合可分离即部分决策变量可以独立优化, 读部分决策变量不可以独立优化。部分可分离指的是决策变量以组为单位可以独立优化的, 即:

$$\operatorname{argmin}_{\mathbf{x}} f(\mathbf{x}) = \left( \operatorname{argmin}_{\mathbf{x}_1} f(\mathbf{x}_1, \dots), \dots, \operatorname{argmin}_{\mathbf{x}_m} f(\dots, \mathbf{x}_m) \right) \quad (3-13)$$

#### (2) 参数设置

在 DDS 中, 有四个参数: 开始标记 ( $s\_t$ )、窗口大小 ( $w$ )、滑动步长 ( $s$ ) 和双分解迭代次数 ( $It$ )。具体地, 当决策变量的数目为 200、500、1000 和 2000 时, 开始标记分别被设置为 40000、40000、200000 和 1000000。将双分解迭代次数、窗口大小和滑动步长分别设置为 10、150、50。此外, 种群大小  $N$  被设置为 100。对于其它比较的参数, 算法使用 PlatEMO 中的默认参数值进行公平比较。

#### (3) 终止条件

本研究使用的终止条件为  $MaxFEs$ 。当决策变量  $D$  的数目分别为 200、500、1 000 和 2 000 时,  $MaxFEs$  分别为 4 000 000、1 000 000、2 000 000 和 6 000

000。应该注意的是,上述  $MaxFEs$  比大多数现有的用于 LSMOPs 的 MOAEs 设置相对较小,后者通常将  $MaxFEs$  设置为  $D \times 15\,000$ 。为了进行公平的比较,我们进行了额外实验,以考察当  $MaxFEs = D \times 15\,000$  时 DDS 的性能。

#### (4) 其它

每个算法在每个测试问题上独立运行 10 次。收集每次测试的 IGD 平均值和标准差。用 Wilcoxon 秩和检验对 DDS-A 算法和各种算法的结果进行统计学意义检验,显著性水平  $\alpha = 0.05$ 。符号“+”、“-”和“ $\approx$ ”分别表示比较算法显著好于 DDS-A、显著差于 DDS-A 和统计上类似于 DDS-A。

### 3.5.2 参数敏感性分析

在 DDS 中,有四个关键参数,包括开始标记 ( $s_t$ )、窗口大小 ( $w$ )、滑动步长 ( $s$ ) 和双分解迭代次数 ( $It$ )。为了显示它们对 DDS 性能的影响,我们为每个参数设置了八个候选值进行比较。我们从简单到复杂选择了三个测试问题,包括 LSMOP2、LSMOP6 和 LSMOP7。其中,LSMOP2 具有线性 PS,LSMOP6 和 LSMOP7 具有非线性 PS,但 LSMOP6 决策变量之间的交互作用更为复杂。实验结果如图3-4所示。在此图中,每条线上的不同色块表示与 IGD 值相同的测试问题上的不同参数值。色块的颜色越浅,IGD 的值越小,这意味着 DDS 取相应的参数值时性能更好。下面讨论了不同参数对算法性能的影响:

#### (1) 双分解迭代次数 ( $It$ )

参数  $It$  表示 DDS 中双分解迭代的次数,其中一次迭代指的是滑动窗口在决策变量上滑动。从图3-4可以看出,10 是  $It$  的最佳值。较大的  $It$  值将导致较差的算法性能。这是因为过多的迭代会浪费更多的  $FEs$ ,这可能会影响后续的全局搜索,并导致算法性能较差。因此,建议  $It$  的取值为 10。

#### (2) 开始标记 ( $s_t$ )

$s_t$  参数决定了 DDS 在现有 MOEA 中的嵌入时间点。从图3-4可以看出,当决策变量的数量为 500 时,DDS 的性能在  $s_t$  取 40 000 时更好。当 DDS 的嵌入时间过早时 ( $FEs < 10\% MaxFEs$ ),现有的 MOEA 没有得到适当的在搜索空间中探索,找不到相对较好的解决方案来引导 DDS 的搜索向正确的方向发展。在这种情况下,DDS 可能无法获得优解。当 DDS 的嵌入时间太晚时 ( $FEs > 90\% MaxFEs$ ),现有的 MOEA 消耗了大部分的  $FEs$ ,但并没有使得解的质量进一步提升。此时,由于  $FEs$  所剩无几,DDS 无法发挥提高解的质量的重要作用。因此,适当的嵌入时间对于 DDS 提高现有 MOEA 的性能至关重要。这些结果表

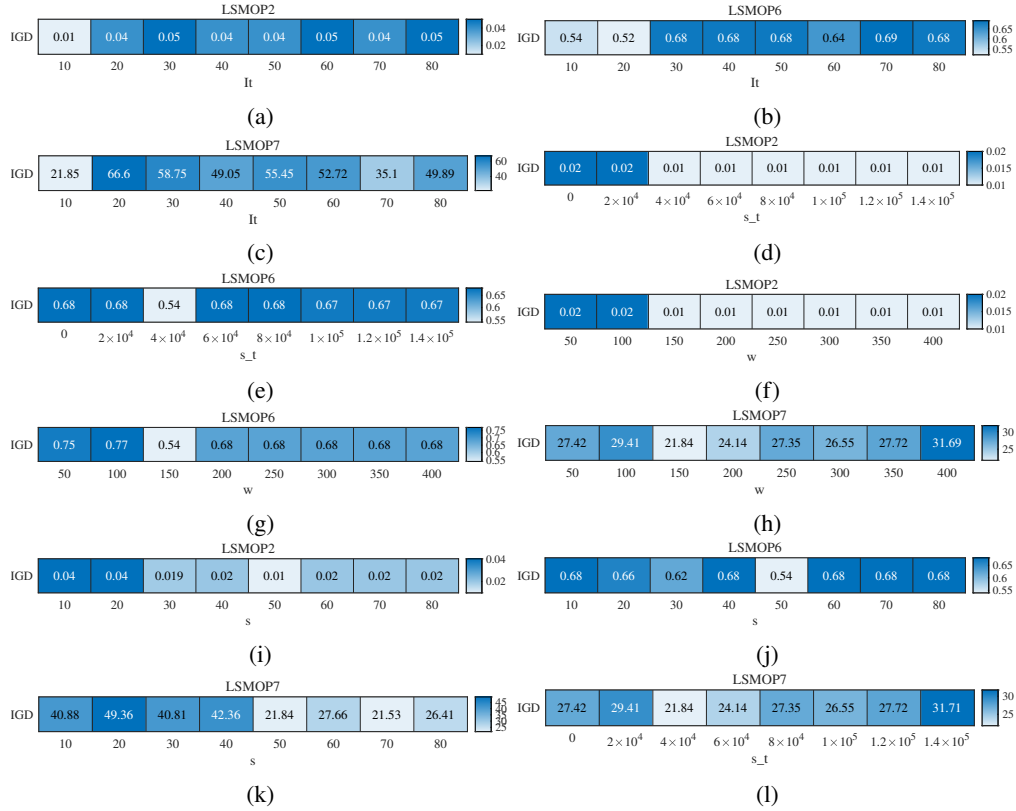


图 3-4 LSMOP2、LSMOP6、LSMOP7 上各参数的参数敏感性结果

Fig. 3-4 The parameter sensitivity result of each parameter on LSMOP2, LSMOP6, and LSMOP7

明, 当决策变量数为 500 时, 开始标记应设置为 4 000 000。同样, 当决策变量数为 200、1000、2 000 时,  $s_t$  分别设置为 40 000、200 000、1 000 000。

### (3) 窗口大小 ( $w$ )

$w$  参数决定了外分解中每个决策变量组的决策变量个数。从图3-4不难看出, 当  $w = 150$  时, DDS 的性能总体上好于  $w > 150$  时或  $w < 150$  时的性能。这是因为以下几个原因: 如果  $w$  太大, 则通过外部分解得到的新 MOP 仍然是 LSMOP, 在这种情况下, DDS 无法完全分解原始问题。因此, 当  $w > 150$  时, DDS 的性能较差。反之, 如果  $w$  太小, 相互作用的变量被分成不同组的风险增加, 从而降低 DDS 性能。基于以上分析, 窗口大小 ( $w$ ) 对于 DDS 改善现有 MOEA 在 LSMOP 上的性能至关重要。根据这些实验结果, 对于所有数量的变量,  $w = 150$  是一个相对合适的设置。

### (4) 滑动步长 ( $s$ )

$s$  确定两个相邻组中重叠变量的数量。从图3-4中不难发现,  $s = 50$  是最佳参数值。由于以下原因: 对于  $s$  较大或较小的值都会影响算法性能。较小的  $s$  值意味着两个相邻组中的重叠变量太多, 这会浪费大量的  $FES$ , 并且在固定  $MaxFES$

表 3-2 不同算法及他们的 DDS 版本的性能比较

Tab. 3-2 The performance comparison between different algorithms and their DDS version

Problem	Algorithms	Algorithms		+/-/≈
		SMEA	DDS-SMEA	
LSMOP2	SMEA V.S. DDS-SMEA	7.7811e-2 (2.73e-3) -	6.4692e-3 (8.24e-4)	0/2/1
LSMOP8		4.2556e-2 (2.61e-3) -	1.1691e-2 (2.67e-3)	
LSMOP9		5.6529e+0 (1.23e+1) ≈	3.2979e-1 (4.38e-1)	
LSMOP2	MOEAD/DE V.S. DDS-MOEAD/DE	7.2517e-2 (5.65e-3) -	5.9729e-3 (5.73e-4)	0/3/0
LSMOP8		4.1764e-2 (5.13e-3) -	2.3858e-2 (1.20e-2)	
LSMOP9		1.5467e-1 (1.01e-1) -	1.0278e-2 (2.36e-3)	
LSMOP2	ToP V.S. DDS-ToP	8.2250e-2 (2.05e-3) -	1.3034e-2 (1.10e-3)	0/2/1
LSMOP8		5.5798e-2 (1.10e-3) -	2.3938e-2 (9.95e-3)	
LSMOP9		4.2306e-1 (1.74e-1) ≈	5.0855e-1 (4.13e-1)	
LSMOP2	CPSMOEA V.S. DDS-CPSMOEA	7.8501e-2 (2.73e-3) -	1.2633e-2 (2.36e-4)	0/3/0
LSMOP8		5.3324e-2 (3.57e-3) -	2.1619e-2 (1.14e-2)	
LSMOP9		2.7109e-1 (3.97e-2) -	5.0326e-2 (5.82e-3)	
LSMOP2	MOEADCMA V.S. DDS-MOEADCMA	8.1869e-2 (1.31e-3) -	1.0357e-2 (8.64e-4)	0/3/0
LSMOP8		4.9249e-2 (1.78e-3) -	3.2085e-2 (1.81e-2)	
LSMOP9		3.4377e-1 (5.60e-3) -	8.8149e-2 (8.82e-2)	
LSMOP2	WVMOEAP V.S. DDS-WVMOEAP	3.2334e-1 (1.68e-4) ≈	3.2299e-1 (4.67e-4)	0/3/0
LSMOP8		4.7373e-1 (6.22e-3) ≈	4.7460e-1 (5.20e-3)	
LSMOP9		1.7962e+0 (3.65e-1) -	8.1004e-1 (1.24e-16)	

的情况下会降低算法性能。较大的  $s$  值意味着两个相邻组中重叠的变量太少，这导致交互变量很可能被分为不同的组，从而损害算法性能。因此，根据图3-4中的这些实验结果，在后续的实验中，设置  $s = 50$ 。

### 3.5.3 双分解分组策略的有效性验证

为了验证 DDS 是否能提高现有 MOEAs 在解决 LSMOPs 时的性能，本节将 DDS 嵌入到六种不同类型的优秀 MOEAs 中。这些 MOEAs 都发表在知名期刊上，用于解决具有少量决策变量的 MOPS。这六种算法分别是 SMEA<sup>[2]</sup>、MOEA/D-DE<sup>[2]</sup>、TOPCPS-MOE<sup>[2]</sup>、MOEAD-CMA<sup>[2]</sup>和 WV-MOE<sup>[2]</sup>。这些算法的 DDS 版本分别称为 DDS-SMEA、DDS-MOEAD/D-DE、DDS-TOP、DDS-CPS-MOE、DDS-MOEAD-CMA 和 DDS-WV-MOE-P。每种算法与其 DDS 版本进行了比较，决策变量的数量设置为 200。实验结果如表3-2所示，其中灰色背景的单元格表示对应算法取得了最佳性能。B/W/S 表示一种原始算法明显好于、差于或接近于其 DDS 版本的测试问题的数量。从该表中可以看出，在 18 种情况中，DDS 版本的算法在 14 种情况下明显优于其对应的原始算法。在其余四种情况下，DDS 版本的算法和其原始算法获得了类似的结果。这些结果表明，DDS 可以有效地改进现有求解 LSMOP 问题的多目标进化算法。在接下来的实验中，我们将以 DDS-MOEAD/D-DE 为代表，与最先进的大型 MOEA 进行比较，以评估

其解质量和计算效率。

### 3.5.4 双分解分组策略的效率验证

本小节以 DDS-MOEA/D-DE 为代表, 考察 DDS 在求解 LSMOPs 上的计算效率。由于 MOEA/DVA 需要花费大量的  $FES$  进行变量分析, 在变量个数等于或大于 500 的情况下, 无法在预设的  $MaxFES$  的条件线完成对 LSMOPs 的求解, 这表明 MOEA/DVA 的计算效率很差。因此, 本节不与 MOEA/DVA 进行比较。图3-5绘制了 WOF、LMOCSSO、LMOEA-DS、LSMOF、FDV、DVCOEA 和 DDS-MOEA/D-DE 在 LSMOP8 和 LSMOP9 测试问题上的收敛曲线。可以清楚地看到, 无论面对何种测试问题, DDS-MOEA/D-DE 的收敛曲线总是位于整个进化过程的底部, 表现出更快的收敛效率。特别是在最复杂的 LSMOP9 测试问题上, DDS-MOEA/D-DE 具有更为明显的优势。值得注意的是, 本研究设置的  $MaxFES$  相对较小, 通常的 LSMOPs 研究中将  $MaxFES$  设置为  $D \times 15000$ , 其中  $D$  是决策变量的数量。因此, 与大多数现有大型 MOEA 相比, DDS 所需要的  $MaxFES$  要高得多。图3-6绘制了 WOF、LMOCSSO、LMOEA-DS、LSMOF、FDV、DVCOEA 和 DDS-MOEA/D-DE 在决策变量数为 200 时, 对 LSMOP2、LSMOP8 和 LSMOP9 三个测试函数的中位数 IGD 值的收敛曲线。

图3-6显示, 将  $MaxFES$  设置为  $200 \times 15\ 000$  时, 尽管 7 种算法都获得了较好的解, 但 DDS-MOEA/D-DE 的收敛速度仍然最快。对于 DDS-MOEA/D-DE, 将  $MaxFES$  设置为  $200 \times 15\ 000$  意义不大, 因为当  $MaxFES$  为 4000 000 时, 它已经获得了完美的解。这表明 DDS-MOEA/D-DE 能以更小的  $MaxFES$  获得更好的解, 即计算效率最高。这一点再次间接反映了 DDS 能够提高现有 MOEA 在解决 LSMOPs 方面的性能。

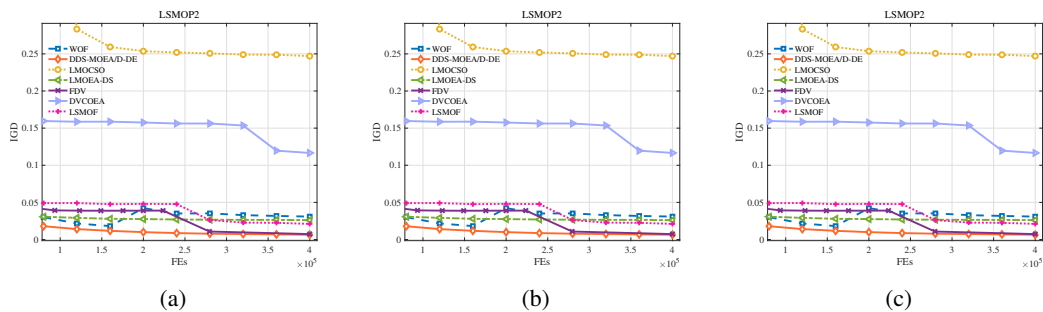


图 3-5 当决策变量设置为 200,  $MaxFES$  设置为 400 000 时, DDS-MOEA/D-DE 和其它六个算法在三个测试问题上的中位数 IGD 曲线

Fig. 3-5 The curve of median IGD values obtained by DDS-MOEA/D-DE and other six algorithms on three test problems when the number of decision variables is 200 and  $MaxFES$  is set to 4000 000



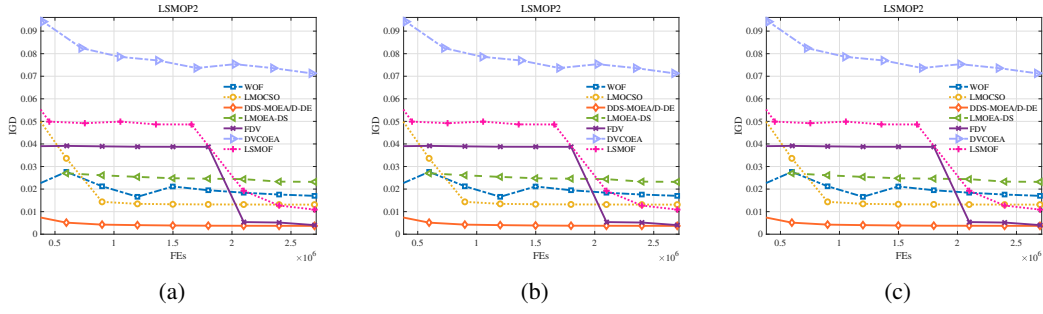


图 3-6 当决策变量为 200,  $MaxFEs$  设置为  $15\,000 \times 200$  时, DDS-MOEAD-DE 和其它六个算法在三个测试问题上的中位数 IGD 曲线

Fig. 3-6 The curve of median IGD values obtained by DDS-MOEAD-DE and other six algorithms on three test problems when the number of decision variables is 200 and  $MaxFEs$  is set to  $15\,000 \times 200$

### 3.5.5 与其它算法在基准测试问题上的比较

在这一小节中,我们以 DDS-MOEAD-DE 为代表,并与七个最先进的大规模 MOEA 进行比较,以深入考察 DDS 在求解 LSMOPs 方面的性能。这七种比较算法是 WOF、LMOCSO、MOEA/DVA、LMOEA-DS、LSMOF、FDV 和 DVCOEA。本节使用 LSMOP 测试套件的所有九个测试问题,决策变量数量分别设置为 200、500、1,000 和 2,000。表3-3提供了实验结果,其中灰色背景表示最佳效果。B/W/S 表示每个比较算法明显好于、差于或类似于 DDS-MOEAD-DE 的测试问题的数量。从表3-3可以看出,用灰色背景标记的区域清楚地表明 DDS-MOEAD-DE 的整体性能优于其它算法。具体来说,DDS-MOEAD-DE 算法在 36 个测试实例中的 18 个上表现最好,FDV 算法是所有比较算法中最具竞争力的,该算法在 36 个测试实例中的 11 个上获得了最好的结果。紧随其后的是 WOF,它在 36 个测试实例中的 5 个上运行得最好。LMOEA-DS 在 36 个测试实例中只有 1 个取得最佳结果。LMOCSO、LSMOF 和 DVCOEA 在任何测试问题上都没有达到最佳结果。最差的是 MOEA/DVA,它只在决策变量数为 200 时才返回结果。这是因为 MOEA/DVA 需要在变量分析上花费大量的  $FEs$ ,随着决策变量数量的增加,用于变量分析的  $FEs$  数量急剧增加。当变量个数等于或大于 500 时,变量分析所需的  $FEs$  远远超过预设的最大  $FEs$ 。因此,在本研究中,当变量个数等于或大于 500 时,MOEA/DVA 算法由于计算成本问题,无法完成求解。根据 Wilcoxon 秩和检验, B/W/S 排序显示,在 36 个测试实例中, DDS-MOEAD-DE 在 26、36、36、30、31、18 和 35 个测试实例上分别显著优于 WOF、LMOCSO、MOEA/DVA、LMOEA-DS、LSMOF、FDV 和 DVCOEA。值得注意的是,除 LSMOP1、LSMOP5 和 LSMOP9 外,其余六个测试问题的决策变量之间都存在一定程度的联系。然

表 3-3 DDS-MOEA/D-DE 和七种最新算法在 LSMOP 测试问题上的结果

Tab. 3-3 Results of DDS-MOEA/D-DE and seven state-of-the-art algorithms on LSMOP test problems

D	Problem	WOF	LMOCSSO	MOEA/DVA	LMOEA-DS	LSMOF	FDV	DVCOEA	DDS-MOEA/D-DE
200	LSMOP1	7.5445e-2 (2.76e-2)-	2.7316e-1 (1.01e-1)-	1.7616e-1 (8.14e-3)-	1.7467e-1 (2.7034e-3)-	5.0632e-1 (2.75e-2) -	6.1823e-3 (3.30e-3) -	1.4007e+0 (2.23e-1)-	4.5752e-3 (1.65e-4)
	LSMOP2	1.7095e-2 (3.29e-3)-	8.9098e-2 (1.87e-3)-	1.3737e-1 (6.32e-4)-	2.556e-2 (3.3524e-4)-	3.3261e-2 (1.69e-3) -	6.7115e-3 (8.67e-4) -	9.6101e-2 (3.39e-3)-	5.9729e-3 (5.73e-4)
	LSMOP3	7.1660e-1 (5.51e-2)-	1.2227e+0 (7.10e-1)-	3.9493e+0 (9.50e-1)-	1.5032e+0 (1.4005e-2)-	7.4956e-1 (4.69e-2) -	7.0804e-1 (2.96e-4) -	2.3705e+1 (1.24e+1)-	6.4250e-1 (5.04e-2)
	LSMOP4	6.2329e-2 (1.09e-2)-	8.6251e-2 (1.61e-2)-	9.0682e-2 (4.42e-3)-	6.5684e-2 (7.1303e-3)-	9.4625e-2 (7.22e-3) -	1.0491e-2 (1.86e-3) -	2.0863e-1 (8.15e-3)-	8.4819e-3 (9.38e-4)
	LSMOP5	1.7378e-1 (1.08e-1)-	3.7231e-1 (5.41e-3)-	4.4549e-1 (3.29e-2)-	2.2287e-1 (2.3273e-2)-	7.4209e-1 (0.00e+0) -	6.7195e-3 (2.99e-4) -	2.1174e+0 (1.93e+0)-	7.8604e-3 (3.00e-3)
	LSMOP6	3.2898e-1 (6.94e-2)+	7.6958e-1 (1.41e-2)-	2.7169e+0 (1.78e+0)-	3.198e-1 (5.0208e-2)+	3.5724e-1 (3.54e-3) -	4.9543e-1 (1.86e-1) +	8.8635e-1 (6.60e-2)+	6.2252e-1 (1.18e-3)
	LSMOP7	1.2622e+0 (2.02e-1)+	3.5157e+0 (7.88e-1)-	1.0312e+2 (1.74e+1)-	1.4835e+0 (1.5901e-2)+	1.4674e+0 (1.57e-3) +	1.1993e+0 (1.44e-1) +	3.7944e+1 (2.79e+1)-	1.5839e+0 (3.83e-1)
	LSMOP8	8.8848e-2 (9.07e-3)-	6.3831e-2 (1.60e-3)-	2.5719e-1 (9.73e-3)-	2.2375e-1 (4.2712e-2)-	7.4209e-1 (0.00e+0) -	2.7862e-2 (8.68e-3) -	8.0034e-1 (7.40e-2)-	2.3858e-2 (1.20e-2)
	LSMOP9	7.1272e-1 (1.26e-1)-	3.9733e-1 (1.18e-2)-	4.2007e-1 (2.91e-2)-	8.1004e-1 (1.2413e-16)-	8.1004e-1 (1.24e-16) -	1.9515e-1 (2.50e-1) -	1.2277e+1 (2.42e+0)-	1.0278e-2 (2.36e-3)
500	LSMOP1	7.1998e-2 (4.11e-2)-	3.0271e-1 (4.16e-2)-	/	1.8343e-1 (2.7011e-3)-	5.8762e-1 (2.29e-2) -	4.7645e-3 (1.32e-4) -	8.8893e+0 (4.82e+0)-	5.0496e-3 (2.92e-4)
	LSMOP2	1.4295e-2 (2.88e-3)-	4.1570e-2 (4.81e-4)-	/	1.3162e-2 (2.7439e-4)-	2.4879e-2 (7.44e-4) -	4.6193e-3 (1.87e-4) +	5.5100e-2 (3.36e-3)-	5.3491e-3 (8.93e-4)
	LSMOP3	1.3952e+0 (2.44e-1)-	1.6336e+0 (1.08e+0)-	/	1.5548e+0 (5.4354e-3)-	1.3414e+0 (3.04e-1) -	7.0796e-1 (8.98e-5) -	1.1389e+3 (1.08e+3)-	6.3276e-1 (6.99e-2)
	LSMOP4	3.8158e-2 (2.97e-3)-	5.7777e-2 (1.53e-3)-	/	3.7403e-2 (6.4163e-4)-	8.0131e-2 (3.01e-3)-	5.4468e-3 (3.60e-4)≈	1.2295e-1 (3.21e-2) -	5.6923e-3 (4.63e-4)
	LSMOP5	1.9564e-1 (8.25e-2)-	4.4751e-1 (1.06e-1)-	/	3.4057e-1 (8.0731e-2)-	7.4209e-1 (0.00e+0) -	6.7195e-3 (2.99e-4) +	2.1174e+0 (1.93e+0)-	6.4069e-2 (1.07e-1)
	LSMOP6	2.2065e-1 (6.59e-2)+	7.6275e-1 (2.86e-3)-	/	3.1693e-1 (4.6391e-3)+	3.1930e-1 (6.12e-4) +	7.4346e-1 (1.99e-2) -	8.1486e-1 (2.07e-3)-	6.5223e-1 (3.87e-4)
	LSMOP7	1.2760e+0 (1.57e-1)≈	2.6957e+0 (4.45e-1)-	/	1.4963e+0 (1.0023e-2)+	1.4960e+0 (2.74e-3) ≈	1.3563e+0 (1.32e-1) ≈	8.0039e+4 (6.61e+3)-	1.4994e+0 (3.28e-1)
	LSMOP8	8.1816e-2 (2.18e-2)-	3.4797e-2 (1.26e-3)-	/	4.0502e-1 (1.2188e-2)-	7.4209e-1 (0.00e+0) -	1.0723e-2 (3.64e-3) -	1.9542e+1 (6.86e-1)-	1.0424e-2 (2.56e-4)
	LSMOP9	4.2123e-1 (2.86e-1)-	1.9185e-1 (3.83e-3)-	/	5.2424e-1 (1.5757e-1)-	8.0890e-1 (1.29e-4) -	9.7081e-3 (9.46e-5) -	5.8186e+1 (7.96e-1)-	7.6670e-3 (1.49e-4)
1 000	LSMOP1	6.5536e-2 (5.45e-3)≈	3.4225e-1 (6.21e-2)-	/	1.9529e-1 (4.1311e-3)-	5.9656e-1 (2.48e-2)-	5.1435e-3 (3.74e-4) +	1.1112e+1 (2.05e-1)-	3.6556e-2 (3.72e-2)
	LSMOP2	9.3541e-3 (1.61e-3)-	2.2663e-2 (2.99e-4)-	/	8.0129e-3 (1.6447e-4)-	1.9001e-2 (3.59e-4)-	4.2020e-3 (3.31e-4)≈	3.7271e-2 (3.59e-3)-	4.5588e-3 (1.93e-4)
	LSMOP3	1.4023e+0 (4.12e-1)-	2.7922e+0 (1.84e+0)-	/	1.5689e+0 (9.7826e-4)-	1.5706e+0 (1.86e-4)-	7.0788e-1 (1.12e-4)+	1.2048e+3 (6.67e+2)-	6.3556e-1 (8.54e-2)
	LSMOP4	2.2589e-2 (3.30e-3)-	3.6656e-2 (1.29e-3)-	/	2.1349e-2 (5.2629e-4)-	4.4081e-2 (3.36e-3)-	4.3547e-3 (1.81e-4)≈	7.8355e-2 (4.68e-4)-	4.6618e-3 (1.88e-4)
	LSMOP5	2.9604e-1 (8.04e-2)≈	5.5002e-1 (8.37e-2)-	/	5.8765e-1 (3.8686e-2)-	7.4209e-1 (0.00e+0)-	9.3417e-3 (5.43e-4)+	2.3708e+1 (2.54e-1)-	3.1017e-1 (3.46e-3)
	LSMOP6	2.1344e-1 (5.50e-2)+	7.6107e-1 (2.82e-3)-	/	3.131e-1 (3.8863e-3)+	3.1084e-1 (3.26e-3)+	5.4742e-1 (1.17e-1)+	1.1414e+3 (2.11e+3)-	7.42960-1 (1.45e-4)
	LSMOP7	1.2962e+0 (1.84e-1)≈	2.1656e+0 (4.10e-1)-	/	1.5096e+0 (4.3109e-3)-	1.5056e+0 (3.06e-4)-	1.2946e+0 (1.99e-1)≈	8.6811e+4 (2.82e+3)-	1.3227e+0 (2.11e-1)
	LSMOP8	7.6388e-2 (8.66e-3)-	2.6333e-2 (4.56e-4)-	/	5.1815e-1 (1.0732e-1)-	7.4209e-1 (0.00e+0)-	8.6629e-3 (1.31e-3)-	2.0217e+1 (2.12e-1)-	7.92100-3 (2.76e-4)
	LSMOP9	3.5152e-1 (2.05e-1)-	9.2598e-2 (1.50e-3)-	/	4.9286e-1 (1.6248e-1)-	8.0650e-1 (1.34e-4)-	3.0342e+1 (4.29e+1)-	4.7220e+1 (2.59e+1)-	7.4033e-3 (1.04e-4)
2 000	LSMOP1	1.0253e-1 (3.12e-2)-	9.6296e-2 (7.79e-2)-	/	1.9381e-1 (1.0003e-3)-	6.2002e-1 (7.51e-3) -	4.4902e-3 (9.21e-5) +	9.0650e+0 (4.94e+0)-	5.3643e-2 (8.64e-2)
	LSMOP2	6.5095e-3 (1.02e-3)-	1.1809e-2 (2.11e-4)-	/	5.8594e-3 (2.3517e-4)-	1.2923e-2 (6.97e-4) -	3.7627e-3 (3.08e-5) +	2.1450e-2 (2.40e-3)-	5.7729e-3 (6.02e-3)
	LSMOP3	1.1638e+0 (4.03e-1)-	2.9405e+0 (1.37e+0)-	/	1.5733e+0 (2.9275e-4)-	1.5754e+0 (8.47e-5) -	7.0747e-1 (5.76e-5) -	1.4778e+3 (1.72e+3)-	6.0540e-1 (9.14e-2)
	LSMOP4	1.5123e-2 (4.99e-3)-	1.2352e-2 (4.77e-4)-	/	1.2297e-2 (7.2346e-05)-	3.2589e-2 (1.67e-3) -	3.7985e-3 (4.89e-5) ≈	4.4822e-2 (3.50e-4)-	3.6610e-3 (5.63e-5)
	LSMOP5	2.7153e-1 (1.04e-1)-	7.9170e-1 (7.05e-2)-	/	5.2796e-1 (1.9272e-1)-	7.4209e-1 (0.00e+0) -	7.7105e-3 (1.97e-4) +	2.4228e+1 (3.68e-1)-	1.1433e-2 (4.11e-4)
	LSMOP6	1.7865e-1 (4.14e-3)+	7.5738e-1 (2.83e-13)-	/	3.2856e-1 (1.5215e-2)+	3.0546e-1 (2.35e-3) +	6.4287e-1 (1.44e-1) ≈	3.7967e+2 (6.45e+2)-	7.4228e-1 (1.64e-5)
	LSMOP7	1.1374e+0 (2.86e-1)≈	9.0619e+1 (2.81e+2)-	/	1.5131e+0 (6.67e-4)-	1.5111e+0 (1.56e-4) -	1.2895e+0 (2.31e-1) ≈	9.0262e+4 (1.39e+3)-	1.1896e+0 (8.18e-2)
	LSMOP8	5.5727e-2 (7.02e-3)-	9.6835e-3 (2.60e-4)-	/	5.0085e-1 (1.4044e-1)-	7.4209e-1 (0.00e+0) -	6.5827e-3 (5.87e-4) -	2.0590e+1 (2.59e-1)-	5.8422e-3 (1.05e-4)
	LSMOP9	1.9302e-1 (2.25e-1)-	3.4114e-2 (1.25e-3)-	/	2.2621e-1 (2.2572e-2)-	8.0364e-1 (4.46e-4) -	7.0526e-3 (6.98e-6) -	4.8718e+1 (2.68e+1)-	6.6123e-3 (6.03e-5)
B/W/S		5/26/5	0/36/0	1/36/0	6/30/0	4/31/1	10/18/8	1/35/0	

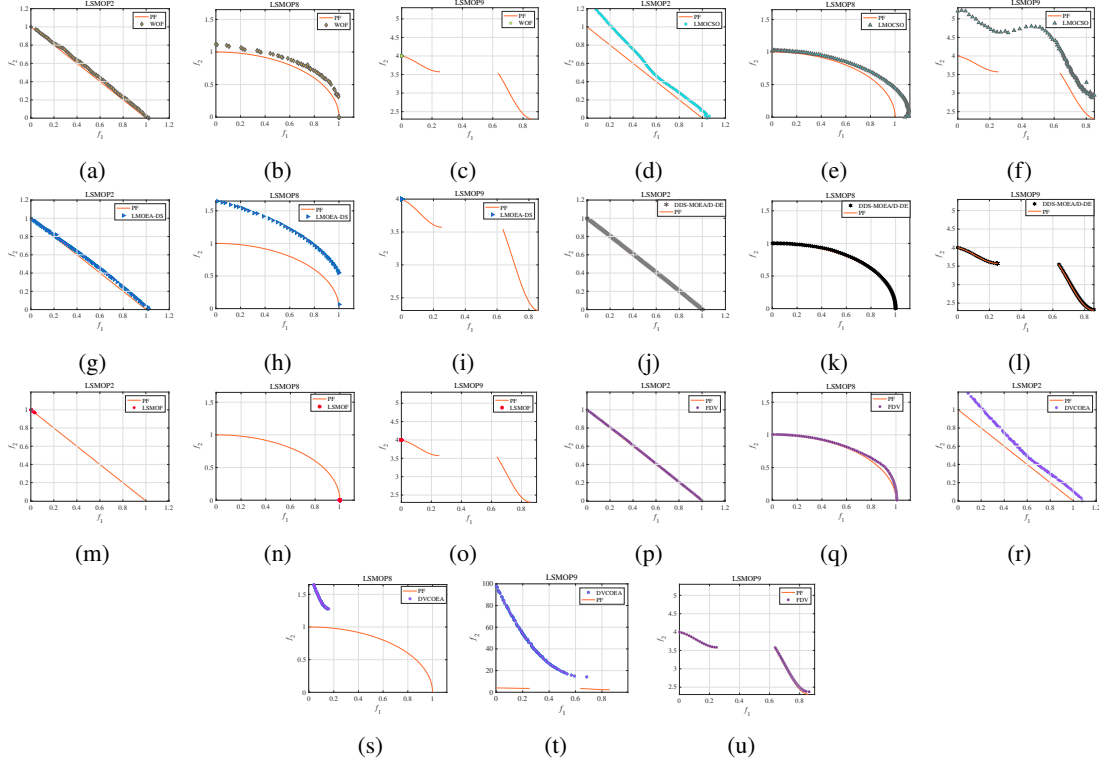


图 3-7 由 WOF、LMOCSO、LMOEA-DS、LSMOF、FDV、DVCOEA 和 DDS-MOEA/D-DE 对具有 200 个决策变量的 LSMOP2、LSMOP8 和 LSMOP9 获得的 PF

Fig. 3-7 The PF obtained by WOF, LMOCSO, LMOEA-DS, LSMOF, FDV, DVCOEA, and DDS-MOEA/D-DE on LSMOP2, LSMOP8, and LSMOP9 with 200 decision variables

而,在这些测试问题上的 DDS 实验仍然取得了令人满意的结果。图3-7在 LSMOP2、LSMOP8 和 LSMOP9 上绘制了除 MOEA/DVA 以外的每种算法取得的  $PF$ , 决策变量数为 200, 直观地展示了实验结果。从该图中, DDS-MOEA/D-DE 在所有三个测试问题上对  $PF_{True}$  进行了最好的覆盖。基于以上分析, DDS-MOEA/D-DE 在解质量方面具有最好的整体性能, 这一点间接地表明 DDS 可以提高现有 MOEA 求解 LSMOP 的性能。

表 3-4 GAN-MOEA/D-DE 和 DDS-MOEA/D-DE 三个变种在 LSMOPs 上的 IGD 平均值  
Tab. 3-4 IGD mean values of three variants of GAN-MOEA/D-DE and DDS-MOEA/D-DE on LSMOP

D	Algorithms	LSMOP1	LSMOP2	LSMOP3	LSMOP4	LSMOP5	LSMOP6	LSMOP7	LSMOP8	LSMOP9
500	GAN-MOEA/D-DE	1.83e-1	2.42e-2	1.51e+0	4.37e-2	6.41e-1	3.66e-1	1.50e+0	6.54e-1	5.84e-1
	DDS-MOEA/D-DE	1.76e-1	1.15e-2	1.47e+0	2.98e-2	5.60e-1	6.67e-1	7.44e+0	1.48e-1	9.30e-2
1 000	GAN-MOEA/D-DE	3.35e-1	1.33e-2	1.58e+0	2.55e-2	6.88e-1	3.84e-1	5.51e+0	7.07e-1 (7.18e-7)	6.26e-1 (3.78e-4)
	DDS-MOEA/D-DE	2.83e-1	7.98e-3	1.51e+0	1.73e-2	7.42e-1 (1.36e-16)	7.57e-1	5.72e+1	7.35e-1	5.31e-1
2 000	GAN-MOEA/D-DE	5.28e-1	8.48e-3	1.92e+0	1.73e-2	6.72e-1	5.37e-1	4.23e+0	6.63e-1	7.13e-1
	DDS-MOEA/D-DE	4.36e-1	6.22e-3	1.59e+0	1.21e-2	7.42e-1 (1.36e-16)	7.26e-1	5.12e+2	7.42e-1	5.54e-1

最近, 部分研究引入了深度学习来解决 LSMOPs。GAN-LMEF 提出了一种基于生成对抗网络的流形插值框架来学习搜索空间的流形结构, 用于生成高质量

的解决方案并提高现有 MOEA 解决 LSMOP 的性能。为了对 DDS 和 GAN-LMEF 进行比较, 本研究仍然以 DDS-MOEA/D-DE 作为代表, 在 LSMOP1-9 上进行进一步的实验。其中,  $MaxFES$  设置为 100 000。表3-4提供了结果。从表3-4中, 用灰色背景标记的区域表明, DDS-MOEA/D-DE 在许多测试用例中优于 GAN-MOEA/D-DE。具体来说, DDS-MOEA/D-DE 在 27 个测试实例中的 17 个上获得了最好的结果。GAN-MOEA/D-DE 在 27 个测试用例中的 10 个上取得了最好的结果。因此, 即使与最新的基于 GAN 的算法框架相比, 所提出的方法也能更有效地解决 LSMOPs。另外, 从表中的对比可以看出, DDS-MOEA/D-DE 在处理 LSMOP1、LSMOP2、LSMOP3、LSMOP4、LSMOP9 方面表现非常好, 而 GAN-MOEA/D-DE 似乎是擅长处理 LSMOP5、LSMOP6、LSMOP7、LSMOP8。这是因为, 在 LSMOP5、LSMOP6、LSMOP7 和 LSMOP8 中, 每个目标问题同时与两个相邻变量组具有重叠相关性。基于 GAN 的算法更擅长处理此类问题, 这也显示了基于 GAN 的算法在 LSMOP 上的潜力。

表 3-5 TREE1-6 的问题设置  
Tab. 3-5 Problem settings for TREE1-6

Problems	D	M	Objectives	Constraints	$MaxFES$
TREE1	6 000	2	2	3	20 000
TREE2	6 000	2	2	3	40 000
TREE3	6 000	2	2	3	2 000 000
TREE4	12 000	2	2	4	1 000 000
TREE5	60 000	2	2	4	600 000
TREE6	120 00	3	3	6	4000 000

### 3.5.6 与其它算法在实际应用问题上的比较

为了进一步研究 DDS 的性能, 本小节以 DDS-MOEA/D-DE 为例, 测试了其在时变比率误差估计问题<sup>[2]</sup>和稀疏对抗攻击问题<sup>[2]</sup>两个实际问题上的性能。其中, 时变比率误差估计问题 (Time-varying Ratio Error Estimation, TREE) 是指电压互感器估计中的一类比率误差漂移故障问题。该问题可以被建模为具有不等约束的六个多目标优化问题, 分别记为: TREE1-TREE6。其中, TREE1、TREE2 和 TREE3 包括两个目标和三个约束; TREE4 和 TREE5 包括两个目标和四个约束; TREE6 则包括三个目标和六个约束。由于 TREE 问题的决策变量数量从 6 000 到 300 000 个不等, 因此 TREE 问题无疑是 LSMOP 的一种。本研究依旧将 MOEA/D-DE 嵌入到 DDS 中, 并与 WOF、LMOCSSO、MOEA/DVA、LMOEA-DS、LSMOF、FDV 和 DVCOEA 等七种算法进行了比较。表3-5显示了实验中使

表 3-6 DDS-MOEA/D-DE 和七种算法在 TREE 问题上的结果

Tab. 3-6 The results of DDS-MOEA/D-DE and seven algorithms on TREE problem

Problem	WOF	LMOCSO	MEOAD/DVA	LMOEA-DS	LSMOF	FDV	DVCOEA	DDS-MOEA/D-DE
TREE1	1.9231e+1	1.9473e+1	—	1.7467e+1	1.7306e+1	—	—	2.4671e-1
TREE2	8.6096e+1	8.8132e+1	—	8.7698e+1	4.3457e+2	—	—	8.2064e+0
TREE3	2.3491e+1	2.3966e+1	—	9.6343e+0	2.1981e-1	1.1993e+2	—	5.6023e-2
TREE4	2.4000e+1	2.3999e+1	—	2.1744e+1	1.1672e+2	1.1239e+2	—	1.4942e-2
TREE5	3.5644e+1	3.6932e+1	—	1.8663e+1	1.8339e+2	1.8517e+2	—	2.8815e-1

用的问题设置和 TREE1-TREE5 的结果。需要注意的是, TREE6 是一个非常复杂的问题, 导致所有算法都不能得到任何可接受的解, 因此 TREE6 的结果未在表中列出。在表3-6中, 用灰色背景标记的区域表明, DDS-MOEA/D-DE 不但在 TREE1-TREE5 上都搜索到了可行解, 而且在这些测试实例中的表现都优于其它算法, 并且至少比其它算法优越一个数量级。而 MOEAD/DVA 和 DVCOEA 没有在任何测试实例上搜索到可行解。此外, 尽管 FDV 在 LMSOP1-9 测试函数集上表现较好, 却没有在 TREE1 和 TREE2 上搜索到可行解, 这些结果表明所提出的 DDS 在解决 TREE 问题上具有显著的优势。

另一方面, 稀疏对抗性攻击问题的目的是找到最小的扰动, 以降低分类器的性能。稀疏对抗性攻击的研究对于自动驾驶的安全性、身份认证、深度学习的可解释性以及深层神经网络的几何结构都具有重要的意义<sup>[2]</sup>。稀疏对抗性攻击问题可以表示为具有两个目标的 MOP 问题, 其形式如下:

$$\begin{aligned}
& \min \quad f_1(\epsilon) = \|\epsilon\|_2 \\
& \quad \quad f_2(\epsilon) = \|\epsilon\|_0 \\
& \text{s.t.} \quad c_1(\epsilon) = F(\mathbf{x} + \epsilon, t) - \max_{i \neq t} F(\mathbf{x} + \epsilon, i) < 0, \\
& \quad \quad t = \operatorname{argmax}_i F(\mathbf{x}, i), \\
& \quad \quad \epsilon \in [-\mathbf{x}, 1 - \mathbf{x}]^D
\end{aligned} \tag{3-14}$$

其中,  $\epsilon$  是要搜索的扰动向量,  $\epsilon_2$  和  $\epsilon_0$  分别表示  $\epsilon$  的  $\ell_2$  和  $\ell_0$  范数,  $\mathbf{x}$  是在行向量处调整为一维的输入图像, 并且  $\mathbf{x}$  的形状与相同。  $F(\mathbf{x}, i)$  是输入  $\mathbf{x}$  的给定神经网络的第  $i$  个输出,  $t$  是对应于  $\mathbf{x}$  的分类标签。稀疏对抗攻击问题是发现不可感知的图像扰动, 以便将添加这些扰动的图像分类到错误的标签。用  $\ell_0$  范数、 $\ell_1$  范数和成功率来评价稀疏对抗性攻击问题的解质量。具体地说,  $\ell_0$  范数衡量的是扰动向量中非零元素的数量。  $\ell_1$  范数衡量扰动向量中元素的绝对值之和的大小,

用于评估微不足道的扰动，其定义如下：

$$\ell_1 = \sum_{i=1}^n |\epsilon_i| \quad (3-15)$$

将算法发现的扰动添加到原始图像中。这些扰动图像然后被用作分类器的输入，如果分类器输出错误的标签，我们可以说攻击成功。成功率可以自然地定义为：

$$\text{Success rate} = \frac{\text{Number of samples successfully attacked}}{\text{Number of total samples}} \quad (3-16)$$

本研究使用了 CIFAR-10 图像分类数据集，每个图像的大小为 [3, 32, 32]，因此决策变量的数量达到 3072，属于一类 LSMOP。为了解决这个问题，我们在 DDS 中嵌入了 DCEA-ISA 算法，这是一种最新进化算法，专门用于解决稀疏对抗攻击问题。除此之外，还有九种比较算法：C&W- $L_2$ <sup>[2]</sup>、BIM<sup>[2]</sup>、OPA<sup>[2]</sup>、NT-JSMA<sup>[2]</sup>、SAP<sup>[2]</sup>、Corner Search<sup>[2]</sup>、QL<sup>[2]</sup>、MOEA-APGA<sup>[2]</sup>和 DCEA-ISA<sup>[2]</sup>。

表 3-7 稀疏对抗攻击问题的 DDS-MOEA/D-DE 和 9 种算法的结果

Tab. 3-7 The results of DDS-MOEA/D-DE and nine algorithms on sparse adversarial attack problem

Metrics	Methods									
	C&W- $L_2$	BIM	OPA	NT-JSMA	SAP	Corner Search	QL	MOEA-APGA	DCEA-ISA	DDS-DCEA-ISA
mean of $L_0$	1024.00	1015.95	1.00	70.83	55.49	22.64	1020.90	706.90	44.33	41.7
std of $L_0$	(0.04)	(12.79)	(0.00)	(53.39)	(108.22)	(15.15)	(10.90)	(7.49)	(36.70)	30.4
mean of $L_2$	0.32	0.40	0.86	0.41	1.75	0.64	1.83	36.90	0.31	0.2
std of $L_2$	(0.12)	(0.03)	(0.13)	(0.18)	(2.66)	(0.26)	(10.56)	(10.19)	(0.15)	0.2
Success rate	100%	98.8%	96.6%	96.3%	87.5%	77.7%	47.5%	98.9%	100%	100%



图 3-8 通过 DDS-DCEA-ISA 成功攻击的示例

Fig. 3-8 Examples of the successful attacks through DDS-DCEA-ISA

这九种算法都被用于进行比较，并且所有参数设置也遵循同样的研究方法。表3-7列出了比较结果，并引用了其它九种比较算法的结果。从表3-7可以看出，在 CIFAR-10 上，DDS-DCEA-ISA 取得了 100% 的成功率。在  $\ell_0$  方面，尽管 OPA 和 CornerSearch 方法获得了更稀疏的扰动，但它们也牺牲了成功率。在  $\ell_2$  方面，DDS-DCEA-ISA 方法获得了比其它方法更好的扰动效果。这表明 DDS 在解决稀疏对抗性攻击问题方面具有显著的优异性能。图3-8显示了通过 DDS-DCEA-ISA 成功攻击的六个示例。具体来说，最左侧的一栏显示了原始图像和正确的标签，中间一列是 DDS-DCEA-ISA 搜索到的最佳扰动，这些干扰几乎是人眼难以察觉的。添加最佳扰动后，得到最右侧列的扰动图像。然后，分类器给出了完全错误的结果（红色文本），例如将“猫”分类为“狗”。这些结果清楚地表明 DDS 在解决稀疏对抗性攻击问题上的卓越表现。

### 3.6 本章小结

本章提出了一种双分解分组策略（DDS），该策略一定程度上缓解了决策变量随机分组策略导致相互关联的决策变量被划分为不同组的问题，同时又能提高 LSMOP 上的多目标进化算法的求解质量和求解效率。DDS 采用了两种不同的分解方法来降低求解 LSMOP 的难度。其中外分解利用滑动窗口将决策变量划分为相互重叠的组，可以缓解相互关联的决策变量被分为不同组而影响性能的问题，内分解根据全局参考点和全局参考向量对决策变量组对应的 MOPs 进一步分解，进一步提高求解性能。此外，采用 BCD 策略对双分解得到的每个小规模 SOP 进行优化，增强了收敛能力。在 LSMOP 基准测试集上的实验结果与先进的算法相比，验证了 DDS 的优异性能，在两个实际问题上的实验结果证明了 DDS 的实际应用潜力。





## 第4章 面向大规模多目标优化问题的保守序列分组策略

针对决策变量分析分组策略计算效率低下的问题，本章受自然进化中保守序列的启发，提出了一种面向大规模多目标优化问题的保守序列分组策略（A Conserved Sequence Grouping Strategy for Large-scale Multi-objective Optimization Problems, CS）。

### 4.1 保守序列分组策略思想

保守序列分组策略的基本思想为：在保守序列分析过程中，对保守序列进行提取，根据决策变量的历史行为将其分类为保守序列和非保守序列。由于在算法搜索过程中，决策变量在决策空间总是向着帕累托最优解方向更新的，因此根据决策变量更新的历史行为可以对决策变量进行较为合理的划分。其中保守序列被定义为种群中更新差异变化较小的决策变量；而非保守序列被定义为种群中更新差异变化较大的的决策变量。这种划分方式不需要进行函数评价，因此可以提高效率。然后，进入特异进化过程，该过程针对保守序列和非保守序列的采取不同的更新策略。具体地，保守序列直接沿着其最近一次的更新方向直接进行更新，而非保守序列则使用常规的进化过程进行优化，对具有不同特点的决策变量采取不同的更新策略，能够有效地缓解算法在庞大的决策空间中盲目搜索的问题，从而提高求解质量。

算法4-1提供了嵌入 CS 的 MOEA 的基本框架。其中  $A$  表示嵌入的 MOEA；输入为：学习率 ( $\eta$ )、衰减权重 ( $\gamma$ )、保守长度 ( $C_l$ )、比率 ( $r$ )、种群大小 ( $N$ ) 和决策变量个数 ( $D$ )。具体的实现过程为：从随机初始化开始，用  $P_c$  表示当前种群（第 1 行）。CS-A 的迭代由函数评价次数  $FES$  和最大函数评价次数  $MaxFES$  控制（第 3-20 行）。首先，使用种群  $P_p$  备份当前种群  $P_c$ （第 4 行），然后通过  $A$  对  $P_c$  进行优化，并得到下一代种群  $P_n$ （第 5 行）。如果  $FES$  小于或等于  $MaxFES \times r$ （第 6 行），则激活 CS。此时，使用  $P_c$  和  $P_p$  计算当前方向  $\Delta_c$ （第 8 行），并更新特异进化参数  $\eta$ （第 9 行）。然后，进行决策变量分析，将决策变量划分为保守序列和非保守序列（第 9-12 行），并使用特异进化来优化  $P_c$ （第 13-15 行）。最后，如果 CS 被激活，则将  $P_n$  赋值为  $P_c$ （第 18 行）；否则， $P_c$  直接由  $A$  优化得到的种群赋值（第 21 行）。

---

**算法 4-1: CS-A**


---

输入: 学习率 ( $\eta$ ), 衰减权重 ( $\gamma$ ), 保守长度 ( $C_l$ ), 比率 ( $r$ ), 种群大小 ( $N$ ), 决策变量个数 ( $D$ )

输出: 最终种群  $P$

```

1:  $P_c \leftarrow$  初始化种群 ( $N, D$ )
2:  $FES \leftarrow 0$ 
3: while  $FES < MaxFES$  do
4:    $P_p \leftarrow P_c$ ,
5:    $P_n \leftarrow A(P_p)$ 
6:   if  $FES \leq MaxFES \times r$  then
7:     /*CS 过程开始 */
8:      $\Delta_c = P_c - P_p$ 
9:      $\eta \leftarrow$  学习率衰减 ( $\eta, FES, MaxFES, r$ )
10:    /* 保守序列分析过程开始 */
11:    conserved_idx, non-conserved_idx  $\leftarrow$  保守分析 ( $P_p, P_n, \Delta_c$ )
12:    /* 保守序列分析过程结束 */
13:    /* 特异进化过程开始 */
14:     $P_c \leftarrow$  特异进化 (conserved_idx, non-conserved_idx,  $P_c, C_l$ )
15:    /* 特异进化过程结束 */
16:    /*CS 过程结束 */
17:  else
18:     $P_c \leftarrow P_n$ 
19:  end if
20: end while
21:  $P \leftarrow P_c$ 

```

---

## 4.2 保守序列分组策略描述

保守序列分组策略包含两个关键部分: 保守序列分析和特异进化。首先, 在保守序列分析中, 根据决策变量的历史更新方向, 提取决策变量的保守序列和非保守序列, 其中保守序列中的决策变量的在最近的两次更新中变化方向变化较小; 而非保守序列在最近的两次更新中的方向变化较大。在完成保守序列分析后, 分别对保守序列和非保守序列采取不同的优化措施, 具体来说, 保守序列根据其最近的更新方向直接更新; 非保守序列则采用常规的进化算法更新。下面将详细描述 CS 的上述两个过程。

### 4.2.1 保守序列分析

本小节根据进化算法的特点, 从自然进化中保守序列的定义、作用和进化算法的机理出发, 先给出进化算法中保守序列的定义, 然后给出保守序列分析的过程。

保守序列在疾病风险分析和多肽药物设计中有广泛应用。在生物学中, 保守

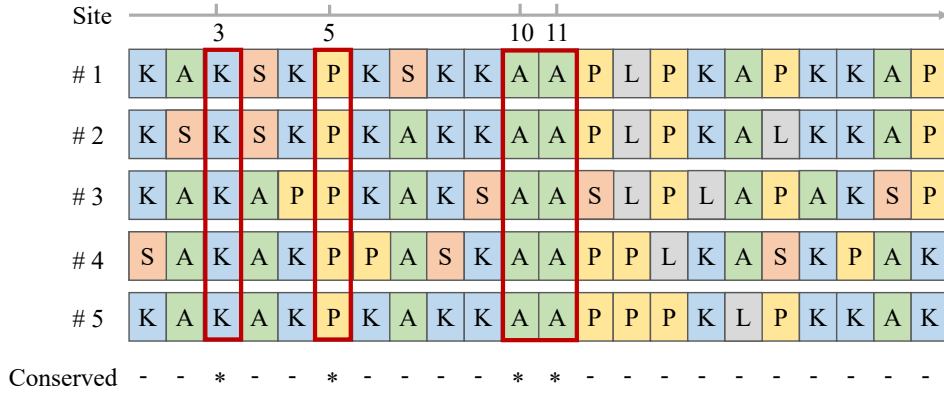


图 4-1 自然进化中保守序列的一个简单例子

Fig. 4-1 A simple example of the conserved sequence in natural evolution

序列是指在性质相似或相同的蛋白质、核酸或多糖序列，在进化过程中被保留了下来<sup>[7]</sup>。许多研究表明，保守序列包含关于物种进化的有价值信息，因此随着时间的推移基本保持不变。研究人员通过分析生物体的蛋白质或氨基酸序列来寻找关于物种进化的有用信息。图4-1提供了五个肽链上的保守序列的简单图例，每条链由 21 个氨基酸组成，不同颜色的长方形代表不同类型的氨基酸。最后一行显示了分析结果，其中破折号 (-) 表示属于非保守序列的位置；星号 (\*) 表示属于保守序列的位置。在本例中，保守序列包括点位 3、点位 5、点位 10 和点位 11（图中用红线包裹的区域），因为这些位置在不同的肽链中具有相同的氨基酸。

在进化算法的过程中，种群朝着全局最优的方向进化，个体进化方向与 PS 密切相关。如果我们能够分析种群中进化最为稳定的部分，就可以利用这些信息对不同决策变量采取有针对性的优化策略。因此，受上述自然进化中保护序列的启发，本章提出了进化过程中保守序列的定义。

进化算法中保守序列的定义为：在最近的两次连续更新中，变化差异较小的决策变量，而非保守序列为最近的两次连续更新中，变化差异较大的决策变量。具体地，对于  $P_c$  中的个体  $\mathbf{x}_c$ 、 $P_p$  中的个体  $\mathbf{x}_p$  和  $P_n$  中的个体  $\mathbf{x}_n$ ， $\mathbf{x}_p$  朝着  $\mathbf{x}_c$  更新的方向  $\Delta_c$  和  $\mathbf{x}_c$  朝着  $\mathbf{x}_n$  更新的方向  $\Delta_n$  为：

$$\Delta_c = \mathbf{x}_c - \mathbf{x}_p \quad (4-1)$$

$$\Delta_n = \mathbf{x}_n - \mathbf{x}_c \quad (4-2)$$

记  $x_i$  为  $\mathbf{x}_c$  中的第  $i$  个决定变量； $\Delta_{ci}$  为  $\Delta_c$  中的第  $i$  个方向变量； $\Delta_{ni}$  为  $\Delta_n$  中的

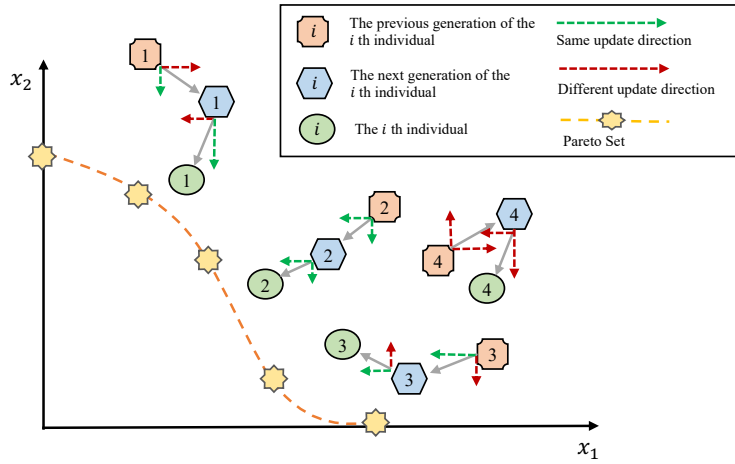


图 4-2 具有四个个体的种群的两次迭代

Fig. 4-2 Two iterations for a population with four individual

第  $i$  个方向变量,  $\mathbf{x}_i$  被认为属于保守序列当且仅当:

$$\delta_{\text{sgn}(\Delta_{ci}), \text{sgn}(\Delta_{ni})} = 1 \quad (4-3)$$

其中,  $\text{sgn}(\cdot)$  是符号函数, 其定义为:

$$\text{sng}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases} \quad (4-4)$$

$\delta_{\cdot, \cdot}$  是克罗内科函数, 其定义为:

$$\delta_{i,j} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad (4-5)$$

图4-2展示了一个包含四个个体的种群在两次迭代中的演化过程, 四个个体分别由数字 1 到 4 进行区分。其中, 每个个体都有两个决策变量。在该图中, 绿色圆圈代表最新一次迭代对应的个体, 蓝色六边形和橙色斑块分别代表先前两次迭代对应的个体。八角星和虚线的组合表示  $PS$ 。灰色箭头表示个体的更新方向, 红色虚线箭头和绿色虚线箭头则表示特异进化方向在不同决策变量维度上的分量, 即沿  $x_1$  和  $x_2$  的更新方向。红色虚线箭头表示两个不同的更新方向, 而绿色虚线箭头表示两个相同的更新方向。根据上述保守序列的定义不难看出, 这四个个体的保守序列分别为:  $\{x_2\}$ ,  $\{x_1, x_2\}$ ,  $\{x_1\}$ , 和  $\emptyset$ 。

为了个体解进化方便, 规定每个个体的保守序列长度一致, 都为  $C_l$ , 若个体

**算法 4-2: 保守序列分析**

输入: 种群 ( $P_p$ ), 种群 ( $P_n$ ), 方向 ( $\Delta_c$ ), 保守长度 ( $C_l$ )

输出: 保守序列索引集合 ( $conserved\_idx$ ), 非保守序列索引集合 ( $non\_conserved\_idx$ )

```

1:  $conserved\_idx \leftarrow \emptyset$ 
2:  $non\_conserved\_idx \leftarrow \emptyset$ 
3:  $\mathcal{I} \leftarrow \{1, 2, \dots, D\}$ 
4: 根据公式4-3提取保守序列, 记录索引 Cons。
5: for  $i \leftarrow 1 : N$  do
6:   if  $|Cons_i| < C_l$  then
7:     计算保守序列与保守长度的差值  $C \leftarrow C_l - |Cons_i|$ 
8:      $\Delta_s \leftarrow$  由小到大排列  $\Delta_i$ 
9:     将  $\Delta_s$  的前  $C$  个个体序号并入 Cons
10:  else
11:    在 Cons 的  $C_l$  处截断
12:  end if
13:   $conserved\_idx(i) \leftarrow \{Cons_i\}$ 
14:   $non\_conserve\_idx(i) \leftarrow \mathcal{I} - conserved\_idx(i)$ 
15: end for

```

的保守序列长度大于  $C_l$ , 其保守序列在  $C_l$  处被截断; 若个体的保守序列长度小于  $C_l$  时, 非保守序列中的决策变量按照最近更新程度的大小进行排序, 然后按从大到小的顺序补全保守序列。以图4-2为例, 当  $C_l$  设置为 1 时, 四个个体的保守序列分别为  $x_2, x_1, x_1$  和  $x_2$ 。其中, 第 2 个个体的保守序列在  $x_1$  处截断, 而第 4 个个体的保守序列由  $x_1$  填充。基于保守序列的分组思想可以在不消耗额外计算次数的情况下对决策变量进行快速分组, 这样可以节省计算资源用于优化。

算法4-2提供了保守序列分析的具体算法流程。首先, 将  $conserved\_idx$  和  $non\_conserved\_idx$  初始化为空序集 (第 1-2 行)。然后, 初始化有序记录集  $\mathcal{I}$  (第 3 行)。根据公式4-3, 提取具有不同长度的保守序列 (第 4 行)。当对应于个体的保守序列的长度小于  $C_l$  时, 它遍历对应于每个个体的保守序列 (第 5-13 行), 并借助个体的非保守序列完成保守序列 (第 6-9 行), 否则将保守序列截断 (第 10-11 行) 以满足长度要求。最后, 将处理后的保守序列保存到  $conserved\_idx$  中 (第 12 行)。同时, 将保守序列与  $\mathcal{I}$  之间的差集作为非保守序列, 并将其保存在  $non\_conserved\_idx$  (第 13 行)。

#### 4.2.2 特异进化策略

对保守序列和非保守序列采用不同的更新策略能够充分利用保守序列和非保守序列的特点, 避免了对保守序列进行更新时所需要的盲目搜索, 能够进一

步提高算法收敛速度并提高求解质量。对于属于保守序列的决策变量，它们在最近的两次更新中方向保持不变，有理由认为在下一次更新中仍然保持与前两次迭代的相同的更新方向。因此它们下一次的更新应该按照它们的历史更新方向进一步更新。这有助于减少个体在更新过程中无意义的随机搜索，并保留合理的优化方向，提高算法的整体效率。然而，重要的是要注意，保守序列的更新规则并不是固定的，可以根据所解决问题的具体特征进行修改或调整。在本研究中，使用的更新规则如下：

$$\begin{aligned}\Delta &= \mathbf{x}_n - \mathbf{x}_c \\ x_i &= x_i + \eta \times \Delta_i \\ \forall i &\in \{i | x_i \text{ 属于保守序列}\}\end{aligned}\tag{4-6}$$

其中， $\eta$  是控制属于保守序列的个体在其最后更新方向上更新的程度的更新率； $\eta$  是一个预置的超参数，在算法的不断迭代和种群逐渐接近  $PS$  的过程中，更新率应该逐渐降低以避免震荡。在本研究中，我们使用  $FES$  来控制算法的迭代次数，所以  $\eta$  的更新函数应该是关于  $FES$  的单调递减函数，如下所示：

$$\eta_{FES} := \eta_0 \times \exp\left(-\frac{FES^2 \times \ln(\eta_0 \times 10000)}{r^2 + MaxFES^2}\right)\tag{4-7}$$

其中， $\eta_i$  是算法在  $\eta = i$  时使用的  $Fes$  值； $\eta_0$  是预设值； $r$  是控制 CS 参与比例的超参数。对于属于非保守序列的决策变量，历史演化信息不具有参考性，因为它们最近的两次更新方向相反。因此对于属于非保守序列的决策变量，必须使用嵌入的 MOEA(A) 重新执行搜索。值得注意的是，大多数 MOEAs 的环境选择策略改变了个体在原始种群中的位置，这可能会影响 CS 中方向的计算。因此，在 CS-A 中，我们将 A 中的环境选择策略替换为基于分解的策略，以确保个体在种群中的位置保持不变。

### 4.3 时间复杂度分析

在本节中，我们将分析保守分析策略在的时间复杂度。从保守分析策略（参见算法4-2）的框架来看，保守分析的每一次迭代的主要计算成本来自两个部分：（1）计算保守序列（第 4 行）；（2）归一化保守序列（第 11 行）。其中， $D$  表示决策变量的个数， $N$  表示种群规模， $C_l$  表示保守长度。

（1）对于保守序列计算部分，提取保守序列需要通过计算两个大小为  $N \times D$  的种群相减得到方向矩阵。这一过程至少需要遍历种群中每个个体的每个决策

表 4-1 在具有 200 个决策变量的 LSMOP 测试集上的参数敏感性分析结果

Tab. 4-1 The parameter sensitivity result on LSMOP test suit with 500 decision variables

Parameters	Vlaues	Problems								
		LSMOP1	LSMOP2	LSMOP3	LSMOP4	LSMOP5	LSMOP6	LSMOP7	LSMOP8	LSMOP9
$C_l$	100	0.494	0.050	11.611	0.088	1.340	0.771	19.488	0.501	0.135
	200	0.046	0.046	10.864	0.075	0.084	0.766	4.456	0.050	0.045
	300	0.144	0.045	6.886	0.019	0.395	0.402	2.547	0.136	0.042
	400	0.256	0.041	11.956	0.087	0.041	0.381	3.124	0.057	0.328
	500	1.313	0.049	12.649	0.095	2.580	0.767	53.827	0.757	0.637
$r$	0.4	0.357	0.045	2.920	0.087	0.409	0.763	6.265	0.046	0.404
	0.6	0.242	0.039	13.819	0.066	0.378	0.770	4.007	0.029	0.050
	0.8	0.256	0.041	11.956	0.087	0.041	0.381	3.124	0.057	0.328
	1	0.224	0.041	9.625	0.044	0.376	0.763	1.534	0.020	0.029
	1.2	0.018	0.041	17.636	0.037	0.380	0.383	1.770	0.034	0.092
$\alpha$	0.1	0.267	0.041	2.026	0.030	0.358	0.390	2.088	0.024	0.095
	0.3	0.024	0.042	7.402	0.087	0.473	0.384	1.760	0.020	0.032
	0.5	0.256	0.041	11.956	0.087	0.041	0.381	3.124	0.057	0.328
	0.7	0.027	0.040	6.341	0.089	0.379	0.770	2.134	0.026	0.073
	0.9	0.257	0.041	4.184	0.033	0.345	0.408	2.925	0.028	0.121

变量至少一次，因此提取保守序列的计算复杂度为  $O(ND)$ 。

(2) 对于归一化保守序列部分，对保守序列进行归一化涉及  $N$  个保守序列。对于每个序列，有两种情况：当保守序列长度小于  $C_l$  时，时间复杂度为  $O(D)$ ；当保守序列长度大于  $C_l$  时，以  $O(1)$  的时间复杂度进行截断处理。因此，归一化  $N$  个保守序列的计算复杂度为  $O(N+DN)$ 。

综上所述，保守分析的总时间复杂度为  $O(N+DN)$ 。而其它需要分析决策变量的方法，如 DECC-DE，需要至少遍历决策变量两次，导致分析部分的时间复杂度至少为  $O(D^2)$ 。这比 CS 中保守分析的时间复杂度要高得多。从时间复杂度分析可以看出，CS 的分析策略相比于其它算法的分析策略，有着更好的性能优势，在接下来的实验部分，将通过实验进一步验证该结论。

## 4.4 实验结果及分析

为了检验 CS 的性能，本节进行如下实验：首先，对 CS 所使用到的关键参数进行参数敏感性分析；随后，验证了 CS 的有效性；验证了 CS 的决策变量分析效率；然后在测试函数集上将 CS 嵌入现有 MOEA 并与六个先进的大规模多目标优化算法，综合验证 CS 的有效性。

### 4.4.1 实验设置

#### (1) 基准测试问题

本节实验采用 LSMOP1-9 作为测试函数集。关于 LSMOP1-9 的特点，参见表3-1。决策变量个数设置为 200-2 000，目标变量个数设置为 2。

## (2) 参数设置

CS 中有 3 个参数：学习率 ( $\eta$ )、保守长度 ( $C_l$ ) 和比率 ( $r$ )，对于其它比较算法的参数，使用 PlatEMO 中的默认值进行公平比较。种群大小  $N$  被设置为 100。对于其它比较的参数，算法使用 PlatEMO 中的默认参数值进行公平比较。

## (3) 终止条件

终止条件为  $MaxFEs$ 。当决策变量个数为 200、500、1 000 和 200 000 时， $MaxFEs$  分别设置为 500 000、500 000、80 000 和 1 000 000。

## (4) 其它

每个算法在每个测试问题上独立运行 10 次。收集每次测试的 IGD 平均值和标准差。用 Wilcoxon 秩和检验对 CS-A 算法和各种算法的结果进行统计学意义检验，显著性水平  $\alpha = 0.05$ 。符号“+”、“-”和“ $\approx$ ”分别表示比较算法显著好于 CS-A，显著差于 CS-A，统计上类似于 CS-A。

### 4.4.2 参数敏感性分析

影响 CS 性能的关键参数有三个，分别为：学习率 ( $\eta$ )、保守长度 ( $C_l$ )、比率 ( $r$ )。为了研究这些参数对算法性能的影响，我们选择 MOEA/D-DE 作为 A，设置  $D = 500$ ，并与每个参数的五个候选值进行比较。我们的实验结果如表4-1所示。最佳结果以灰色背景突出显示。下面讨论每个参数对 CS 性能的具体影响。

(1) 保守长度  $C_l$  参数  $C_l$  表示 CS 中的保守序列的长度，即使用历史信息直接进行更新的决策变量个数。当  $C_l = 0$  时，表示使用嵌入的 MOEA 更新所有决策变量。如表4-1所示， $C_l$  的最佳值为 200。大于或小于 200 的值对算法的最终性能有很大影响。因此，本研究中使用  $C_l = 200$  以获得最佳结果。

## (2) 比率 $r$

参数  $r$  决定了 CS 在 CS-A 中的参与程度，同时也影响  $\eta$  的更新率。当  $r \leq 1$  时，CS 将完全参与 CS-A。如表4-1所示， $r$  的最佳值在  $[0.81.2]$  的范围内。较小的  $r$  值的性能较差，表明 CS 需要在 CS-A 中占相对较大的比例。因此，本研究将对  $r$  使用  $[0.81.2]$  范围内的值。

## (3) 学习率 $\eta$

参数  $\eta$  决定了根据历史信息确定属于 CS 中保守序列的决策变量被更新到的范围。值越大，表明对历史信息的依赖程度越高。如表4-1所示，与 0.7、0.9 相比，较小的  $\eta$  值可以获得更好的结果。这是因为较小的  $\eta$  减少了迭代过程中的振荡。因此，根据实验结果，本研究对  $\eta$  使用  $[0.10.5]$  范围内的值。



表 4-2 不同算法与相应 CS 的版本性能比较

Tab. 4-2 The performance comparison between different algorithms and the corresponding CS version

	NSGAII	CS-NSGAII	GDE3	CS-GDE3	LMOCSSO	CS-LMOCSSO	MOEA/D-DE	CS-MOEA/D-DE
LMSOP1	2.7648e+0	2.7387e+0	1.4509e+0	1.2696e+0	1.4536e+0	1.4900e+0	1.0979e+0	2.0213e-1
LMSOP2	6.2010e-2	6.1726e-2	4.8376e-2	4.8485e-2	4.7948e-2	4.7460e-2	4.7726e-2	4.6935e-2
LMSOP3	1.9771e+1	1.9406e+1	2.4774e+1	2.3449e+1	1.8449e+1	1.8446e+1	1.3018e+1	9.5699e+0
LMSOP4	9.9772e-2	9.7917e-2	9.1299e-2	8.2703e-2	9.2427e-2	9.2174e-2	9.5561e-2	8.2216e-2
LMSOP5	8.5393e+0	8.3856e+0	3.2063e+0	2.9492e+0	3.3041e+0	3.2407e+0	2.1150e+0	3.7444e-1
LMSOP6	8.0840e-1	8.1092e-1	7.8885e-1	7.9420e-1	8.0554e-1	8.1408e-1	7.5539e-1	5.5160e-1
LMSOP7	8.1965e+2	1.7140e+3	9.9121e+2	1.0238e+3	8.9460e+2	7.4903e+2	3.0642e+1	3.7897e+0
LMSOP8	3.6931e+0	3.7465e+0	2.3580e+0	2.0487e+0	2.2791e+0	2.2385e+0	6.2793e-1	3.3003e-2
LMSOP9	1.3654e+0	1.3382e+0	3.0759e+0	3.3671e+0	1.3518e+0	1.5981e+0	5.1201e-1	5.8307e-2

#### 4.4.3 保守序列分组策略的有效性验证

为了验证 CS 在改善现有 MOEAs 在 LSMOP 上的性能方面的有效性, 本节将 CS 嵌入到已发表在著名期刊上的四个不同的 MOEAs。这四种算法分别是 NSGAII<sup>[2]</sup>、GDE3<sup>[2]</sup>、LMOCSSO<sup>[2]</sup>和 MOEA/D-DE<sup>[2]</sup>。将在中 A 嵌入了 CS 版本的算法命名为 CS-A, 并且在 LSMOP1-9 上进行了对比实验, 将每种算法与其 CS 版本进行了比较。决策变量的数量被设置为 200。表4-2提供了实验结果。最佳结果以灰色显示。从该表中可以清楚地看到, 算法的 CS 版本在大多数情况下, 其性能都好于其原始版本。这些结果表明, CS 可以有效地提高现有 MOEAs 在 LSMOPs 上的性能。同时, 实验表明, 无论嵌入式算法采用何种关键技术 (非支配排序、差分进化、粒子群或基于分解), CS 都具有较广泛的普适性。CS 可以很好地适应, 并提升原始算法的性能, 并在大多数测试用例中达到了至少一个数量级的性能提升。在接下来的实验中, 我们将以 CS-MOEA/D-DE 为代表, 在解质量和计算效率方面与目前最先进的 LSMOEAs 进行比较。

#### 4.4.4 保守序列分组策略的效率验证

在这一部分中, 我们将 CS-MOEA/D-DE 与 FDV、LMOCSSO、LMOEAs-DS、WOF 和 CCGDE3 等六种最新算法进行比较, 验证 CS 算法在求解 LSMOP 问题上的效率。使用的测试问题包括 LSMOP1、LSMOP8 和 LSMOP9, 这三个测试题有线性 PF、非线性 PF 和不连续 PF, 可以提供较为全面地进行测试。图4-3绘出了当  $D = 200$  时, CS-MOEA/D-DE 和六种比较算法得到的 IGD 值的中位数在三个测试问题上的收敛曲线。这个图清楚地表明, 无论哪种测试问题, CS-MOEA/D-DE 的收敛曲线都处于图表的最底部。这些结果表明, CS-MOEA/D-DE 具有更快的

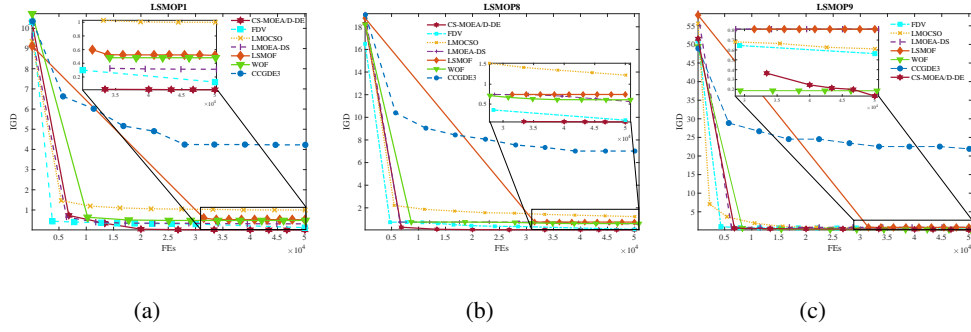


图 4-3 当决策变量数为 200 时，由 CS-MOEA/D-DE 和六个对比算法在 LSMOP1、LSMOP8 和 LSMOP9 上的 IGD 中位数曲线

Fig. 4-3 When the number of decision variables is 200, the IGD median curves obtained by CS-MOEA/D-DE and six comparison algorithms on LSMOP1, LSMOP8, and LSMOP9

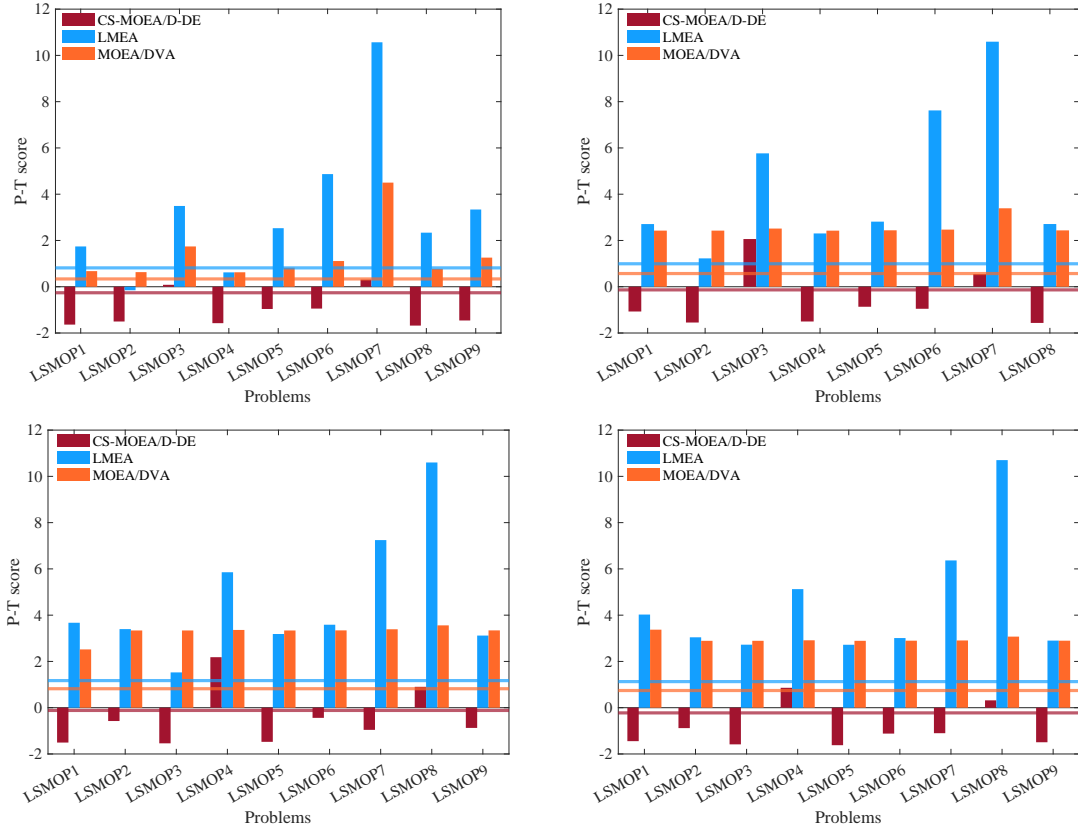


图 4-4 MOEA/DVA、LMEA 和 CS-MOEA/D-DE 对在包含 200、500、1000 和 2000 个决策变量的 LSMOP 测试集上的 P-T 分数

Fig. 4-4 The P-T score obtained by MOEA/DVA, LMEA, and CS-MOEA/D-DE on LSMOP test suit with 200,500,1000 and 2000 decision variables

收敛效率。此外，CS 作为一种分组策略，需要对决策变量进行分析。分析决策变量所花费的时间也是影响算法实用性的一个重要指标。为了从解的质量、决策变量的分析、运算时间等方面综合考虑算法的综合评价。设计了一种用于考察性

能-时间 (P-T 分数) 的复合度量, 其定义如下:

$$\text{P-T Score} = \log(\alpha P + \beta \tau + (1 - \alpha - \beta) F_e) \quad (4-8)$$

其中,  $\alpha$  表示 P-T 分数对解质量或运行时间的偏好, 当  $\alpha > 0.5$  时, P-T 分数侧重于算法的解质量而不是运行时间。同时,  $\beta$  代表了 P-T 分数对决策变量分析时间或总体时间消耗的偏好。当  $\beta > 0.25$  时, P-T 得分更关注算法在决策分析中的时间消耗。本文采用  $\alpha = 0.5, \beta = 0.25$ 。 $P$  表示算法的解质量度量, 如 IGD。 $\tau$  表示决策变量分析算法所需的计算机时间的无量纲形式。假设算法需要对  $T_a$  决策变量进行分析, 对  $i$  决策变量进行 CPU 时间分析, 而  $T_t$  表示算法整体运行的 CPU 时间, 则  $\tau$  可以由  $\frac{\sum_i T_{a_i} t_i}{T_t}$  计算。 $F_e$  表示由于决策变量分析额外消耗的  $FEs$ 。

随后, 将 CS-MOEA/D-DE 与 MOEA/DVA 和 LMEA 在决策变量分析效率方面进行了比较。图4-4在 LSMOP1-LSMOP9 上绘制了  $D$  为 200、500、1000 和 2000,  $MaxFEs$  设置为 500、500、800000 和 1000000 时, 三个算法的 P-T 得分。P-T 得分越低, 算法的整体性能越好。结果表明, CS-MOEA/D-DE 在保证解质量的同时, 明显优于其它两种比较算法。决策变量分析任务可以用很少的 CPU 时间来完成, 而不需要额外的  $FEs$  消耗, 进一步证明了 CS 的效率优势。

#### 4.4.5 与其它算法在基准测试问题上的比较

在这一部分中, 我们以 CS-MOEA/D-DE 为代表, 并将其与七个最先进的大规模 MOEAs 进行比较, 以深入考察 CS 在解决 LSMOP 问题时的解质量性能。这八种比较算法是 FDV、LMOCSSO、LMOEADS、WOF、CCGDE3、LSMOF、MOEA/DVA 和 LMEA。本节使用 LSMOP 测试集的所有九个测试问题。决策变量的数量分别设置为 200、500、1 000 和 2 000。表4-3提供了关于 IGD 的实验结果。在灰色背景下显示算法在当前测试用例的效果最佳。B/W/S 表示每个比较算法明显好于、差于或类似于 CS-MOEA/D-DE 的测试问题的数量。

在表4-3中, 灰色背景的区域清楚地表明, CS-MOEA/D-DE 的整体性能优于其它算法。具体地说, CS-MOEA/D-DE 在 36 个测试实例中的 18 个上表现最好, 而 WOF 是最具竞争力的比较算法, 在 36 个测试实例中的 10 个上获得了最好的结果。紧随其后的是 LMOEADS, 它在 36 个测试实例中的 5 个上运行得最好。LSMOF 在 36 个测试实例中有 3 个获得了最好的结果, 而 FDV 在 36 个测试实例中只有一个得到了最好的结果。LMOCSSO 和 CCGDE3 在任何测试问题上都没有达到最好的结果。对于 MOEA/DVA 和 LMEA, 这两种算法在最大后验误差

表 4-3 CS-MOEA/D-DE 和六种最新算法在 LSMOP 测试问题上的 IGD 值

Tab. 4-3 IGD of CS-MOEA/D-DE and six state-of-the-art algorithms on LSMOP test problems

D	Problem	FDV	LMOCSSO	LMOEADS	WOF	CCGDE3	LSMOF	CS
200	LSMOP1	3.02e-01	1.03e+00	3.13e-01	2.87e-01	3.79e+00	5.45e-01	2.30e-02
		8.43e-2 -	7.81e-2 -	1.73e-2 -	1.93e-1 =	1.34e-1 -	5.53e-2 -	0.00e+00
	LSMOP2	7.33e-02	1.02e-01	2.82e-02	2.80e-02	1.48e-01	3.91e-02	4.11e-02
		1.47e-3 -	2.44e-3 -	9.38e-4 +	9.59e-3 +	1.27e-3 -	1.44e-3 +	0.00e+00
	LSMOP3	4.98e+00	1.15e+01	1.54e+00	1.39e+00	1.98e+01	1.54e+00	4.51e+00
		4.51e+0 =	5.08e+0 -	1.85e-3 +	2.00e-1 +	2.04e+0 -	1.18e-3 +	-1.61e+00
	LSMOP4	1.69e-01	1.74e-01	8.74e-02	9.53e-02	2.17e-01	1.04e-01	6.09e-02
		5.02e-3 -	4.50e-3 -	1.75e-3 -	6.15e-3 -	8.99e-3 -	1.35e-3 -	-1.67e-02
	LSMOP5	5.82e-01	2.42e+00	7.42e-01	3.49e-01	9.68e+00	7.42e-01	2.18e-01
		9.55e-2 -	2.61e-1 -	1.56e-4 -	6.25e-2 =	7.32e-1 -	0.00e+0 -	-1.73e-01
	LSMOP6	8.26e-01	8.40e-01	3.50e-01	3.44e-01	9.30e-01	3.58e-01	5.67e-01
		1.28e-1 =	1.49e-2 -	8.29e-3 +	8.22e-2 =	4.58e-3 -	1.87e-3 +	-1.87e-01
	LSMOP7	4.22e+00	1.90e+02	1.48e+00	1.48e+00	1.53e+04	1.48e+00	1.93e+00
		7.04e-1 -	3.27e+1 -	3.48e-3 +	1.50e-3 +	3.39e+3 -	2.50e-3 +	-3.48e-01
	LSMOP8	3.05e-01	1.36e+00	6.71e-01	2.94e-01	7.84e+00	7.42e-01	4.21e-02
		1.35e-1 -	9.87e-2 -	9.74e-2 -	7.50e-2 -	9.89e-1 -	0.00e+0 -	-1.53e-02
	LSMOP9	9.87e-01	5.96e-01	8.10e-01	6.53e-01	2.05e+01	8.10e-01	1.69e-01
		9.93e-2 -	3.05e-2 -	1.97e-4 -	2.68e-1 -	7.58e+0 -	1.24e-16 -	-1.16e-01
500	LSMOP1	7.32e-01	1.45e+00	3.18e-01	3.73e-01	5.49e+00	6.02e-01	2.02e-01
		2.89e-1 -	6.89e-2 -	7.73e-3 =	8.36e-2 =	4.93e-1 -	3.76e-2 -	-1.25e-01
	LSMOP2	3.50e-02	4.79e-02	1.47e-02	1.49e-02	7.22e-02	2.32e-02	4.69e-02
		1.61e-3 +	6.11e-4 =	3.91e-4 +	2.77e-3 +	1.28e-3 -	3.25e-4 +	-1.01e-03
	LSMOP3	4.14e+01	1.84e+01	1.57e+00	1.54e+00	2.71e+01	1.57e+00	9.57e+00
		6.66e+1 =	5.95e+0 =	1.00e-3 +	7.16e-2 +	1.44e+0 -	9.82e-4 +	-4.07e+00
	LSMOP4	1.06e-01	9.24e-02	4.25e-02	4.79e-02	1.17e-01	5.21e-02	8.22e-02
		4.93e-3 -	1.45e-3 -	5.72e-4 +	6.37e-3 +	1.81e-3 -	5.21e-4 +	-7.82e-03
	LSMOP5	2.06e+00	3.30e+00	7.42e-01	6.70e-01	1.26e+01	7.42e-01	3.74e-01
		1.58e+0 -	1.31e-1 -	9.07e-5 -	1.21e-1 -	7.96e-1 -	0.00e+0 -	-1.72e-01
	LSMOP6	3.62e+00	8.06e-01	3.20e-01	4.77e-01	8.15e-01	3.20e-01	5.52e-01
		3.39e+0 -	9.19e-3 -	6.98e-4 +	1.74e-1 =	2.89e-3 -	4.03e-4 +	-1.96e-01
	LSMOP7	6.83e+01	8.95e+02	1.51e+00	1.51e+00	2.34e+04	1.50e+00	3.79e+00
		3.70e+1 -	1.46e+2 -	5.25e-4 +	3.47e-3 +	3.56e+3 -	7.00e-4 +	-1.05e+00
	LSMOP8	6.13e-01	2.28e+00	7.42e-01	4.48e-01	9.17e+00	7.42e-01	3.30e-02
		2.44e-2 -	1.58e-1 -	1.34e-4 -	1.53e-1 -	6.13e-1 -	0.00e+0 -	-1.05e-02
	LSMOP9	1.14e+00	1.35e+00	7.47e-01	4.93e-01	2.79e+01	8.09e-01	5.83e-02
		1.68e-1 -	6.81e-1 -	6.88e-2 -	3.31e-1 -	4.90e+0 -	6.30e-4 -	-5.26e-03
1000	LSMOP1	6.38e-01	1.60e+00	3.16e-01	2.41e-01	5.65e+00	6.28e-01	1.45e-01
		2.77e-2 -	8.01e-2 -	5.54e-3 -	1.36e-1 =	3.19e-1 -	4.38e-2 -	-8.83e-02
	LSMOP2	2.09e-02	2.62e-02	8.90e-03	1.02e-02	4.00e-02	1.82e-02	2.61e-02
		1.10e-3 +	4.02e-4 =	3.73e-4 +	1.55e-3 +	4.43e-4 -	6.34e-4 +	-6.85e-04
	LSMOP3	8.88e+01	7.90e+02	1.57e+00	1.54e+00	2.90e+01	1.57e+00	1.35e+01
		1.59e+2 =	1.71e+3 -	4.75e-4 +	8.36e-2 +	1.62e+0 -	4.76e-4 +	-5.06e+00
	LSMOP4	6.32e-02	5.30e-02	2.31e-02	2.59e-02	7.11e-02	3.25e-02	5.11e-02
		6.91e-4 -	3.58e-4 -	5.74e-4 +	1.97e-3 +	1.91e-3 -	1.01e-3 +	-1.06e-03
	LSMOP5	8.13e-01	3.53e+00	7.42e-01	6.10e-01	1.31e+01	7.42e-01	8.99e-02
		8.18e-2 -	2.06e-1 -	8.63e-5 -	1.21e-1 -	5.01e-1 -	0.00e+0 -	-1.82e-02
	LSMOP6	8.26e-01	8.40e-01	3.50e-01	3.44e-01	9.30e-01	3.58e-01	5.67e-01
		1.28e-1 =	1.49e-2 -	8.29e-3 +	8.22e-2 =	4.58e-3 -	1.87e-3 +	-1.87e-01
	LSMOP7	3.64e+01	1.14e+03	1.51e+00	1.51e+00	2.87e+04	1.51e+00	4.37e+00
		2.01e+1 -	1.15e+2 -	2.19e-4 +	1.13e-3 +	7.34e+3 -	6.53e-4 +	-3.77e+00
	LSMOP8	6.70e-01	2.39e+00	7.42e-01	4.08e-01	1.01e+01	7.42e-01	7.13e-02
		2.25e-2 -	1.22e-1 -	1.38e-4 -	3.00e-1 -	9.86e-1 -	0.00e+0 -	-9.56e-02
	LSMOP9	1.02e+00	2.14e+00	6.18e-01	7.97e-01	3.16e+01	8.07e-01	2.70e-02
		2.24e-1 -	3.51e-1 -	2.34e-2 -	2.24e-2 -	4.81e+0 -	1.20e-3 -	-2.87e-03
2000	LSMOP1	5.30e-01	3.19e-01	1.56e+00	9.93e-02	5.80e+00	6.39e-01	6.66e-02
		1.77e-2 -	6.09e-3 -	-6.03e-02	1.53e-2 -	9.40e-2 -	1.47e-2 -	-1.62e-02
	LSMOP2	1.05e-02	5.92e-03	1.45e-02	6.76e-03	2.18e-02	1.24e-02	1.41e-02
		6.24e-4 +	1.73e-4 +	-2.46e-04	3.89e-4 +	3.13e-4 -	3.50e-4 +	3.99e-4
	LSMOP3	6.71e+00	1.58e+00	7.43e+00	1.29e+00	2.97e+01	1.58e+00	7.22e+00
		3.33e+0 =	4.09e-4 +	-2.19e+00	3.87e-1 +	1.57e+0 -	1.94e-4 +	-3.11e+00
	LSMOP4	1.09e-02	1.26e-02	2.96e-02	1.44e-02	4.10e-02	3.29e-02	2.93e-02
		3.47e-4 +	1.74e-4 +	-2.47e-04	1.40e-3 +	1.45e-3 -	1.16e-3 -	3.47e-3
	LSMOP5	2.30e-01	7.42e-01	2.85e+00	3.62e-01	1.31e+01	7.42e-01	2.14e-01
		2.24e-2 =	1.09e-5 -	-6.52e-02	9.11e-2 =	8.76e-1 -	0.00e+0 -	-8.38e-02
	LSMOP6	7.48e-01	3.10e-01	7.57e-01	1.88e-01	7.57e-01	3.09e-01	6.01e-01
		3.40e-4 =	3.36e-3 +	-1.66e-04	6.14e-3 +	4.24e-5 -	2.27e-5 +	-2.03e-01
	LSMOP7	1.51e+00	1.51e+00	4.92e+02	1.42e+00	2.65e+04	1.51e+00	1.87e+00
		1.75e-3 +	1.12e-4 +	-2.87e+01	1.30e-1 +	7.63e+2 -	2.64e-4 +	-8.91e-02
	LSMOP8	8.29e-02	7.22e-01	1.80e+00	1.62e-01	9.57e+00	7.42e-01	3.77e-02
		8.81e-3 -	1.38e-2 -	-4.93e-02	2.47e-2 -	2.61e-1 -	0.00e+0 -	-2.11e-02
	LSMOP9	1.04e-01	4.99e-01	8.42e-01	2.87e-01	2.87e+01	8.05e-01	1.05e-02
		1.98e-1 -	7.92e-2 -	-3.90e-01	2.37e-1 -	4.72e+0 -	4.01e-4 -	-5.50e-03
+/-/=		5/23/8	0/30/6	19/16/1	12/8/16	0/36/0	18/18/0	

下都不能完成优化。这是因为他们在决策变量分析中消耗了太多的  $FES$ 。根据 Wilcoxon 秩和检验，B/W/S 列表明，在 36 个测试实例中，CS-MOEA/D-DE 在 23、30、16、12、36 和 18 个测试实例上明显优于 FDV、LMOCSSO、IMOEDS、WOF、CCGDE3 和 LSMOF。另一方面，表4-4提供了 HV 指标上的实验结果，这也显示了 CS-MOEA/D-DE 的性能优势。它在 36 个测试实例中的 18 个上执行得最好。其次是 WOF、LSMOF、FDV 和 LMOEADS。它们在 36 个测试用例上分别获得了 4、3、2、2 的最好结果。同时，在本实验中，其它基于决策变量分析的算法，如 MOEA/DVA，并没有参与比较，因为这些基于决策变量分析的算法不能在预设的  $MaxFES$  下完成求解，甚至不能完成决策变量分析过程。这证明了 CS 作为决策变量分析方法的效率优势。

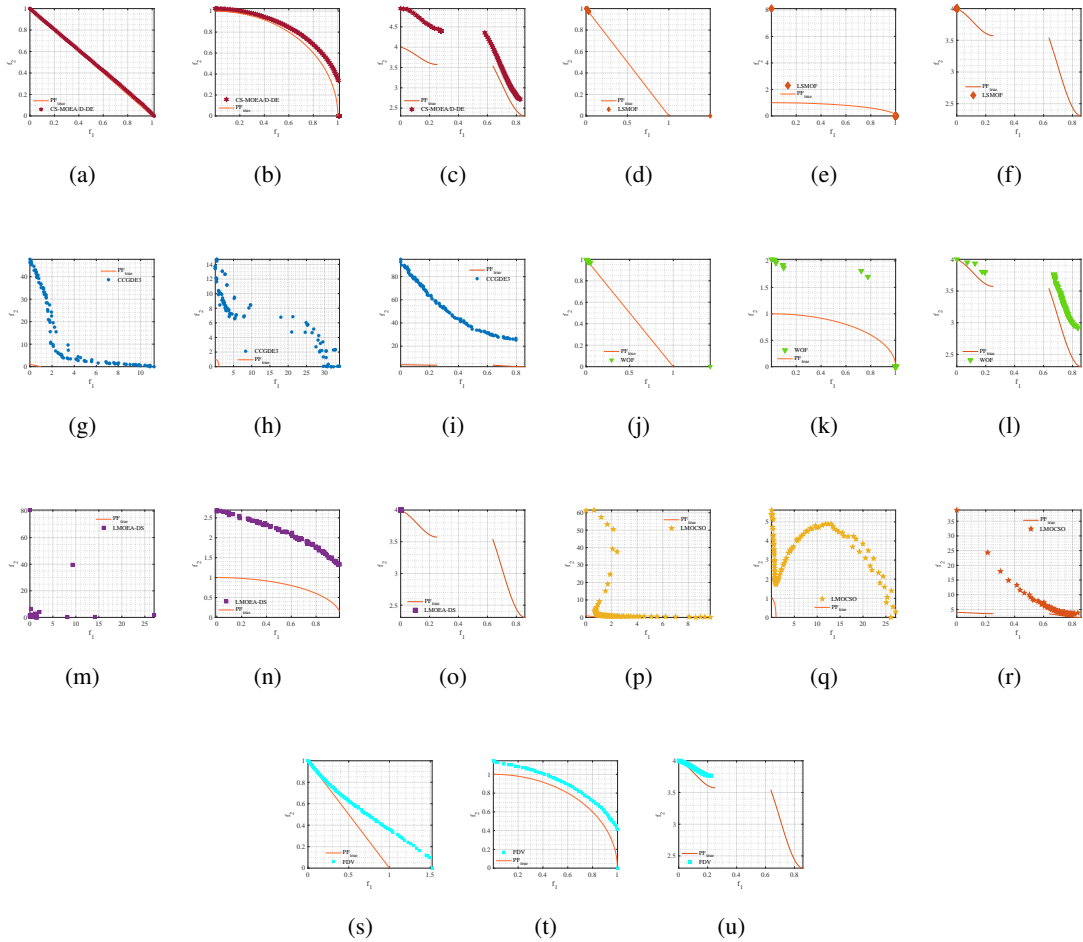


图 4-5 CS-MOEA/D-DE 和其他七个对比算法在具有 200 个决策变量的 LSMOP1、LSMOP8 和 LSMOP9 上得到的  $PF$

Fig. 4-5 The  $PF$  obtained by CS-MOEA/D-DE and the other seven comparison algorithms on LSMOP1, LSMOP8, and LSMOP9 with 200 decision variables.

为了直观地展示实验结果，图4-5根据 LSMOP1、LSMOP8 和 LSMOP9 上绘

表 4-4 CS-MOEA/D-DE 和六种最新算法在 LSMOP 测试问题上的 HV 值

Tab. 4-4 HV of CS-MOEA/D-DE and six state-of-the-art algorithms on LSMOP test problems

D	Problem	FDV	LMOCSSO	LMOEADS	WOF	CCGDE3	LSMOF	CS-MOEA/D-DE
200	LSMOP1	2.42e-01 5.01e-2 -	0.00e+00 0.00e+0 -	2.36e-01 1.23e-2 -	2.84e-01 1.74e-1 =	0.00e+00 0.00e+0 -	4.25e-01 2.84e-2 =	4.75e-01 -1.19e-01
	LSMOP2	4.93e-01 1.72e-3 +	4.57e-01 2.74e-3 -	5.48e-01 9.71e-4 +	5.48e-01 1.22e-2 +	4.02e-01 5.70e-4 -	5.49e-01 2.66e-3 +	4.77e-01 -2.01e-03
	LSMOP3	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+00
	LSMOP4	3.82e-01 4.52e-3 -	3.76e-01 5.11e-3 -	4.74e-01 2.26e-3 -	4.61e-01 8.18e-3 -	3.30e-01 9.18e-3 -	4.69e-01 8.04e-3 -	5.07e-01 -2.07e-02
	LSMOP5	8.44e-02 1.11e-2 -	0.00e+00 0.00e+0 -	9.09e-02 1.45e-5 -	9.84e-02 1.68e-2 -	0.00e+00 0.00e+0 -	1.08e-01 2.28e-2 =	1.91e-01 -1.05e-01
	LSMOP6	7.74e-06 1.73e-5 -	0.00e+00 0.00e+0 -	1.83e-02 2.28e-4 -	4.09e-02 1.06e-2 =	0.00e+00 0.00e+0 -	1.93e-02 5.08e-3 -	4.51e-02 -1.61e-02
	LSMOP7	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+00
	LSMOP8	9.76e-02 2.15e-2 -	0.00e+00 0.00e+0 -	9.09e-02 8.21e-6 -	1.24e-01 2.52e-2 -	0.00e+00 0.00e+0 -	2.93e-01 2.53e-3 =	2.98e-01 -2.01e-02
	LSMOP9	2.55e-02 1.81e-2 -	5.07e-02 2.73e-3 -	9.09e-02 9.99e-6 -	1.12e-01 3.52e-2 -	0.00e+00 0.00e+0 -	1.52e-01 1.13e-2 =	1.72e-01 -4.58e-02
500	LSMOP1	3.08e-02 2.85e-2 -	0.00e+00 0.00e+0 -	2.30e-01 1.64e-2 =	1.77e-01 4.84e-2 -	0.00e+00 0.00e+0 -	9.09e-02 0.00e+0 -	3.62e-01 -1.22e-01
	LSMOP2	5.39e-01 2.91e-3 +	5.22e-01 7.92e-4 =	5.65e-01 6.65e-4 +	5.65e-01 3.46e-3 +	4.91e-01 1.29e-3 -	5.30e-01 9.13e-4 +	5.23e-01 -1.22e-03
	LSMOP3	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+00
	LSMOP4	4.53e-01 5.84e-3 -	4.69e-01 1.67e-3 -	5.28e-01 8.93e-4 +	5.21e-01 8.34e-3 +	4.38e-01 2.26e-3 -	4.72e-01 2.79e-3 -	4.81e-01 -9.17e-03
	LSMOP5	0.00e+00 0.00e+0 -	0.00e+00 0.00e+0 -	9.09e-02 8.79e-6 =	9.09e-02 0.00e+0 =	0.00e+00 0.00e+0 -	9.09e-02 0.00e+0 =	7.78e-02 -9.30e-02
	LSMOP6	0.00e+00 0.00e+0 -	0.00e+00 0.00e+0 -	3.67e-02 2.63e-3 -	7.54e-02 2.50e-2 =	0.00e+00 0.00e+0 -	3.32e-02 2.42e-3 -	5.48e-02 -1.25e-02
	LSMOP7	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+00
	LSMOP8	4.01e-02 7.62e-3 -	0.00e+00 0.00e+0 -	9.09e-02 3.52e-7 -	9.96e-02 1.67e-2 -	0.00e+00 0.00e+0 -	9.09e-02 0.00e+0 -	3.04e-01 -1.43e-02
	LSMOP9	1.14e-02 2.13e-2 -	7.50e-03 7.16e-3 -	1.01e-01 1.20e-2 -	1.41e-01 4.97e-2 -	0.00e+00 0.00e+0 -	1.15e-01 1.85e-3 -	2.15e-01 -1.79e-03
1000	LSMOP1	5.59e-02 1.87e-2 -	0.00e+00 0.00e+0 -	2.40e-01 8.60e-3 -	3.08e-01 1.41e-1 =	0.00e+00 0.00e+0 -	9.46e-02 2.50e-3 -	4.14e-01 -9.03e-02
	LSMOP2	5.57e-01 1.05e-3 +	5.50e-01 6.03e-4 =	5.72e-01 5.80e-4 +	5.72e-01 2.00e-3 +	5.32e-01 5.45e-4 -	5.74e-01 1.29e-3 +	5.50e-01 -9.61e-04
	LSMOP3	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+00
	LSMOP4	5.04e-01 8.00e-4 -	5.16e-01 4.67e-4 -	5.54e-01 7.34e-4 +	5.51e-01 2.48e-3 +	4.93e-01 2.13e-3 -	5.62e-01 1.02e-3 +	5.18e-01 -1.31e-03
	LSMOP5	1.57e-02 3.51e-2 -	0.00e+00 0.00e+0 -	9.09e-02 9.11e-6 -	9.09e-02 0.00e+0 -	0.00e+00 0.00e+0 -	9.09e-02 0.00e+0 -	2.26e-01 -2.31e-02
	LSMOP6	7.74e-06 1.73e-5 -	0.00e+00 0.00e+0 -	1.83e-02 2.28e-4 -	4.09e-02 1.06e-2 =	0.00e+00 0.00e+0 -	1.93e-02 5.08e-3 -	4.51e-02 -1.61e-02
	LSMOP7	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+00
	LSMOP8	1.54e-02 9.42e-3 -	0.00e+00 0.00e+0 -	9.09e-02 1.34e-5 -	1.47e-01 5.41e-2 =	0.00e+00 0.00e+0 -	9.09e-02 0.00e+0 -	2.80e-01 -6.90e-02
	LSMOP9	4.75e-02 3.37e-2 -	0.00e+00 0.00e+0 -	1.27e-01 4.96e-3 -	9.28e-02 3.37e-3 -	0.00e+00 0.00e+0 -	1.45e-01 2.76e-3 -	2.29e-01 -1.44e-03
2000	LSMOP1	7.24e-02 1.41e-2 -	0.00e+00 0.00e+0 -	2.31e-01 8.05e-3 -	4.51e-01 2.85e-2 -	0.00e+00 0.00e+0 -	1.13e-01 6.79e-3 -	5.03e-01 -2.03e-02
	LSMOP2	5.71e-01 9.39e-4 +	5.66e-01 3.24e-4 =	5.77e-01 2.53e-4 +	5.76e-01 5.43e-4 +	5.56e-01 3.67e-4 -	5.69e-01 3.98e-4 +	5.66e-01 -5.43e-04
	LSMOP3	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+00
	LSMOP4	5.71e-01 4.63e-4 +	5.45e-01 3.34e-4 =	5.68e-01 2.34e-4 +	5.66e-01 1.71e-3 +	5.31e-01 1.49e-3 -	5.42e-01 1.30e-3 -	5.46e-01 -4.60e-04
	LSMOP5	1.51e-01 1.24e-2 =	0.00e+00 0.00e+0 -	9.09e-02 1.00e-6 -	1.01e-01 2.22e-2 =	0.00e+00 0.00e+0 -	9.09e-02 0.00e+0 -	1.36e-01 -5.36e-02
	LSMOP6	7.86e-02 7.24e-4 =	5.92e-02 3.13e-4 -	9.04e-02 4.08e-4 +	1.32e-01 1.72e-2 +	5.91e-02 8.00e-5 -	8.91e-02 4.44e-5 =	7.98e-02 -7.12e-03
	LSMOP7	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+0 =	0.00e+00 0.00e+00
	LSMOP8	2.56e-01 6.46e-3 -	0.00e+00 0.00e+0 -	9.09e-02 2.77e-7 -	1.97e-01 2.15e-2 -	0.00e+00 0.00e+0 -	9.09e-02 0.00e+0 -	3.01e-01 -2.47e-02
	LSMOP9	2.22e-01 2.90e-2 -	2.04e-02 1.15e-2 -	1.46e-01 4.06e-3 -	1.93e-01 3.21e-2 -	0.00e+00 0.00e+0 -	9.16e-02 5.13e-5 -	2.39e-01 -4.49e-04
+/-/=		5/21/10	11/25/00	8/18/10	8/13/15	0/28/8	5/17/14	

制了除 MOEA/DVA 和 LMEA 之外的其它算法在决策变量数为 200 时得到的  $PF$  图。从图中可以清除地看到, CS-MOEA/D-DE 在所有三个测试问题上对  $PF$  进行了最好的覆盖。基于以上分析, CS-MOEA/D-DE 在解质量方面具有最好的整体性能。这一点间接地表明, CS 可以提高现有 MOEA 求解 LSMOPs 的性能。

## 4.5 本章小结

本章提出了一种新的保守序列分组策略 CS, 该策略一定程度上缓解了决策变量分析分组策略中的计算效率低下的问题, 同时能提高 LSMOPs 上多目标进化算法的性能。CS 受到自然进化中保守序列概念的启发, 将决策变量分为保守序列和非保守序列。其中保守序列被定义为连续两次更新中变化差异较小的决策变量, 非保守序列被定义以为连续两次更新中变化差异较大的决策变量。其中保守序列根据其父代进化过程中的历史信息进行优化, 减少了算法的盲目搜索, 增强了算法性能。在 LSMOP 基准测试集上的实验结果表明验证了 CS 令人满意的性能。





## 结 论

近年来,大规模多目标优化问题逐渐成为多目标优化领域的研究热点。然而,由于大规模决策变量会造成决策空间的维度灾难,一般的多目标进化算法不能有效解决这类问题。目前,决策变量分组策略是提高大规模优化算法解决大规模多目标优化问题的重要策略之一。本文深入研究了现有决策变量分组策略存在的问题,并完成了两个研究工作:

第一,针对随机分组策略可能将相互关联的决策变量划分为不同组从而影响求解性能的问题,本研究提出了一种双分解分组策略。该策略采用基于滑动窗口的外分解策略将决策变量划分为重叠的组,通过内分解进一步将决策变量组所对应的 **MOP** 分解为一系列 **SOPs**,并使用差分变异算法作为基本优化器来搜索新的解。同时,它还采用块坐标下降思想来维护原始问题的解。通过将 **DDS** 嵌入现有的 **MOEA** 并与 8 种先进的进化算法在基准测试问题上的比较实验表明,嵌入了 **DDS** 的 **MOEA** 在求解效率和求解质量方面具有显著优势。此外,在两个实际问题上的实验结果表明,嵌入了 **DDS** 的算法能够处理多达 3072 个决策变量的问题,并以至少一个数量级优于其他对比算法获得最佳性能。

第二,针对决策变量分析分组策略在求解 **LSMOP** 问题时计算效率低下的问题,本研究受自然进化中的保守序列概念的启发,提出了一种名为 **CS** 的保守序列分组策略。该策略分为两个阶段:首先,根据保守序列分析策略将所有决策变量划分为保守序列和非保守序列;然后,对保守序列和非保守序列分别采用特定的优化策略。通过针对保守序列和非保守序列的特点采取针对性的优化策略,该算法能够同时提高求解质量和求解效率。在一系列基准测试问题上进行的实验表明,与现有的 8 种最先进的算法相比,嵌入了 **CS** 的算法具有较强的竞争力,而与两种典型的决策分析分组策略相比,**CS** 在决策变量分析部分的计算量可以忽略不计。

本研究提出了两种不同的分组技术来处理大规模多目标优化问题,并为大规模多目标优化问题求解提供了新的手段。然而,仍有很多内容值得深入研究。未来的工作将主要集中在以下两个方向:

(1) 探索更灵活的滑动窗口模式,在 **DDS** 中滑动窗口的单一模式导致泛化能力较差。因此,未来的研究打算从参数自动化和更灵活的滑动窗口模式来扩展本研究所提出的 **DDS**。具体地,可以使用网格搜索或贝叶斯优化来自动设置参数。此外,可以尝试使用广义滑动窗口,使其可以向不同方向滑动,而不是一维

方向滑动，这有利于求解现实应用中的多维编码决策变量，滑动窗口还可以使用聚类法来组织，以同时处理个体中相距较远的相关决策变量。因此，将参数自动化设置和更灵活的滑动窗口引入 CS 进行研究算十分有意义的工作。

(2) 探索多层次的保守序列分组方法，在 CS 中，决策变量仅仅被划分为保守序列和分保守序列，当决策变量个数进一步增加时，分组后的保守序列和非保守序列很有可能仍然具有大规模特点，导致一次分组带来的性能和效率提升有限。因此，未来的研究打算从多层次的保守序列分析出发来扩展提出的 CS。具体地，可以使用凝聚式分层聚类或最大化原则最优策略分层的思想自底向上地不同层次上进行多次保守序列分析分组，有利于 CS 处理具有更大规模决策变量的大规模多目标优化问题。因此，如何进行多层次的保守序列分析分组算未来需要解决的重要研究问题。

## 攻读硕士学位期间索取得的主要科研成果

### 论文和专利

1. Yang C C, **Wang P K**, Ji J Z. A dual decomposition strategy for large-scale multiobjective evolutionary optimization[J]. Neural Computing and Application , 2023, 35(5): 3767-3788. (中科院 2 区, JCR Q1, IF 5.102)
2. Yang C C, **Wang P K**, Ji J Z. A conserved sequence grouping for large-scale multi-objective optimization[J]. IEEE/CAA Journal of Automatica Sinica (中科院 1 区, JCR Q1, IF 7.847 投稿中)
3. 杨翠翠, **王佩科**, 冀俊忠. 针对大规模多目标优化问题的双分解策略. 中国发明专利. (已送审)

### 参与的项目

1. 国家自然科学基金委员会, 青年科学基金项目, 61906010, 面向 fMRI 脑功能连接的粗糙集特征归约方法研究, 2020-01-01 至 2022-12-31.
2. 北京市教委, 北京市教委科技计划一般项目, KM202010005032, 基于菌群算法的 fMRI 脑功能连接粗糙集特征归约方法研究, 2020-01 至 2022-12.

### 参与的科技竞赛

1. **王佩科**, 王飞鹏, 冀予, 杨思思. 2021 年第十五届中国计算机博弈锦标赛围棋项目二等奖 (亚军)
2. 刘钺锋, **王佩科**, 宋晓. 2021 年第十五届中国计算机博弈锦标赛麻将项目三等奖
3. 陈京, 赵润丰, 叶传泰, **王佩科**. 2022 年第十六届中国计算机博弈锦标赛久棋项目一等奖 (冠军)
4. 赵润丰, 陈京, 王亭, **王佩科**. 2022 年第十六届中国计算机博弈锦标赛麻将项目二等奖
5. 叶传泰, 贺静, 陈京, **王佩科**. 2022 年第十六届中国计算机博弈锦标赛围棋项目三等奖

## 获得的主要奖励

1. 2021 国家奖学金
2. 2021 北京工业大学优秀研究生奖
3. 2021 北京工业大学研究生学习优秀奖（一等）
4. 2021 北京工业大学科技创新奖（一等）
5. 2021 北京工业大学科技创新奖（优秀）

## 致 谢

在此，我衷心感谢所有在我攻读硕士学位期间给予我帮助、支持和关爱的老师、同学、家人和朋友。

首先，我要向杨翠翠老师表示由衷的谢意。杨老师严谨的治学态度、独到的学术见解以及高度负责的教书育人精神深深地感染和激励着我。在整个研究过程中，她给予了我莫大的支持，耐心细致地指导我的研究方向，引导我正确分析问题并解决困难。她对题目的探讨，使我在学术上拓宽视野，在思维上不断突破。在论文的写作过程中，针对我的不足提出了许多宝贵的建议和改进措施，帮助我一步步完善论文结构和内容。在这里，我由衷地向杨老师表达最真挚的感谢。

同时，我也要向冀俊忠老师表达深深的敬意与感激。冀老师严谨的科研态度、严密的逻辑思维以及广博的专业知识给我带来了极大的启发。没有冀老师的悉心指导，我的研究工作很难取得如今的成果。我向冀老师表达由衷的感谢。

实验室的同学们也是我攻读硕士期间重要的支持者。在这个充满智慧与活力的环境中，我们共同度过了无数欢乐时光，共同分享了科研成果。实验室同学们提供了无私的帮助，不仅在学术上有着严谨的态度，更在生活中给予温暖和陪伴。感谢每一个实验室成员的付出，让我在这里收获友情和知识。

我还要感谢我的父母和家人。他们始终关心我，鼓励我，尽全力让我无忧地投入学术研究。在我遇到困难时，他们从未放弃过我，总是给我信心和动力。感谢他们无尽的爱和支持，我才能够勇往直前，追求自己的梦想。在此，我愿向父母和家人送上最深沉的爱意，希望用我未来的辉煌成就回报他们的付出。

我要特别感谢我的女朋友杨思思，无论是在学习还是生活上，她无条件地给予了我很多的鼓励、支持和帮助。我十分感谢有杨思思这样优秀的人在我成长的道路上支持并引导我。

最后，我要向那些在背后默默关注、支持和帮助我的人表示感谢。正是你们的陪伴，让我在攻读硕士学位的道路上不再孤单，相信自己可以完成这段旅程。

请原谅我不能一一列举所有给予我帮助鼓励的人，但请相信，你们的名字已经深深地刻在我的心中，会永远珍藏。谢谢！

Not farewell, but fare forward, voyagers.

感谢支持、鼓励和信任。