



汕 頭 大 學

SHANTOU UNIVERSITY

硕 士 学 位 论 文

题 目 面向不平衡优化问题的进化算法研究及其在
减速器设计中应用

英文题目 **Research on Evolutionary Algorithm for Imbalanced**
Optimization Problem and Its Application in Reducer Design

姓 名 杨知 学 号 151809040

所在学院 工学院 导师姓名 唐雅娟, 范衡

专 业 电子与通信工程

入学日期 2018年9月 答辩日期 2021年5月

学位论文原创性声明

本论文是我个人在导师指导下进行的工作研究及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其他机构已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在论文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

作者签名：_____ 日期：____年____月____日

学位论文使用授权声明

本人授权汕头大学保存本学位论文的电子和纸质文档，允许论文被查阅和借阅；学校可将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存和汇编论文；学校可以向国家有关部门或机构送交论文并授权其保存、借阅或上网公布本学位论文的全部或部分内容。对于保密的论文，按照保密的有关规定和程序处理。

作者签名：_____ 导师签名：_____
日期：____年____月____日 日期：____年____月____日



汕頭大學
SHANTOU UNIVERSITY

硕士学位论文

论文中文题目：面向不平衡优化问题的进化算法研究及其在
减速器设计中应用

论文英文题名：Research on Evolutionary Algorithm for Imbalanced
Optimization Problem and Its Application in Reducer Design

指导教师：唐雅娟

申请人：杨知

论文答辩委员会成员

主席：沈民奋 教授 (汕头市智能制造产业联盟理事长)

委员：庄哲民 教授 (汕头大学工学院电子系)

李芬兰 教授 (汕头大学工学院电子系)

袁野 副教授 (汕头大学工学院电子系)

魏楚亮 教授 (汕头大学工学院电子系)

摘要

多目标优化是目前广泛被关注的一类研究，在各类优化算法中，来源于生物进化思想的进化算法因其具有全局性、鲁棒性、自组织、广泛适用性等特点，能够有效的处理传统优化算法难以解决的复杂多目标优化问题，逐渐受到广大研究学者们的关注。由于大多数现实的优化问题都包含约束，而约束存在的多样性难度和收敛性难度，导致这类问题难以有效被解决，那么利用进化计算来求解约束多目标优化问题就具有了十分重要的应用前景。本文以进化算法以及约束优化处理技术的最新研究成果作为出发点，提出了一种混合的约束多目标优化算法，用来求解约束多目标优化问题。主要研究内容包括以下部分：

1、为了解决目标不平衡且约束同时具有多样性困难和收敛性困难的约束多目标优化问题（CMOPs），提出了一种混合约束多目标进化算法—M2M-IEpsilon，该方法将改进的 Epsilon 约束处理方法与 M2M 分解方法结合用来处理 CMOPs。所提出的算法由于采用了 M2M 分解方法，它能将一个 CMOP 分解为一组简单的约束多目标优化子问题，能够处理目标不平衡优化问题，从而维持算法的多样性。同时，M2M-IEpsilon 采用 IEpsilon 约束处理方法，能够在可行域和不可行域之间保持良好的搜索平衡，帮助种群跨越大的不可行区域，提高了所提出算法的收敛性能。

2、为有效评估 M2M-IEpsilon 算法的性能，弥补现有 CMOPs 只单独关注目标或者约束的不足，设计了一组目标不平衡和约束同时具有多样性困难和收敛性困难的约束多目标测试集，即 ICD-CMOPs，用于评估 CMOEAs 算法的性能。6 中先进的 CMOEAs 与提出的 M2M-IEpsilon 算法在 ICD-CMOPs 进行实验比较。综合实验结果表明所提出的 M2M-IEpsilon 在 ICD-CMOPs 上的性能明显优于其他 6 种算法，说明了所提出的算法在处理目标不平衡且约束具有多样性困难和收敛性困难的 CMOPs 方面的优越性。

3、将提出的 M2M-IEpsilon 与其他 6 种 CMOEAs (CM2M, CM2M2, NSGA-II-CDP, MOEA/D-CDP, PPS-MOEA/D 和 MOEA/D-IEpsilon)算法用于减速器设计优化问题上，测试该算法在实际优化问题上的性能。实验结果表明，M2M-IEpsilon 在该实际优化问题上的表现优于其他 6 种算法。

关键词：多目标优化；进化算法；约束处理方法；减速器设计

Abstract

Multi-objective optimization is a type of research which has been widely concerned at present. Among the various optimization algorithms, the evolutionary algorithm which is derived from the idea of biological evolution has been gradually gaining the attention of scholars because it can effectively deal with complex multi-objective optimization problems that are difficult to be solved by traditional optimization algorithms and its global, robust, self-organizing and widely applicable characteristics. Since most real-world optimization problems have constraints, the convergence-hard and diversity-hard constraints make such problems difficult to be solved effectively. Then the use of evolutionary computation to solve constrained multi-objective optimization problems has a very important application prospect. In this paper, we propose a hybrid constrained multi-objective optimization algorithm for solving constrained multi-objective optimization problems based on the latest research results of evolutionary algorithms and constrained optimization handling techniques. The research mainly includes the following parts.

1. In order to solve imbalanced CMOPs with simultaneous convergence-hard and diversity-hard constraints, this paper proposes a hybrid algorithm which combines an improved epsilon constraint-handling method (IEpsilon) with a multi-objective to multi-objective (M2M) decomposition approach, namely M2M-IEpsilon. Since M2M-IEpsilon adopts the M2M decomposition method, it can decompose a CMOP into a set of simple constrained multi-objective optimization subproblems, which can deal with the imbalanced objective optimization problem, to enhance the diversity of the proposed algorithm. At the same time, M2M-IEpsilon employs the IEpsilon constraint handling method, which can help populations get across large infeasible regions, and improve the convergence performance of the proposed algorithm.

2. To effectively evaluate the performance of the M2M-IEpsilon algorithm and compensate for the lack of existing CMOPs that only focused on objectives or constraints, imbalanced CMOPs with simultaneous convergence-hard and diversity-hard constraints, namely ICD-CMOPs, are employed to evaluate the performance of M2M-IEpsilon and six state-of-the-art constrained multi-objective evolutionary algorithms (CMOEA), including CM2M, CM2M2, NSGA-II-CDP, MOEA/D-CDP, PPS-MOEA/D and MOEA/D-IEpsilon. The experimental results demonstrate that the proposed M2M-IEpsilon significantly

outperforms the other six algorithms on ICD-CMOPs, which illustrates the superiority of the proposed algorithm in dealing with imbalanced CMOPs with simultaneous convergence-hard and diversity-hard constraints.

3. With the six state-of-the-art CMOEAs, including CM2M, CM2M2, NSGAI-CDP, MOEA/D-CDP, MOEA/D-IEpsilon, PPS-MOEA/D, the proposed M2M-IEpsilon carried out on speed reducer design optimization problem evaluate the performance of the proposed algorithm in real-world optimization problem. The experimental results show that M2M-IEpsilon outperforms the other six algorithms for this real-world optimization problem.

Keywords: Multi-objective Optimization, Evolutionary Algorithm, Constraint Handling Method, Speed Reducer Design

目 录

摘要.....	I
Abstract	II
目 录.....	IV
第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 进化算法研究现状.....	2
1.2.1 基于分解的 MOEAs.....	5
1.2.2 基于帕累托支配的 MOEAs.....	5
1.2.3 基于指标的 MOEAs.....	6
1.3 进化算法评价指标.....	6
1.3.1 收敛性评价指标.....	7
1.3.2 多样性评价指标.....	7
1.3.3 综合性评价指标.....	7
1.4 本文主要研究内容.....	8
1.5 论文结构.....	8
第 2 章 约束处理技术及相关内容.....	11
2.1 约束优化问题的定义.....	11
2.2 约束处理方法.....	12
2.2.1 罚函数法.....	13
2.2.2 约束和目标分离方法.....	15
2.2.3 混合方法.....	16
2.3 测试问题集.....	17
2.4 本章小结.....	19
第 3 章 M2M-IEpsilon 约束多目标进化算法.....	21
3.1 M2M 分解方法.....	21
3.2 IEpsilon 约束处理方法.....	22
3.3 M2M-IEpsilon.....	23
3.3.1 M2M-IEpsilon 的算法框架.....	24
3.3.2 种群初始化.....	25
3.3.3 子代个体生成.....	26

3.3.4 更新 $\varepsilon(k)$ 的值	26
3.3.5 选解	26
3.4 测试函数设计	27
3.4.1 测试问题集设计背景	27
3.4.2 测试问题细节描述	29
3.5 本章小结	33
第 4 章 实验研究	35
4.1 实验设置	35
4.2 比较算法	35
4.3 性能比较	36
4.4 本章小结	42
第 5 章 M2M-IEpsilon 算法在减速器设计上的应用	43
5.1 减速器设计	43
5.2 实验结果与分析	45
5.3 本章小结	47
第 6 章 总结	49
6.1 研究总结	49
6.2 研究展望	49
参考文献	51
攻读学位期间主要研究成果	58
致谢	59

第 1 章 绪论

随着现代科学和工程的发展,许多现实生活及工程应用中的优化问题变得越来越具有挑战性。非线性、复杂、约束、高维、目标之间相互冲突始终围绕着优化问题的特性,使得优化问题的求解面对的困难也随之增加。《物种起源》学说揭示了自然界物种的起源,低级、简单的生物经过“物竞天择、优胜劣汰、适者生存”的长期演化完成了到高级、复杂生物的蜕变。借用这一思想,研究者们从生物进化的过程中获得启发,开发了模拟生物演化行为的进化算法(Evolutionary Algorithms, EAs)。随着这一算法的发展,因其具备高鲁棒性、自组织、自适应、自学习以及全局最优搜索能力的特征,运用进化算法求解优化问题也逐渐成为了一大趋势,广泛被用于约束多目标优化问题的求解。

1.1 研究背景及意义

优化问题是日常生活中最常见的问题之一,从计算机科学、工程到运筹学和经济数学等各个领域,优化问题涉及到所有的定量学科中,广泛应用于经济、工程、交通、金融、通信、国防等各个领域。通常来讲,利用优化算法对优化问题进行求解,找到给定约束条件下某个目标函数的最优值是解决优化问题的通用过程。目前,优化算法的研究成果在调度^[1],金融投资^[2],电力电子^[3],机器学习^[4],工程优化^[5],以及网络通信^[6]等领域得到了广泛的应用。

优化问题根据目标函数或者约束函数的不同,分类有以下几种:

- 1、是否有约束条件:约束/非约束优化问题;
- 2、目标的数量:单目标/多目标优化问题;
- 3、函数类型:线性/非线性问题;

决策变量、目标函数、约束条件构成了优化问题的三个要素。单目标优化问题的含义是只包含一个目标,而多目标优化问题(Multi-Objective Optimization Problems, MOPs)的含义是小于或等于 3 个目标的优化问题,而超过三个目标的优化问题则定义为超多目标优化问题。一般来讲,对于单目标优化问题,其在解的数量上,是存在唯一最优解的,而对于多目标或超多目标优化问题往往包含多个相互冲突的目标,需要在多个目标之间进行权衡,从而获得一组相对的最优解。而在众多的研究中,多目标优化算法的研究一直占据主要位置,一方面是因为超多目标优化问题建模困难,而且逐渐增多的目标函数数量,导致其整体的计算量出现指数级增长。另一方面是高维目标空间中解的可视化问题并未得到很好地解决,同时用于解决实际中的超多目标优

化问题依然停留在理想化状态。基于此，多目标优化问题的研究因更具有实际研究的可行性，吸引了众多研究学者的广泛关注。

当谈及人工智能算法，基本上是以深度学习为主流的几类算法，深度学习的快速发展直接将之矗立于时代的前沿，棋类博弈、计算机视觉、语音识别以及自动驾驶等领域都表现得惊人的好。但任何一类算法都并非完美而没有缺陷的，如梯度下降法、牛顿法等，对于某些特定类型的问题，具有较好的优化效果。但是这些算法对问题的性质往往具有一定的要求，对于一些复杂的问题，这些算法往往不再适用。因其算法本身的特性，容易导致算法陷入局部最优，而得不到较好的解，而且无法解决不可微的问题。相比之下，进化算法是不需要梯度就能跳出局部最优的黑箱算法，在某些复杂的优化问题上表现出普遍的鲁棒性、自组织、广泛适用性，能够有效的处理传统优化算法难以解决的复杂问题（比如 NP 困难问题），在优化领域占据了无可比拟的重要位置。进化算法经常应用在多目标问题的优化求解中来，这类算法称之为多目标进化算法（Multi-objective Evolutionary Algorithms, MOEAs）。多目标进化算法通过借鉴生物进化的思想在解决多目标优化问题中提供了比较新颖的方法和思路，广泛应用与参数优化、工业调度、资源分配、复杂网络分析等，在解决多目标优化问题上无疑是巨大的进步。

鉴于许多现实生活中的优化问题，比如工程问题、调度问题等优化问题都存在约束条件，虽然这类问题描述简单且具有很强的代表性，但这些众多的限制条件使得这类问题求解面临着极大的挑战。主要原因是求解这类问题需要强大的运算能力以及存储空间，而且约束条件的存在也导致了搜索空间的复杂性，另一方面带来的“组合爆炸”也成了求解这类问题的难点。因此，受约束函数的多目标优化问题一直备受优化领域研究学者的关注，在过去几十年，研究人员对于如何高效率来求解约束多目标优化问题的关注度与日俱增，逐渐发展了这一学科对于业界的影响。

1.2 进化算法研究现状

借鉴自然界的某种现象或受生物某种机制启发而发展起来的算法，经过无数科研工作者的几十年的研究，逐步形成了进化算法^[7]，蚁群算法（Ant Colony Algorithm, ACO）^[8]、粒子群算法（Particle Swarm Optimization, PSO）^[9]、人工鱼群算法（Artificial Fish Swarm Algorithm, AFSA）^[10]、免疫算法（Immune Algorithm, IA）^[11]、模拟退火算法（Simulated Annealing, SA）^[12]、人工蜂群算法（Artificial Bee Colony, ABC）^[13]等。这些算法解决了传统方法难以解决的问题，求解出一组近似最优解集，解决了多目标优化问题的困难，应用前景十分广阔。其中进化算法在求解高度复杂的非线性问题时具有较好的效果，能够有效避开局部最优，再加上具备较好的通用性，是一种高

鲁棒性的全局优化算法，进而逐步得到广泛的应用。

进化算法又称演化算法，是一种基于种群的通用元启发式优化算法，借鉴自然界中生物进化的思想，包括种群初始化、个体适应度评价、繁殖、突变、重组、和选择、基因编解码等基本操作完成种群的进化过程，进化的过程即对种群（解）不断迭代改进的过程，通过选择、变异和交叉遗传算子，不断产生新的个体，个体历经“优胜劣汰”，最终进化成优秀的个体，满足进化的终止条件之后输出最优个体（最优解）。目前来说，进化算法主要包括了遗传算法（Genetic Algorithms, GA）^[14]、进化策略（Evolutionary Strategy, ES）^[15]、遗传编程（Genetic Programming, GP）^[16]以及进化编程（Evolutionary Programming, EP）^[17]等，见图 1-1 所示，这几类算法非常相似但又是彼此独立发展起来的，下面简单介绍这四种经典的进化算法。

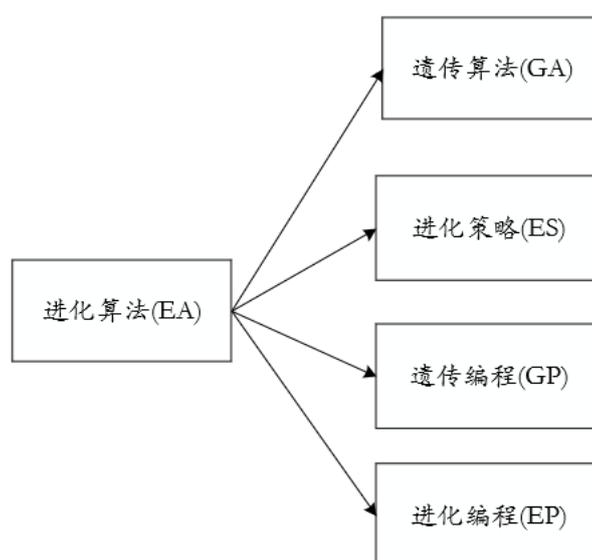


图 1-1 进化算法分类

遗传算法由美国 J.Holland 创建，是一种众所周知的基于自然的优化技术，也是目前最常用的一种 EA 算法，流程框图如 1-2 所示。在 GA 中，对解空间的搜索模仿了环境中发生的自然过程，并考虑了达尔文的物种进化理论。在 GA 中，由个体构成种群；每一个个体，被称为染色体，表示问题的潜在解。被解决的问题由目标函数定义，为了更新当前种群，该算法利用一些“选择”策略从当前种群中反复选择父染色体（一对个体解），通常是根据它们的适合度函数来进行选择。然后，通过交叉和突变算子的操作，将被选中的父代染色体用于繁殖，并将产生的一对子代用于形成下一代种群，反复迭代进化，直至找出最优个体（最优解）。常用来解决机器学习^[18]、优化问题^[19]、自动控制^[20]以及机器人^[21]等研究领域，是进化算法中最流行的一种子算法。

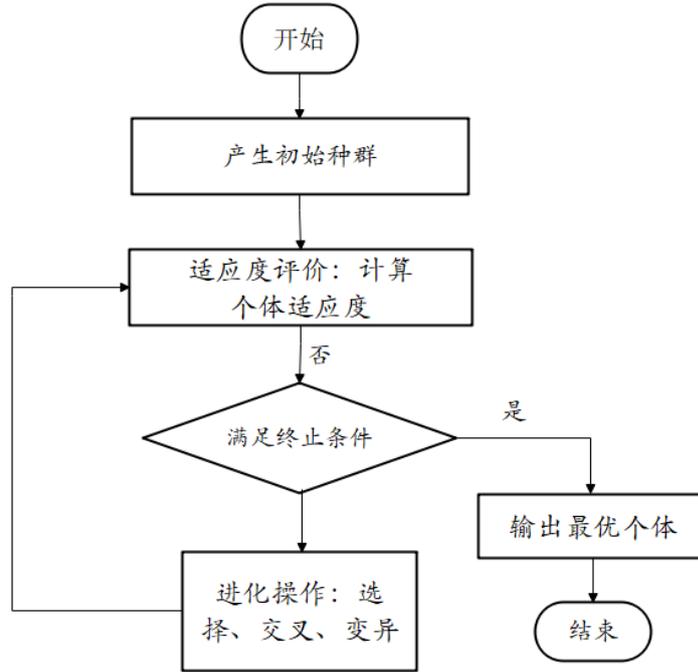


图 1-2 遗传算法流程图

进化策略是由德国的 Rechenberg 建立的，是一种模仿生物进化的一种求解参数优化问题的方法；不过与遗传算法不一样的是，它采用实数值作为基因；它总遵循零均值、某一方差的高斯分布的变化产生新的个体，然后保留好的个体。进化策略的实现更加简单（不需要反向传播），更容易在分布式环境中扩展，不会受到奖励稀疏的影响，有更少的超参数。其在现代强化学习基准（如 Atari/MuJoCo）上的表现在一定程度上比肩标准的强化学习技术，同时还能克服强化学习的诸多不便^[22]。目前分类包括了 PAES^[23]、SMES^[24]、CMA^[25]、S3-CMA-ES^[26]。

遗传编程是从生物进化过程得到灵感的自动化生成和选择计算机程序来完成用户定义的任务的技术。由美国斯坦福大学的 John Koza 基于自然选择原则创造性的提出了用层次化的计算机程序来表达问题的遗传程序设计（遗传编程）。具有结构灵活，能同时执行多项任务，生成模型具有很好的解释能力，适应面广，搜索能力强等。主要应用在符号回归、电路设计、游戏以及图像处理方面，包括了基于树的遗传编程、线性遗传编程以及基于图的遗传编程^[27]。

进化规划的概念最早由美国的 Fogel 提出，它的工作原理与进化策略的工作原理相当接近。与 ES 不同的是，它对属性数据类型的使用没有限制。EP 有固定的程序结构，它允许数值参数的进化。预测、泛化、游戏和自动控制是进化编程的一些合适的应用领域^[28]。

单目标和多目标优化，是一类常见的优化决策问题，经过研究学者们多年的努力，有关于多目标进化算法的研究以及应用方面都已经获得非常大的成功，在多目标决策

领域占据了主流位置。多目标进化算法用于解决多目标优化问题方面是非常流行的，这也从侧面反映了多目标进化算法在这方面的优越性。MOEA 的三个目标分别是，其一，找到一组尽可能接近于真实 Pareto 前沿的解集，称为收敛性；其二，找到分布良好的解集，称为多样性；其三，覆盖整个帕累托前沿（Pareto Front, PF），称为覆盖率。而在常见分类中，MOEAs 根据选择策略的不同可以大致分为三大类，第一类是基于分解的方法；第二类是基于非支配排序的方法；第三类是基于指标的方法。这三类方法是目前主要流行的方法，基于这三者的基础也逐步形成了多种新颖的算法，旨在让算法的效率更高、性能方面更好。

1.2.1 基于分解的 MOEAs

第一类是基于分解的 MOEAs，即把一个原始的 MOP 分解成若干个单目标优化问题或更简单的 MOP 来协同解决。一些基于分解的 MOEAs 如 MOGLS^[29]、MOEA/D^[30]、RVEA^[31]通过目标函数聚合将一个 MOP 分解为若干个多目标优化问题，使候选解能够有效地收敛到每个单目标的最优值。分解策略常见于传统多目标优化，但在此之前分解策略并未在进化算法中得到广泛应用，直至 2007 年张青富教授研究出基于分解的多目标进化方法（MOEA/D）之后才逐渐得到广泛关注，这一方法也被多目标优化研究同行认同为一类独立的算法。MOEA/D 的核心思想是把多目标优化问题分解成一组单目标子问题或多个多目标子问题，并通过利用子问题之间的领域关系，以相互协作的方式来同时优化所有子问题，从而接近真实的 Pareto 面。近年来，MOEA/D 逐步得到了越来越广泛的应用，成为极具影响力的进化算法。研究者们在此基础上进行研究，一系列新的基于分解的多目标进化算法被开发出来，MOEA/D-DE^[32]，MOEA/D-DD^[33]，MOEA/D-ACO^[34]，MOEA/D-DU^[35]。此外，也有一些研究者试图把基于分解的进化算法框架扩展至约束多目标的优化，并将基于分解的 MOEAs 应用到解决复杂的现实世界的优化问题方面，这些都取得了较好的研究成果，比如 MOEA/D-ACDP^[36]和改进的 MOEA/D 用来解决主动配电网（Active Distribution Network, ADN）的调度优化^[37]等。

1.2.2 基于帕累托支配的 MOEAs

第二类是基于帕累托支配的方法，该方法通过同时优化所有目标来优化多目标优化问题，基于帕累托支配原则来确定一个解的适应度，这一原则在基于支配的 MOEAs 的收敛中起着关键的作用，先后形成了 NSGA-II^[38]，SPEA2^[39]，和 PESA-II^[40]等几种算法。NSGA-II（Non-dominated Sorting Genetic Algorithm-II）是目前来说最具代表性的基于支配的算法，有效降低这一算法的复杂性，具有运行速度快，解集的收敛性好，使得这一算法通常作为其他的多目标优化算法性能比较的基准。NSGA-II 利用对个体

的快速非支配排序方法、比较个体拥挤距离以及精英个体保留策略,在求解低维 MOP 的时候计算效率是比较高的,而在解集的质量上表现也是比较好的。基于此,该算法逐渐获得了快速的发展,基于 NSGA-II 发展起来的算法,例如: R-NSGA-II^[41], TKRNSGA-II^[42]和 ϵ -NSGA-II^[43]等。而对于高维复杂 MOP 的求解,种群中非支配个体的数量是非常大的,导致计算的复杂度进一步加深,通过 NSGA-II 就无法获得有效帕累托解集。Deb 等人在 NSGA-II 的基础上,提出了一种针对高维多目标问题的多目标进化算法 NSGA-III^[44],随后 Yang 等人基于网格优势的演化算法 (GrEA)^[45]和 Zhang 等人提出基于 Knee Point 驱动的进化算法 (KnEA)^[46]都是用来解决超多目标优化问题。

1.2.3 基于指标的 MOEAs

基于指标的方法,其设计思路是根据评价指标的数值来引导种群搜索解,即在环境选择中采用解决方案质量测量的性能指标作为选择标准。近年来,大量的性能指标被提出,比如世代距离 (Generational Distance, GD)^[47]、超体积 (Hypervolume, HV)^[48]、反转世代距离 (Inverted Generational Distance, IGD)^[49]、R2 指标^[50]、Spacing 指标 (Spacing Metric)^[51]和 IGD⁺^[52]。这些评价指标不仅广泛应用在评价解决方案的质量,而且在 MOEAs 中也作为选择标准。这类的代表有 SMS-EMOA^[53]、GDE-MOEA^[54]和 TS-R2EA^[55]等,其环境选择策略分别基于超体积指标、世代距离和 R2 指标进行设计的。Bader 和 Zitzler 提出了 HypE^[56],采用基于 Monte Carlo 模拟的 HV 估计方法来代替精确的 HV 计算,这样当目标数较多时, HV 计算的计算效率可以大幅提高。

1.3 进化算法评价指标

多目标进化算法的快速发展对于求解多目标优化问题是重要的,这也为如何有效衡量这些算法的性能带来了新的挑战,使得多目标进化算法的评价指标逐渐成为一大热门研究方向。从一定角度来看,建立行之有效的评价标准不仅能够客观的反应进化算法对于求解多目标优化问题的优劣,且对于多目标进化算法的发展在一定程度上有着深远的意义。而在目前,诸多进化算法的性能评价方法,可以大致围绕以下三大类来考虑:

- 1、进化算法所求得解集与真正的帕累托前沿的逼近程度,亦称收敛性。
- 2、进化算法所获得的解集覆盖目标空间的范围,以及分布在真实帕累托前沿上的均匀程度,亦称分布性。
- 3、综合考虑进化算法所得到解集的收敛性和分布性,亦称综合性能。

1.3.1 收敛性评价指标

世代距离是指算法求得的非支配解集中所有个体到真实 Pareto 前沿的平均距离。它的定义如下：

$$\left\{ \begin{array}{l} GD(P^*, A) = \frac{\sum_{y \in A} d(y, P^*)}{|A|} \\ d(y, P^*) = \min_{y_i^* \in P^*} \left\{ \sqrt{\sum_{i=1}^m (y_i - y_i^*)^2} \right\} \end{array} \right. \quad (1-1)$$

P^* 是真实 PF 的代表点集， A 是算法求得的 PF。 $d(y, P^*)$ 的值是 m 维目标空间中 A 中第 i 个体 y_i 到 P^* 中最近个体 y_i^* 的欧几里得距离。值得注意的是，GD 值越小，表明获得的解集越逼近真实 Pareto 解集，意味着此算法的收敛性越好。

1.3.2 多样性评价指标

空间评价方法可用于来衡量帕累托前沿近似解集中的个体在目标空间的一个分布情况。Spacing 指标的定义如下：

$$\left\{ \begin{array}{l} S(A) = \sqrt{\frac{1}{|A|-1} \sum_{i=1}^{|A|} (\bar{d} - d_i)^2} \\ d_i = \min \\ \bar{d} = \frac{1}{|A|} \sum_i d_i \end{array} \right. \quad (1-2)$$

\bar{d} 表示为所有 d_i 的均值； d_i 表示为 $|A|$ 中第 i 个个体与其最近个体的欧几里得距离； $|A|$ 为解集 A 中个体的总数量。 S 值越小意味着解集的个体分散越均匀，多样性越好。

1.3.3 综合性评价指标

综合评价指标是利用一个标量值来衡量多目标进化算法的收敛性能和分布性能，即用来评价算法的综合性能，对于 MOEA 的评价具有重要意义。目前广泛使用的综合评价指标包括了超体积和反转世代距离两种，也是目前非常流行的两种评价指标。

IGD 是各参考点到其最近解的平均距离，是评价 MOEAs 性能最常用的方法。IGD 指标的计算方法如下：

$$\left\{ \begin{array}{l} IGD(P^*, A) = \frac{\sum_{y^* \in P^*} d(y^*, A)}{|P^*|} \\ d(y^*, A) = \min_{y \in A} \left\{ \sqrt{\sum_{i=1}^m (y_i^* - y_i)^2} \right\} \end{array} \right. \quad (1-3)$$

P^* 是真实 PF 的代表点集, A 是算法求得的 PF。 $d(y^*, A)$ 的值是 m 维目标空间中参考 y_i^* 与解 y_i 之间的最小欧氏距离。值得注意的是, IGD 值越小, 表明算法的多样性和收敛性能都越好。

超体积是指通过计算非支配解集与参考点 $z^r = (z_1^r, \dots, z_m^r)^T$ 围成的目标空间区域的超体积的值, 超体积的计算方法如下:

$$HV(S) = VOL(\bigcup_{x \in S} [f_1(x), z_1^r] \times \dots \times [f_m(x), z_m^r]) \quad (1-4)$$

$VOL(\cdot)$ 是勒贝格测度, m 是目标的数量。值得注意的是, HV 值越大, 表明算法的多样性和收敛性能都越好。

1.4 本文主要研究内容

本文主要研究内容围绕约束多目标优化问题展开, 然后结合基于分解的多目标进化算法框架和约束处理机制提出了一种混合的约束多目标进化算法, 用来解决目标不平衡且约束同时具有收敛性难度和多样性难度的 CMOPs。论文的主要工作有以下三个方面:

1、提出了一种混合约束多目标进化算法—M2M-IEpsilon, 用来解决目标不平衡且约束同时具有多样性困难和收敛性困难的 CMOPs。首先, 采用 M2M 分解方法来处理目标, 来维持工作种群的多样性。其次, 利用 IEpsilon 约束处理技术, 以提高该算法的收敛性。M2M-IEpsilon 可以根据当前种群中可行解的比例, 动态调整约束违反的松紧度, 在可行与不可行区域之间保持良好的平衡搜索能力, 具有跨越大面积不可行区域的能力。

2、为有效评估 M2M-IEpsilon 算法的性能, 弥补现有 CMOPs 只单独关注目标或者约束的不足, 设计了一组目标不平衡和约束同时具有多样性困难和收敛性困难的约束多目标测试问题集, 即 ICD-CMOPs, 用于评估 CMOEAs 算法的多样性和收敛性。最后通过在 ICD-CMOPs 约束测试函数上进行实验, 综合实验结果表明所提出的 M2M-IEpsilon 相较于其他 6 种算法具有优越性。

3、为进一步验证 M2M-IEpsilon 的性能, 选择经典的减速器设计作为优化目标。并与其他 6 种约束多目标进化算法的实验对比, 结果表明所提出算法在实际工程优化上也具有优越性。

1.5 论文结构

本文围绕约束多目标优化算法, 针对目标不平衡且约束具有多样性难度和收敛性难度的约束多目标优化问题开展研究, 提出的算法在工程设计问题上进行了测试。全

文分为 6 章，主要的结构安排如下：

第一章是绪论部分。这部分主要围绕研究背景、进化算法的研究现状以及进化算法的评价指标进行了阐述。简要介绍了进化算法的定义、研究现状和对于求解多目标优化问题方面的重要意义。

第二章围绕本文研究的约束多目标优化问题，介绍了约束多目标优化问题的定义，现有约束多目标的处理机制和约束多目标优化问题的测试集。

第三章介绍了 M2M 和 IEpsilon 约束处理机制的核心思想，提出了一类混合多目标进化算法，即 M2M-IEpsilon，详述了 M2M-IEpsilon 算法的核心框架。最后围绕约束多目标测试问题，设计了一组约束具有多样性困难和收敛性困难的不平衡 CMOPs，即 ICD-CMOPs。

第四章对提出的 M2M-IEpsilon 算法在 ICD-CMOPs 测试集上进行了对比实验和性能分析。

第五章提出的 M2M-IEpsilon 方法在现实工程优化问题上进行了实验和比较。

第六章主要围绕本文的研究内容开展了总结，同时展望了未来的研究方向。

第 2 章 约束处理技术及相关内容

优化问题由于约束的存在使得这类优化问题具有不一样的特性，导致算法求解的难度进一步加深。如何设计约束处理技术也就成为了算法十分关注的一个关键点。本章将围绕约束处理技术展开描述，并简要介绍现有测试问题集，方便本文之后的工作研究。

2.1 约束优化问题的定义

通常来说一个约束多目标优化问题（Constrained Multi-objective Optimization Problem, CMOP）可以用下列数学表达式描述，为方便处理，本文把多目标优化函数统一变换成最小化目标优化问题的形式^[57]：

$$\begin{cases} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ \text{subject to} & g_i(\mathbf{x}) \geq 0, i = 1, \dots, q \\ & h_j(\mathbf{x}) = 0, j = 1, \dots, p \\ & \mathbf{x} \in \mathbf{R}^n \end{cases} \quad (2-1)$$

其中 $F(\mathbf{x})$ 是一个 m 维的目标函数， $\mathbf{x} \in \mathbf{R}^n$ 是一个 n 维的决策向量。 $g_i(\mathbf{x}) \geq 0$ 表示不等式约束， $h_j(\mathbf{x}) = 0$ 表示等式约束。 q 和 p 分别代表不等式和等式约束的个数。

为了处理多目标优化问题的约束，总约束违反 ϕ 定义如下：

$$\phi(\mathbf{x}) = \sum_{i=1}^q |\min(g_i(\mathbf{x}), 0)| + \sum_{j=1}^p |\min(h_j(\mathbf{x}), 0)| \quad (2-2)$$

如果一个解 $x \in \mathbf{R}^n$ ，并且它的总约束违反 $\phi(x)$ 等于 0，那么 x 就表示是一个可行解。否则， x 是不可行解。对于任意两个可行解 x_p 和 x_q ，当满足下列两个条件时， x_p 就支配 x_q ，表示为 $x_p \preceq x_q$ 。

- 1) $\forall i \in \{1, \dots, m\}, f_i(x_p) \leq f_i(x_q)$
- 2) $\exists j \in \{1, \dots, m\}, f_j(x_p) < f_j(x_q)$

对于一个可行解 $x^* \in X$ ，如果在 X 中没有其他可行解支配 x^* ，那么 x^* 可定义为帕累托最优解。所有帕累托最优解的集合称为帕累托集合（Pareto Set, PS）。将帕累托集合映射到目标空间，得到一组向量，这个集合称为帕累托前沿（Pareto Front, PF）。

为更容易理解这些概念，图 2-1 展示了两个双目标优化问题中目标空间中可行解与不可行解以及支配关系的定义。灰色格子区域表示不可行区域，那么位于不可行区

域的点 A、B、G 则代表不可行解；点 C、D、E、F 表示可行区域的可行解，点 C、D 是 PF 上两个相互非支配的解。灰色阴影区域是指点 D 所能支配的区域，表明在该区域的所有点都能被点 D 所支配，从图中可以看出点 E 是落在该区域的，说明点 E 被点 D 所支配。

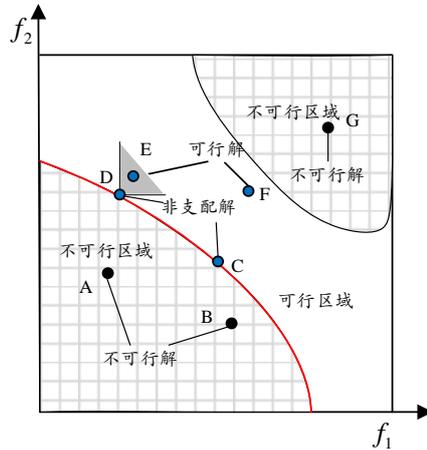


图 2-1 目标空间可行解与不可行解、支配关系示例

图 2-2 描述了决策空间到目标空间的映射，其中图(a)和图(b)灰色格子区域分别表示可行决策空间和可行目标空间。点 A1、B1、C1 是决策空间的解集，其中点 A1、B1、C1 为帕累托最优解的集合；点 A2、B2、C2 为 PS 映射到目标空间的向量即帕累托前沿。

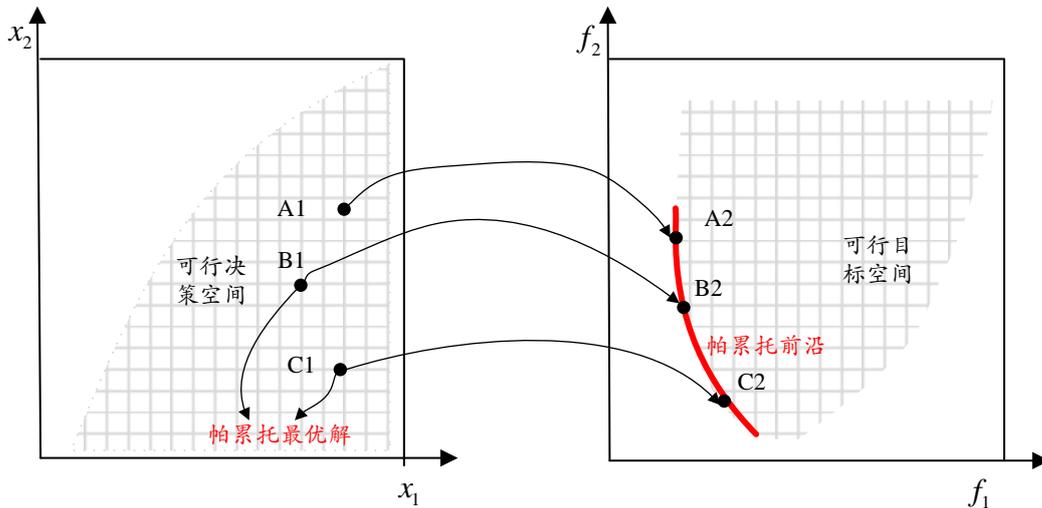


图 2-2 PS 和 PF 示例

2.2 约束处理方法

由于约束条件的存在，如何做到有效处理约束条件成为进化算法求解约束多目标

优化问题面临的主要任务。用来处理约束多目标优化问题的进化算法称为约束多目标进化算法 (CMOEA), 一般来说, CMOEA 有两个方面, 一是 MOEA; 二是约束处理方法 (Constraint Handling Methods, CHMs)。不同的约束处理方法对约束多目标优化算法具有不同的影响, 好的约束处理方法能够提高约束多目标优化算法对问题的求解效率。目前来说, CMOEA 面临的挑战是如何开发出合适的 CHMs 集成到 MOEA 中来处理搜索过程中的约束条件。根据近年来的研究趋势, 常见的约束处理方法分为几类: 罚函数法、约束和目标分离的方法以及混合方法等。

2.2.1 罚函数法

罚函数的总体思想是根据某一解中存在的约束违反量, 在目标函数上加 (或减) 一定的值, 将约束优化问题转化为非约束优化问题。

在数学编程中, 考虑了两种惩罚函数: 外部惩罚函数和内部惩罚函数。一般在进化算法中都利用外部惩罚函数法, 该方法不需要提供初始解, 而内部惩罚函数法需要提供初始的可行解, 这对于解决实际问题可以说是致命的, 所以本文中主要围绕外部惩罚函数法来进行介绍。

进化算法中通常采用以下形式的外部惩罚函数:

$$\varphi(x) = f(x) \pm \left[\sum_{i=1}^n r_i \times G_i + \sum_{j=1}^p c_j \times L_j \right] \quad (2-3)$$

其中 $\varphi(x)$ 是要优化的新目标函数, G_i 和 L_j 是分别是约束 $g_i(x)$ 和 $h_j(x)$ 的函数, 最常见的形式如公式所示, β 和 γ 的值通常是 1 或者 2。 r_i 和 c_j 都是正数, 称之为惩罚因子。

$$\begin{cases} G_i = \max[0, g_i(x)]^\beta \\ L_j = |h_j(x)|^\gamma \end{cases} \quad (2-4)$$

罚函数既可以处理等式约束, 也可以处理不等式约束, 通常的方法是将等式约束转换为如下形式的不等式约束。其中 ε 是可容忍的正容差值, 一般是非常小的数。

$$|h_j(x)| - \varepsilon \leq 0 \quad (2-5)$$

如何确定惩罚因子和构造惩罚函数是非常关键的, 基于此, 逐渐形成了几种不同的罚函数方法, 包括死罚函数法、静态罚函数法、动态罚函数法以及自适应罚函数法。

1) 死罚函数法

死罚函数法, 拒绝一切不可行的个体 (也叫“死刑”), 可能是处理约束最简单的方法, 而且它在计算上也很有效率, 因为当某个解违反了一个约束条件, 它就被拒绝再次生成^[58]。因此, 不需要进一步的计算来估计这种解决方案的不可行程度, 但这种方法在可行域非常小的优化问题上几乎处于停滞状态。

2) 静态罚函数法

静态罚函数法 (Static penalty functions) 是 Homaifar, Lai and Qi 提出的方法^[59], 在这一类方法下, 惩罚因子不以任何方式来依赖当前代数, 而是在整个进化过程中保持恒定。其定义的约束违反程度以及与之相关的惩罚因子如下所示:

$$\text{fitness}(x)=f(x)+\sum_{i=1}^m(R_{k,i} \times \max[0, g_i(x)]^2) \quad (2-6)$$

其中 $R_{k,i}$ 为惩罚系数, m 是总的约束数量, $f(x)$ 是未惩罚的目标函数, $k=1,2,\dots,l$, l 是由用户定义的违反程度的数目。但这种方法惩罚因子设置依赖实际问题, 在整个进化过程中保持相同的惩罚因子从某种角度来说实际效果并不佳。

3) 动态罚函数法

动态罚函数 (Dynamic penalty functions) 在这一类别中, 考虑了任何惩罚函数, 其中当前的代数参与了相应的惩罚因子的计算 (通常惩罚因子的定义方式是随时间增加—即随代数增加)^[60]。具体的惩罚函数形式构造如下:

$$\text{fitness}(x)=f(x)+(C \times t)^\alpha \times \text{SVC}(\beta, x) \quad (2-7)$$

C , α , β 是由用户定义的预设参数, t 表示当前种群代数。

$$\text{SVC}(\beta, x)=\sum_{i=1}^n D_i^\beta(x)+\sum_{j=1}^p D_j(x) \quad (2-8)$$

$$D_i(x)=\begin{cases} 0, & g_i(x) \leq 0 \\ |g_i(x)|, & \text{otherwise} \end{cases} \quad 1 \leq i \leq n \quad (2-9)$$

$$D_j(x)=\begin{cases} 0, & -\varepsilon \leq h_j(x) \leq \varepsilon \\ |h_j(x)|, & \text{otherwise} \end{cases} \quad 1 \leq j \leq p \quad (2-10)$$

4) 自适应罚函数法

自适应罚函数法 (Adaptive penalty functions) 是由 Bean 和 Hadj-Alouane 开发了一种使用惩罚函数的方法^[61], 该函数从搜索过程中获得反馈。每个个体都由下列公式进行评估:

$$\text{fitness}(x)=f(x)+\lambda(t)\left[\sum_{i=1}^n g_i^2(x)+\sum_{j=1}^p |h_j(x)|\right] \quad (2-11)$$

$\lambda(t)$ 在每一代的更新根据下列规则:

$$\lambda(t+1)=\begin{cases} \lambda(t)/\beta_1, & \text{if case I occurs} \\ \beta_2 \cdot \lambda(t), & \text{if case II occurs} \\ \lambda(t), & \text{Otherwise} \end{cases} \quad (2-12)$$

其中 $\beta_1 > \beta_2 > 1$ ，情况 1 和 2 表示最后 k 次迭代中的最佳个体总是可行的(case I)或是总是不可行的(case II)。换句话说，如果最后 k 代中所有的最佳个体都是可行的，则 $t+1$ 代的惩罚分量 $\lambda(t+1)$ 减少；如果它们都是不可行的，则惩罚分量会增加。这种方法试图避免拥有一个全部可行或全部不可行的种群，最近的约束处理方法也倾向于关注这个问题。

惩罚函数的主要问题是，对于一个任意问题，惩罚函数中所采用的理想惩罚因子不能先验地知道。如果采用的惩罚系数过高或过低，那么就会出现以下问题。如果惩罚过高，而最优值位于可行区域的边界，EA 将很快被推入可行区域内，无法再向边界与不可行区域移动。另一方面，如果惩罚过低，由于惩罚对目标函数的影响可以忽略不计，因此大量的搜索时间将被用于探索不可行区域。

2.2.2 约束和目标分离方法

约束和目标分离的方法是当前进化算法中最广泛使用的一类约束处理机制。与罚函数将目标函数和约束函数结合起来处理不同，约束和目标分离方法的核心是将目标和约束分开进行处理，一方面构建新的罚函数的参数设置问题是可以有效避免的，另一方面也充分考虑到了目标函数的优化以及约束的有效处理。

1) 约束支配原则

约束支配原则 (CDP) 是 Deb 教授提出的^[62]，是目前启发式算法中最常用的约束处理机制。在这种方法中，采用以下规则对个体进行比较。其一，可行个体总是要优于不可行个体；其二，当两个个体都是可行解时，具有更好适应度值的解优于适应度差的解；其三，当两个解都是不可行解时，具有约束违反值小的解要优于约束违反值大的解。

$$\varphi(x) = f_{worst} + \left[\sum_{i=1}^{i=p} \text{Max}(0, g_i(x)) \right] \quad (2-13)$$

其中 $\varphi(x)$ 表示个体约束违反值的大小， f_{worst} 代表最差可行个体的适应度。 $g_i(x)$ 是一个不等式约束，所有的约束都可转化为不等式约束。

2) Epsilon 处理方法

Epsilon 约束处理方法，由 Takahama 和 Sakai 在 Deb 教授提出的 CDP 原则上提出的^[63]，它通过 ε 值的大小来控制约束放松。对于两个解 x^1 和 x^2 ， ϕ^1 和 ϕ^2 分别是对应的约束违反值， f^1 和 f^2 分别是对应的目标函数值，Epsilon 比较准则定义如下：

$$(f^1, \phi^1) <_{\varepsilon} (f^2, \phi^2) \Leftrightarrow \begin{cases} f^1 < f^2, & \text{if } \phi^1, \phi^2 \leq \varepsilon \\ f^1 < f^2, & \text{if } \phi^1 = \phi^2 \\ \phi^1 < \phi^2, & \text{otherwise} \end{cases} \quad (2-14)$$

$$(f^1, \phi^1) \leq_{\varepsilon} (f^2, \phi^2) \Leftrightarrow \begin{cases} f^1 \leq f^2, \text{ if } \phi^1, \phi^2 \leq \varepsilon \\ f^1 \leq f^2, \text{ if } \phi^1 = \phi^2 \\ \phi^1 \leq \phi^2, \text{ otherwise} \end{cases} \quad (2-15)$$

如果两个解的约束违反都小于 ε 时, 目标函数值小的解是优于目标函数值大的解; 如果两个解的约束违反相等时, 同样的, 目标函数值小的解优于目标函数值大的解; 否则, 约束违反值小的解要比约束违反大的解更优。值得注意的是, 当 $\varepsilon = 0$ 时, epsilon 比较方法等同于 CDP 处理原则; 当 $\varepsilon = \infty$ 时, 不需要考虑任何约束, 意味着任意两个解的比较等同于对目标函数值的非支配排序。

3) 随机排序法

Runarsson 和 Yao 提出了随机排序法 (Stochastic Ranking, SR)^[64], 这类方法的优势是不需要定义任何惩罚因子, 而是基于气泡排序算法的随机排序过程, 通过一个用户定义参数 P_f 来达到目标和惩罚函数的平衡。对于任何一对相邻的两个个体, 如果这两个个体都是可行的, 那么根据目标函数比较它们的概率是一样的, 通过直接比较个体的适应度值, 适应度值好的个体则保留进入下一代。否则, 即当两个个体有一个不是可行区域的解或都是不可行解时, 设置一个概率 P_f , 以概率 P_f 按照目标适应度值大小对解进行排序, 保留适应度好的个体, 以概率 $1 - P_f$ 按照约束违反程度对解进行排序, 保留约束违反程度低的个体。

以上三种目标与约束分离的方法是将种群分为可行和不可行部分, 然后通过不同的方法比较来保留精英个体。但在解决不可行区域比例较低的 CMOPs 由于没有把不可行的解处理得最好, 所以在某些多目标约束问题上并不能取得非常满意的结果。

2.2.3 混合方法

没有一个最先进的约束处理方法 (Constraint Handling Method, CHM) 能够在每个问题上都能胜过其他所有的 CHM, 因为每个 CHM 都有自己的优势, 或者只适合于某一问题。基于这一点, 为了利用每个 CHM 的优势, 不同 CHM 的组合逐渐得到了新的发展, 也就是混合方法。这类方法结合几种不同约束处理方法对约束进行处理, 通过应用几种现有的约束处理方法进行综合进而改善算法的性能。

自适应均衡模型^[65] (Adaptive Trade-off Model, ATM) 将种群的进化过程分为了三个不同的阶段, 且不同的阶段使用了包括多目标方法和自适应罚函数法两种不一样的约束处理机制, 从而提高算法的性能。

约束处理技术的集成框架^[66] (Ensemble of Constraint Handling Techniques, ECHT) 应用了多种群策略的方式, 在不同的种群中使用不同的约束处理机制, 且种群之间通过相互学习处理复杂的搜索情景。

为了进一步提高 CHMs 的组合性能, Hsieh 等人将罚函数和 ε 比较方法结合在一起开发了一种混合约束处理机制^[67], 它将每个约束的 ε 值分别关联起来, 然后在搜索过程中应用惩罚方法在 ε 不可行的区域进行搜索。

Mani 等人提出了双种群的自适应协同方法^[68], 该方法利用了 Deb 提出的可行性规则和自适应惩罚函数两种方法的最佳特点, 每个种群独立使用一种约束处理技术, 在每个周期结束时交换部分种群, 双种群中个体彼此相互指导。通过自适应惩罚因子法可以更好地进行探索, 同时, 通过可行性规则法可以保证最优解的可行性。

如上所述, CHMs 的关键问题是如何在整个搜索过程中处理不可行的个体。低可行比的 CMOPs 的搜索过程有两个阶段, 一是无可行解阶段; 二是包含可行解阶段。一个有效的约束处理技术应考虑两个方面的问题, 其一是利用不可行个体的优势, 引导种群从不同的不可行区域进入可行区域; 其二是尽可能快速接近最优可行区域。在这两个阶段, 约束处理技术需要同时考虑所有目标和约束。但在无可行解的阶段, 现有的约束处理方法往往忽略了个体之间的分布情况, 导致找不到可行解或易陷入局部最优解。基于此, 设置行之有效的约束处理机制对于算法在进化过程中跨越不可行区域, 在搜索空间中有效寻找最优个体具有非常重要的指导意义。

2.3 测试问题集

测试问题对于评估 MOEAs 的质量和设计 MOEAs 都是非常重要的。一方面这些测试问题的来源是抽象自现实世界中的实际问题, 具备一定的现实含义; 另一方面能有效验证算法的综合性能。针对约束多目标优化问题, 近年来也设计了一些测试问题集, 基于本文面向的研究, 所以围绕以下几个广泛使用的约束多目标测试问题集进行简要介绍。

1) DTLZ 测试组件

Deb 等人创建的 DTLZ 基准问题套件与大多数多目标测试问题不同, 该问题可扩展到任何数量的目标, 并且可以有任意数目个决策变量^[69]。因为变量数与目标数易于控制, 所以 DTLZ 函数集被广泛应用于多目标优化问题当中作为标准测试函数。DTLZ 测试套件包括 9 个测试问题。DTLZ1-DTLZ4 的目标归类为可分离的, PF 是连续且非退化的; DTLZ5 和 DTLZ6 是退化的帕累托最优前沿的问题; DTLZ7 在 Pareto 最优集和 Pareto 最优帕累托前沿是断开的。DTLZ8 和 DTLZ9 是后续的扩展。

DTLZ1 问题定义如下:

$$\left\{ \begin{array}{l} \min f_1(X) = \frac{1}{2} x_1 x_2 \cdots x_{M-1} (1 + g(X_M)) \\ \min f_2(X) = \frac{1}{2} x_1 x_2 \cdots (1 - x_{M-1}) (1 + g(X_M)) \\ \vdots \\ \min f_{M-1}(X) = \frac{1}{2} x_1 (1 - x_2) (1 + g(X_M)) \\ \min f_M(X) = \frac{1}{2} (1 - x_1) (1 + g(X_M)) \\ g(X_M) = 100[|X_M| + \sum_{x_i \in X_M} ((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)))] \end{array} \right. \quad (2-16)$$

其中 $x_i \in [0,1] (i=1,2,\dots,n)$ ，决策变量总数为 $n = M+k-1$ ， X_M 表示决策变量后的 $k = n - M + 1$ 个变量。函数 $g(X_M)$ 中的 $|X_M|$ 包含 k 个变量且满足 $g(X_M) \geq 0$ 。

2) MOP1-7 测试集

MOP1-7 是 Liu 提出的 ZDT 和 DTLZ 的修改版本^[70]，修改后的函数中使用的 $g(\mathbf{x})$ 函数与原始版本中的函数不同。根据测试问题集在搜索空间的有偏性或无偏性，即搜索空间中均匀分布的决策向量样本是否映射到目标空间中均匀分布的目标向量集。而 MOP1-7 具有偏向性搜索空间的特性，即目标不平衡特性。

MOP1 问题定义如下：

$$\left\{ \begin{array}{l} \min f_1(\mathbf{x}) = (1 + g(\mathbf{x}))x_1 \\ \min f_2(\mathbf{x}) = (1 + g(\mathbf{x}))(1 - \sqrt{x_1}) \\ \text{where } g(\mathbf{x}) = 2 \sin(\pi x_1) \sum_{i=2}^n (-0.9t_i^2 + |t_i^{0.6}|) \\ t_i = x_i - \sin(0.5\pi x_1) \\ n = 10, \mathbf{x} \in [0,1]^n \end{array} \right. \quad (2-17)$$

对应帕累托前沿是 $f_2 = 1 + \sqrt{f_1}$ ，其中 f_1 的范围是 $[0,1]$ ，对应的帕累托最优解集是 $x_j = \sin(0.5\pi x_1), j=1,\dots,n$ ，其中 x_1 的范围是 $[0,1]$ 。

3) DAS-CMOPs 测试集

Fan 等人针对现有约束多目标测试问题的不足，通过使用三种难度类型表征约束函数，即约束具有多样性困难、可行性困难和收敛性困难^[71]。实现了难度可调、目标和约束可扩展的约束多目标测试问题构建框架，能够全面综合评估约束多目标进化算法在单一难度或多种难度下的性能。

DAS-CMOP1 问题目标定义如下：

$$\left\{ \begin{array}{l} \min \quad f_1(\mathbf{x}) = x_1 + g(\mathbf{x}) \\ \min \quad f_2(\mathbf{x}) = 1 - x_1^2 + g(\mathbf{x}) \\ \text{where} \quad g(\mathbf{x}) = \sum_{j=1}^n (x_j - \sin(0.5\pi x_j)) \\ n = 30, \mathbf{x} \in [0, 1]^n \end{array} \right. \quad (2-18)$$

具有多样性困难、可行性困难、收敛性困难的约束函数：

$$\left\{ \begin{array}{l} c_1(\mathbf{x}) = \sin(a\pi x_1) - b \geq 0 \\ c_2(\mathbf{x}) = (e - g(\mathbf{x}))(g(\mathbf{x}) - d) \geq 0 \\ c_{k+2}(\mathbf{x}) = ((f_1 - p_k)\cos\theta_k - (f_2 - q_k)\sin\theta_k)^2 / a_k^2 \\ \quad + ((f_1 - p_k)\sin\theta_k - (f_2 - q_k)\cos\theta_k)^2 / b_k^2 \geq r \\ p_k = [0, 1, 0, 1, 2, 0, 1, 2, 3] \\ q_k = [1.5, 0.5, 2.5, 1.5, 0.5, 3.5, 2.5, 1.5, 0.5] \\ a_k^2 = 0.3, b_k^2 = 1.2, \theta_k = -0.25\pi \\ a = 20, d = 0.5, \eta = (b+1)/2, \xi = \exp(d - e), \gamma = 2r \end{array} \right. \quad (2-19)$$

其中 $c_1(\mathbf{x})$ 是多样性约束， $a > 0$ ， $b \in [-1, 1]$ ， η 是多样性难度等级参数。 $c_2(\mathbf{x})$ 是可行性约束， $\zeta \in [0, 1]$ 代表难度等级。 $c_k(\mathbf{x})$ 是收敛性约束， γ 表示难度等级，其范围是 $[0, 1]$ 。

从多目标优化的早期一直到现在，可扩展的测试问题也一直备受关注，可扩展测试问题的不同特点促进了对不同机制形成多目标方法的研究和评价。但是，测试问题之间的属性差异很大，且不同问题之间可以互补。这样，多目标方法就可以在不同的问题上进行不同特征的测试，表现良好的方法就会成为解决现实世界问题的良好候选者。因此，一个多目标测试问题应该包括各种特性，以模拟在现实世界问题中观察到的特性。

2.4 本章小结

本章首先主要围绕多目标约束优化问题展开了介绍，引入了 Pareto 支配关系、Pareto 最优解集、Pareto 前沿等与多目标优化问题有关的概念。然后介绍了针对处理约束多目标优化问题的现有主流约束处理方法。最后简要介绍了一些常用的约束多目标测试问题集，为之后展开研究提供基础理论依据。

第 3 章 M2M-IEpsilon 约束多目标进化算法

对比单目标优化问题、无约束优化问题，约束多目标优化问题不仅仅是要同时优化多个相互冲突的目标，还必须考虑由于约束存在带来的系列影响。由于约束条件的存在，增加了搜索空间的复杂度，比如不可行区域较大的情况，如何高效探索可行区域对于算法来说是非常重要的；其次现有多目标进化算法往往倾向于不考虑约束而单独讨论算法的收敛性、多样性，在一定程度上并不能完全反应算法的有效性。所以综合考虑、设计约束多目标进化算法，应常常从可行性、收敛性及多样性之间的平衡作为算法设计的出发点。本章从这些点出发，重点研究有效处理约束和目标的约束多目标优化进化算法的设计。

近年来，不平衡多目标优化问题也逐渐得到学者们的关注。在一些实际的优化问题中，目标是不平衡的。而现实世界的优化问题通常也有一系列的约束，比如机械臂的设计、焊接梁等实际工程问题；无线传感器网络、云计算网络等通信计算领域的实际优化问题等。因此，具有不平衡目标的 CMOPs 问题值得科研工作者深入探究。在本文中，重点关注解决约束同时具有收敛性困难和多样性困难的不平衡 CMOPs，为处理这类不平衡的 CMOPs，本文提出了一种混合 CMOEAs，命名为 M2M-IEpsilon。即在 M2M 框架中嵌入 IEpsilon 约束处理方法来处理约束，它分别采用了两种机制 M2M 和 IEpsilon。首先，采用一种基于分解的方法（M2M）来处理不平衡的目标，此框架有助于维持工作种群的多样性。其次，采用了 IEpsilon 约束处理方法来提高算法的收敛性和多样性。在 M2M-IEpsilon 中，这两种机制结合用来解决约束同时具有多样性难度和收敛性难度的不平衡 CMOPs，使算法获得较好的性能效果。另外，为有效测试 M2M-IEpsilon 的性能和效率，设计了一组新的约束测试问题集，这类问题集的统一特性是，目标不平衡且约束同时具有多样性难度和收敛性难度，称之为 ICD-CMOPs。在本章中，首先介绍文章使用的两种关键技术，一是基于分解的多目标进化算法，二是改进的 Epsilon 约束处理方法，然后详细介绍 M2M-IEpsilon 算法的核心框架。最后介绍了提出的约束同时具有多样性难度和收敛性难度的不平衡 CMOPs，即 ICD-CMOPs。

3.1 M2M 分解方法

M2M 方法是刘海林教授在近几年提出的基于分解的进化算法^[70]，众多研究表明该方法在解决不平衡多目标优化问题上具有极大优势，目前已经成为独立的一类分解方法并得到了越来越多的应用。M2M 方法的主要思想是将一个多目标优化问题按照

方向向量分解成一组简单的多目标优化子问题。这些子问题以协作的方式被解决，有助于维持工作种群的多样性。

在 M2M 中，分解是在目标空间进行的。具体来说，在目标空间的第一个象限上，目标空间 \mathbf{R}_+^m 中的 J 个方向向量 v^1, \dots, v^j 被选择，方向向量在目标空间是均匀分布的。然后目标空间被划分为 J 个子区域 P_1, \dots, P_j 。子区域 P_j 定义如下所示：

$$P_j = \{u \in \mathbf{R}_+^m \mid \langle u, v^j \rangle \leq \langle u, v^i \rangle \text{ for any } i = 1, \dots, J\} \quad (3-1)$$

其中 $\langle u, v^i \rangle$ 是向量 u 和方向向量 v^i 之间的最小夹角。为了在每个子区域搜索最优解，M2M 将种群分解成 J 个子种群，每个子种群对应一个多目标子问题。子问题可以表示为公式 3-2：

$$\begin{cases} \text{minimize } \mathbf{F}(x) = (f_1(x), \dots, f_m(x)) \\ \text{subject to } x \in \prod_{i=1}^n [a_i, b_i] \\ F(x) \in P_j \end{cases} \quad (3-2)$$

图 3-1 显示了使用 M2M 分解方法的种群分解结果，目标数量为 2 且子区域的数目 $J = 5$ 时。如下图所示，目标空间被分为 5 个子区域，每个子区域有 S 个个体， v^1, \dots, v^5 是五个均匀分布的方向向量。通过以上分解，可以知道每一个个体都只能属于一个子区域，即一个子种群。这意味着每个子种群是相互独立的，每个子种群的进化不会影响其他子种群。这对于维持种群多样性方面发挥了重要的影响，尤其是对于解决不平衡多目标优化问题是非常具有优越性的。

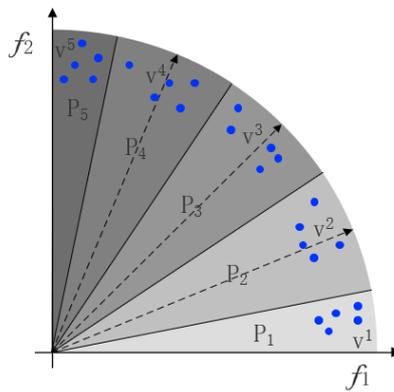


图 3-1 M2M 种群分解方法

3.2 IEpsilon 约束处理方法

原有的 ϵ 约束处理方法通过动态的调整 ϵ 值可以控制约束放松，但 ϵ 值的设定在进化过程中总是减少的，在 ϵ 小于不可行解中约束违反最小值的条件下，Epsilon 约

束处理方法的表现与 CDP 一样。不可行解个体之间的比较仅仅基于约束违反，不可行解较多的情况下，意味着不能很好地保证群体的收敛性，这不太适合解决具有大不可行区域的 CMOPs。基于此，为了克服 ε 约束处理方法在处理大不可行区域的缺点，改进的 ε 约束处理方法最近被提出。

改进的 ε 约束处理方法，称为 IEpsilon^[72]，可以根据可行解在当前种群中的比例动态调整约束违反的松弛，帮助种群跨越较大的不可行区域。关于 IEpsilon 的具体设置说明如下。

$$\varepsilon(k) = \begin{cases} \text{rule1: } \phi(x^\theta), & \text{if } k = 0 \\ \text{rule2: } (1-\tau)\varepsilon(k-1), & \text{if } r_k < \alpha \text{ and } k < T_c \\ \text{rule3: } (1+\tau)\phi_{max}, & \text{if } r_k \geq \alpha \text{ and } k < T_c \\ \text{rule4: } 0, & \text{if } k \geq T_c \end{cases} \quad (3-3)$$

其中 $\varepsilon(k)$ 是第 k 代的 ε 值。 $\phi_k(x^\theta)$ 是初始种群中最高 θ 个体的总体约束违反， r_k 是第 k 代中可行解占总解的比例。 α 用于可行区域和不可行区域之间搜索偏好的控制，其范围为 $[0,1]$ 。 T_c 是控制代数， τ 用来控制减小约束松弛的速度，其范围为 $[0,1]$ 。如果 $r_k < \alpha$ 且 $k < T_c$ ，根据规则 2 来设置 $\varepsilon(k)$ 的值，设置为 $(1-\tau)\varepsilon(k-1)$ 并呈现指数递减，能有助于增强在可行区域的探索。此外， τ 还用于控制规则 3 中的比例因子，增大约束松弛的速度，如果 $r_k \geq \alpha$ 且 $k < T_c$ ， $\varepsilon(k)$ 则设置为 $(1+\tau)\phi_{max}$ ，用来加强不可行区域的探索。 ϕ_{max} 表示到目前为止发现的最大总体约束违反。当 $k > T_c$ ， $\varepsilon(k)$ 设置为零，以专注于寻找可行的解决方案，如规则 4 所示。

IEpsilon 约束处理方法能够在探索可行区域和不可行区域之间保持良好的平衡，根据种群的状态自适应的调整 ε 值，有能力解决约束同时具有收敛性难度和多样性难度的 CMOPs。

3.3 M2M-IEpsilon

基于上述 M2M 分解方法和 IEpsilon 约束处理技术的特性，本节试图将 IEpsilon 约束处理机制嵌入 M2M 算法框架中，提出了一种新的约束多目标进化算法—M2M-IEpsilon，旨在解决约束同时具有多样性难度和收敛性难度的不平衡 CMOPs。在本节中，详述了 M2M-IEpsilon 约束多目标进化算法的核心框架。M2M-IEpsilon 整体框架流程如算法 3-1 所示，算法框架包括了几个主要步骤：种群初始化、子代解的生成、更新 ε 的值、选解、算法终止等几个主要关键步骤。具体步骤的情况将在后面小节详细给出描述。

3.3.1 M2M-IEpsilon 的算法框架

算法 3-1 M2M-IEpsilon 算法框架

输入:

- J : 子问题的数目;
- J 个单位方向向量: v^1, \dots, v^j ;
- S : 子种群的大小;
- Q : 个体解的集合以及对应函数值;
- T_{\max} : 最大代数;
- T_c : $\varepsilon(k)$ 的控制代数;
- IEpsilon 中的参数 α 、 τ ;

输出:可行非支配解集。

- 1 初始化:
- 2 根据算法 3-2 将种群分解为 J 个子种群 P_1, \dots, P_J , 每个子种群由 S 个个体组成。
- 3 根据等式 3-3 初始化 $\varepsilon(0)$, r_k 和 ϕ_{\max} ;
- 4 **while** $gen \leq T_{\max}$ **do**
- 5 **for** $j \leftarrow 1$ **to** J **do**
- 6 **foreach** $x \in P_j$ **do**
- 7 从 P_j 中随机选择个体 y ;
- 8 个体 x 和个体 y 统一应用遗传操作生成新的解 z ;
- 9 计算 $F(z)$;
- 10 $R := R \cup \{z\}$;
- 11 **end**
- 12 $Q := R \cup (\cup_{j=1}^J P_j)$;
- 13 根据等式 3-1 和等式 3-2 使用 Q 来设置 P_1, \dots, P_J ;
- 14 **end**
- 15 **if** $k > 0$ **then**
- 16 $\varepsilon(k) = \text{UpdateEpsilon}(\tau, r_k, \alpha, T_c, k)$;
- 17 **end**
- 18 **for** $j \leftarrow 1$ **to** J **do**
- 19 $P_{j_{\text{fea}}} = \{y \in P_j \mid \phi(y) < \varepsilon(k)\}$;
- 20 $P_{j_{\text{inf}}} = \{y \in P_j \mid \phi(y) \geq \varepsilon(k)\}$;

```

21         if  $|P_j| \leq S$  then
22             随机选择  $|S - P_j|$  加入  $P_j$  中;
23         end
24         if  $P_j > S$  then
25             if  $|P_{j_{fea}}| \geq S$  then
26                 使用 NSGA-II 从  $P_{j_{fea}}$  选择  $S$  个解;
27             else if  $|P_{j_{fea}}| < S$  then
28                 根据  $P_{j_{inf}}$  中每个解的约束违反情况以升序进行排序, 并先选择
                 $S - |P_{j_{fea}}|$  个排序靠前解和所有的  $P_{j_{fea}}$  个体进入下一代。
29             end
30         end
31     end
32     gen=gen+1;
33 end
34 输出: 最优解集。

```

3.3.2 种群初始化

在开始阶段, 第 2-3 行是算法初始化。第 2 行, 将工作种群根据 M2M 分解方法分解为 J 个子种群, 子种群与均匀分布的方向向量 v^1, \dots, v^j 相关联, 每个子总体包含 S 个个体, 细节如算法 3-2 所示。在算法 3-2 中, 第 2-7 行根据等式 3-2 和等式 3-2 初始化 P_j , 因此, 获得 J 个子区域 P_1, \dots, P_J 。第 3 行初始化了 M2M-IEpsilon 中的几个参数, 包括 $\varepsilon(0)$ 的值, k 代中可行解的比例 r_k , 以及到目前为止发现的最大总体约束违反 ϕ_{max} 等。

算法 3-2 子种群的分解框架

```

1 Function output = AllocationSub-pop(Q, S, J)
2     for  $j \leftarrow 1$  to  $J$ 
3         根据等式 3-1 和等式 3-2 初始化  $P_j$ ;
4         //  $Q$ : 个体解的集合和对应的函数值;
5         //  $J$ : 子问题的数量;
6         //  $S$ : 子种群的大小;
7     end

```

```

8   return  $P_1, \dots, P_J$ 
9 end

```

3.3.3 子代个体生成

第 5-14 行显示了使用遗传算子为每个子群体产生新子代的过程。对于 P_j 中的每个个体 x ，首先从 P_j 中随机选择个体 y ，运用遗传操作算子生成新的子代 z ，计算对应的函数值，并在第 13 行，根据等式 3-1 和等式 3-2 使用 Q 来设置 P_1, \dots, P_j 。

3.3.4 更新 $\varepsilon(k)$ 的值

第 15-17 行是 $\varepsilon(k)$ 值的更新过程，详见算法 3-3。在算法 3-3 中， $\varepsilon(k)$ 的值根据等式 3-3 中的规则设置。在算法 3-3 中，第 2-3 行，如果可行解的比例 r_k 小于预定义参数 α ，且生成计数器 k 小于控制代数 T_c ，则 $\varepsilon(k)$ 设置为 $(1-\tau)\varepsilon(k-1)$ 。在第 4-5 行中，可行解的比例 r_k 等于或大于预定义参数 α ，生成计数器 k 小于控制代数 T_c ，将 $\varepsilon(k)$ 设置为 $(1+\tau)\phi_{max}$ 。在第 6-7 行中，当生成计数器 k 等于或大于控制代数 T_c 时， $\varepsilon(k)$ 被设置为零。

算法 3-3 更新 Epsilon 的值

```

1 Function  $\varepsilon(k)=UpdateEpsilon(\tau, r_k, \alpha, T_c, k)$ 
2   if  $r_k < \alpha$  and  $k < T_c$  then
3      $\varepsilon(k)=(1-\tau)\varepsilon(k-1)$ 
4   else if  $r_k \geq \alpha$  and  $k < T_c$  then
5      $\varepsilon(k)=(1+\tau)\phi_{max}$ 
6   else if  $k \geq T_c$  then
7      $\varepsilon(k)=0$ 
8   end
9   return  $\varepsilon(k)$ 
10 end

```

3.3.5 选解

第 18-31 行描述了每个子种群的更新过程，根据 $\varepsilon(k)$ 的值定义可行解 $|P_{j_{fea}}|$ （第 19 行所示）和不可行解 $P_{j_{inf}}$ （第 20 行所示）。具体来说，在第 21-23 行，如果子区域 P_j 的个体数小于 S ，则从整个种群中随机选取 $|S - P_j|$ 解，将其加入 P_j 中。第 24-30 行，

如果子区域 P_j 的个体数大于 S ，采用两种策略来选择 S 个解。当 $|P_{j_{\text{fea}}}|$ 个数等于或大于 S 时，我们利用 NSGA-II 从 $|P_{j_{\text{fea}}}|$ 中选取 S 个解。否则，我们首先选择 $|P_{j_{\text{fea}}}|$ 中的所有解到下一代，然后，将 $P_{j_{\text{inf}}}$ 中的所有解都按照约束违反情况进行升序排序，并将 $S - |P_{j_{\text{fea}}}|$ 个约束违反小的解选择进入下一代。

3.4 测试函数设计

由于科学和工程领域中带约束的目标优化问题比比皆是，人们对于约束多目标优化的兴趣越来越大。在相关的研究中，CMOPs 常常作为基准测试算法的重要工具，这进一步推动一些设计新的 CMOPs 的研究工作。在前面的工作中已经提及 M2M-IEpsilon 算法使用的两种机制具有解决约束同时具有多样性难度和收敛性难度的不平衡 CMOPs，而现有的 CMOPs 往往只单独关注目标或者约束，而没有综合去考虑目标空间的复杂性和约束阻碍算法接近帕累托前沿的不可行障碍。基于此，为了验证 M2M-IEpsilon 算法的有效性，本节提出了一组约束同时具有多样性难度和收敛性难度的不平衡 CMOPs，命名为 ICD-CMOPs。

3.4.1 测试问题集设计背景

测试问题对于评估 MOEAs 和设计 MOEAs 都是十分重要的，一个理想的 CMOP 应该同时考虑目标空间的复杂性和接近帕累托前沿的不可行障碍。然而，大多数相关研究只单独关注这两个挑战。例如，大多数可扩展的测试问题(如 DTLZ^[73]、WFG^[74] 和 MaF^[75])只包含边界约束。CTP^[76]和 CF^[77]测试问题集目标空间维度固定，目标数量不可伸缩。综合考虑目标、约束等各方面对约束多目标优化问题的设计已经成为了设计 CMOPs 的一大热点。

DAC-CMOPs 是一类具有难度可控制，目标和约束数量可调的多目标测试问题集^[71]。首次对约束问题的难度进行了定义，即多样性困难、可行性困难和收敛性困难。三种难度类型的约束能够任意组合，构成同时具有多种难度类型的约束多目标测试问题。每种约束类型的难度都可以自由调整，问题可以进行自由定制，能够全面综合评价约束多目标进化算法在单一难度或多种难度下的性能。

Zhou 等人构建了一个新的 16 个可扩展和约束涵盖各种类型的测试问题集^[78]，该框架将决策变量拆分为位置和距离变量，并强制其最优值位于一个非线性超曲面上，从而在位置变量之间和距离变量之间分别引入相互依赖关系。在这个框架中，引入了收敛性难度和多样性难度两种约束条件。第一种在接近最优值的过程中引入了不可行的障碍，同时令位置变量和距离变量相互关联。第二种限制了可行的最优区域，从而可以得到不同形状的 Pareto 前沿。

从以上研究发现，考虑目标、约束等各方面因素对约束多目标优化问题集的设计

也逐渐有了新的发展。实际问题的复杂和多变性，现实问题在优化目标函数时需要满足许多的约束条件，那么综合考虑目标空间的复杂多变性以及不同类别约束导致收敛到帕累托前沿的困难也将成为下一步研究方向。从现有的研究中发现，针对目标不平衡的约束多目标测试问题集的设计较少，基于以上考虑，设计同时具有多样性和收敛性困难的不平衡测试函数对于评价 M2M-IEpsilon 的性能非常有必要，对于研究约束多目标进化算法的发展有积极促进作用。

为了更清楚的了解关于不平衡问题以及约束具有多样性和收敛性困难的 CMOPs, 下文结合图形化的思想来解释这两个概念。

不平衡问题^[79]，如果 PF 的某个特定子集（称“有利”的子集）满足以下条件。即与有利子集对应的优化子问题的复杂度明显低于 PF 的另一部分（“不利”子集）对应的子问题的复杂度。其二，在可行的变量空间中，有利子集的帕累托集（PS）比不利子集的 PS 占主导地位要大得多。在这类问题中，种群在进化过程中，算法往往倾向于收敛到 PF 容易满足约束的部分，这就意味着有利的 PF 比较容易找到，而相对应的 PS 占据了大部分的可行搜索空间，进而导致算法很难找到整个 PF。

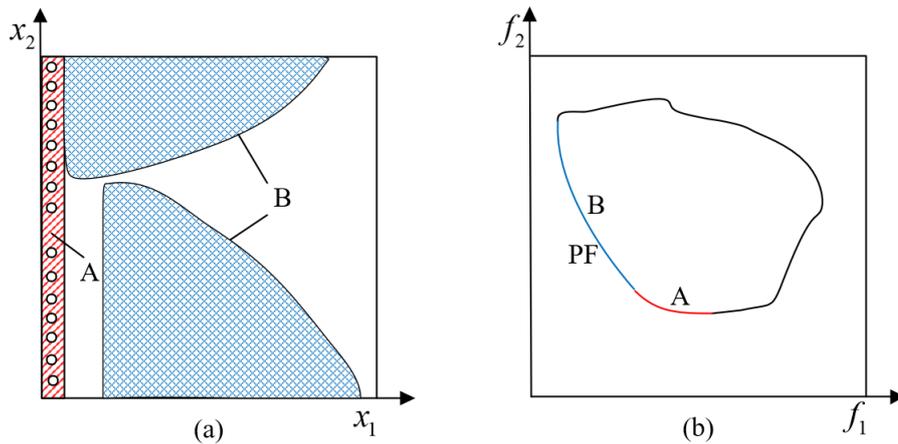


图 3-2 不平衡搜索空间示例

图 3-2(a)绘制了一个不平衡 MOP 的搜索空间。其中 A 和 B 分别代表不同的区域。区域 A 在搜索空间中对应的是一个非常小的区域，A 中的每一个点都支配着区域 B 中的任何一个点。完整的区域 A 映射到目标空间图 3-2(b)成为帕累托最优解的一部分（红色标记），并且位于 PF 的有利子集中，而区域 B 映射到目标空间图 3-2(b)也同样成为帕累托最优解的一部分（蓝色标记），位于 PF 的不利子集中。所以，大部分 MOEAs 很容易被困在区域 A 中，也就意味着大部分解会更多的收敛于 PF 的有利子集 A 中，从而导致算法很难找到整个 PF，算法的多样性也就缺失了。

通常来讲，具有多样性困难的 CMOPs 的 PF 有许多离散的片段。而具有收敛性困难的 CMOPs 阻碍了 CMOPs 向 PFs 的收敛。通常，CMOPs 在获得 PFs 时会

遇到更多的困难，因为它们收敛到 PFs 时，不可行的区域会阻碍它们。

图 3-3 描绘了约束具有多样性难度的不平衡约束多目标测试问题的可行区域（绿色区域标记）、在约束条件下的帕累托前沿（红色曲线）、无约束条件下的帕累托前沿（蓝色曲线）。引入这种约束之后，真实帕累托前沿分段分布。

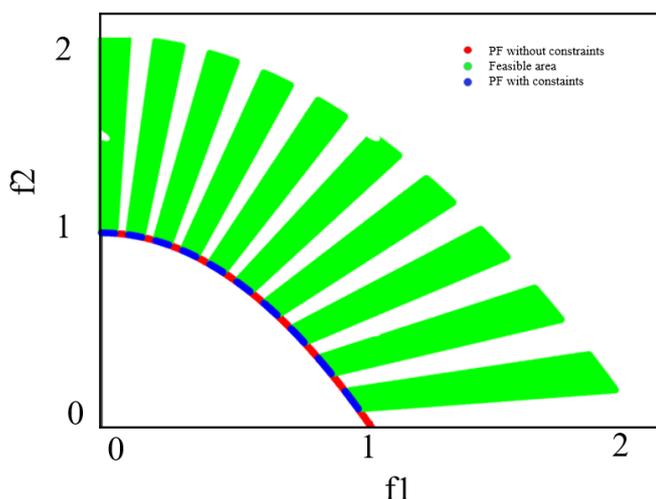


图 3-3 约束具有多样性困难的 CMOP 示例

图 3-4 描绘了约束具有多样性难度和收敛性难度的不平衡约束多目标测试问题的可行区域（绿色区域标记）、在约束条件下的帕累托前沿（红色曲线）、无约束条件下的帕累托前沿（蓝色曲线）。引入这两种约束之后，真实帕累托前沿分段分布且目标搜索空间的可行区域较小，这就导致算法在可行区域和不可行区域的平衡搜索能力的难度加大以及维持算法的多样性和收敛性更加具有难度。

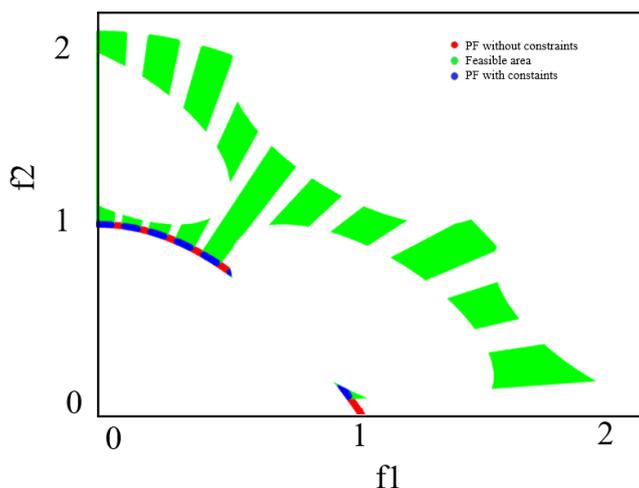


图 3-4 约束具有多样性和收敛性困难的 CMOP 示例

3.4.2 测试问题细节描述

在本章中，主要研究了解决同时具有收敛性困难和多样性困难的不平衡 CMOPs。

因此，依据 MOPs 与 DAS-CMOPs 设计了一组新的目标不平衡的约束多目标优化问题，称为 ICD-CMOPs。其中 ICD-CMOPs 的目标函数与 MOPs^[70] 的目标函数相同，也就意味着它具有同 MOPs 的一般特性，即都是不平衡的。而约束函数来源于 DAC-CMOPs^[71]，具有多样性困难和收敛性困难。由 DAS-CMOPs 可知，对应难度级别可以轻松通过一个三原组 (η, ζ, γ) 来进行设定，分别表示多样性、可行性以及收敛性困难，且三个参数的取值范围为 $(0, 1)$ ，可以根据实际需求自由调整。针对此 ICD-CMOPs 的约束函数问题的设计，本文只考虑了多样性和收敛性难度类型，因此设定了一定的难度级别，分别为 $(0.5, 0, 0.5)$ 。

本文一共构造了 7 个约束测试函数，其中 ICD-CMOP1-5 的目标是二维的，ICD-CMOP6-7 是三维目标函数，7 个约束测试函数的约束都同时具有多样性困难和收敛性困难，难度等级相同。

作为示例，具有不平衡双目标且约束同时具有多样性困难和收敛性困难的 CMOP 定义如下，其中目标函数的公式是：

$$\left\{ \begin{array}{l} \min \quad f_1(\mathbf{x}) = (1 + g(\mathbf{x}))x_1 \\ \min \quad f_2(\mathbf{x}) = (1 + g(\mathbf{x}))(1 - x_1^2) \\ \text{where} \quad g(\mathbf{x}) = 10 \sin(\pi x_1) \sum_{i=2}^n \left(\frac{|t_i|}{1 + e^{5|t_i|}} \right) \\ \quad \quad \quad t_i = x_i - \sin(0.5\pi x_1) \\ \quad \quad \quad n = 10, \mathbf{x} \in [0, 1]^n \end{array} \right. \quad (3-4)$$

对应帕累托前沿是 $f_2 = 1 + \sqrt{f_1}$ ，其中 f_1 的范围是 $[0, 1]$ ，对应的帕累托最优解集是 $x_j = \sin(0.5\pi x_1), j = 1, \dots, n$ ，其中 x_1 的范围是 $[0, 1]$ 。

约束函数公式是：

$$\left\{ \begin{array}{l} c_1(\mathbf{x}) = \sin(a\pi x_1) - b \geq 0 \\ c_k(\mathbf{x}) = ((f_1 - p_k)\cos\theta_k - (f_2 - q_k)\sin\theta_k)^2 / a_k^2 \\ \quad + ((f_1 - p_k)\sin\theta_k - (f_2 - q_k)\cos\theta_k)^2 / b_k^2 \geq r \\ a = 20, b = 0 \\ p_k = [0, 1, 0, 1, 2, 0, 1, 2, 3] \\ q_k = [1.5, 0.5, 2.5, 1.5, 0.5, 3.5, 2.5, 1.5, 0.5] \\ a_k^2 = 0.4, b_k^2 = 1.6, \theta_k = -0.25\pi \\ c = 20, n = 30, x_i \in [0, 1], k = 1, \dots, 9 \end{array} \right. \quad (3-5)$$

其中 $c_1(\mathbf{x})$ 是多样性约束， $a > 0, b \in [-1, 1]$ ，指定难度等级参数 η 决定了 b ，这里 $b = 2\eta - 1$ ，帕累托前沿断开的段数由 a 控制。此文中设定断开段数为 $a = 10$ ，多样性难度为 0.5，所以 $\eta = 0.5, b = 0$ 。 $c_k(\mathbf{x})$ 是收敛性约束，限制了目标的可达区域。 γ 表示难度等级，

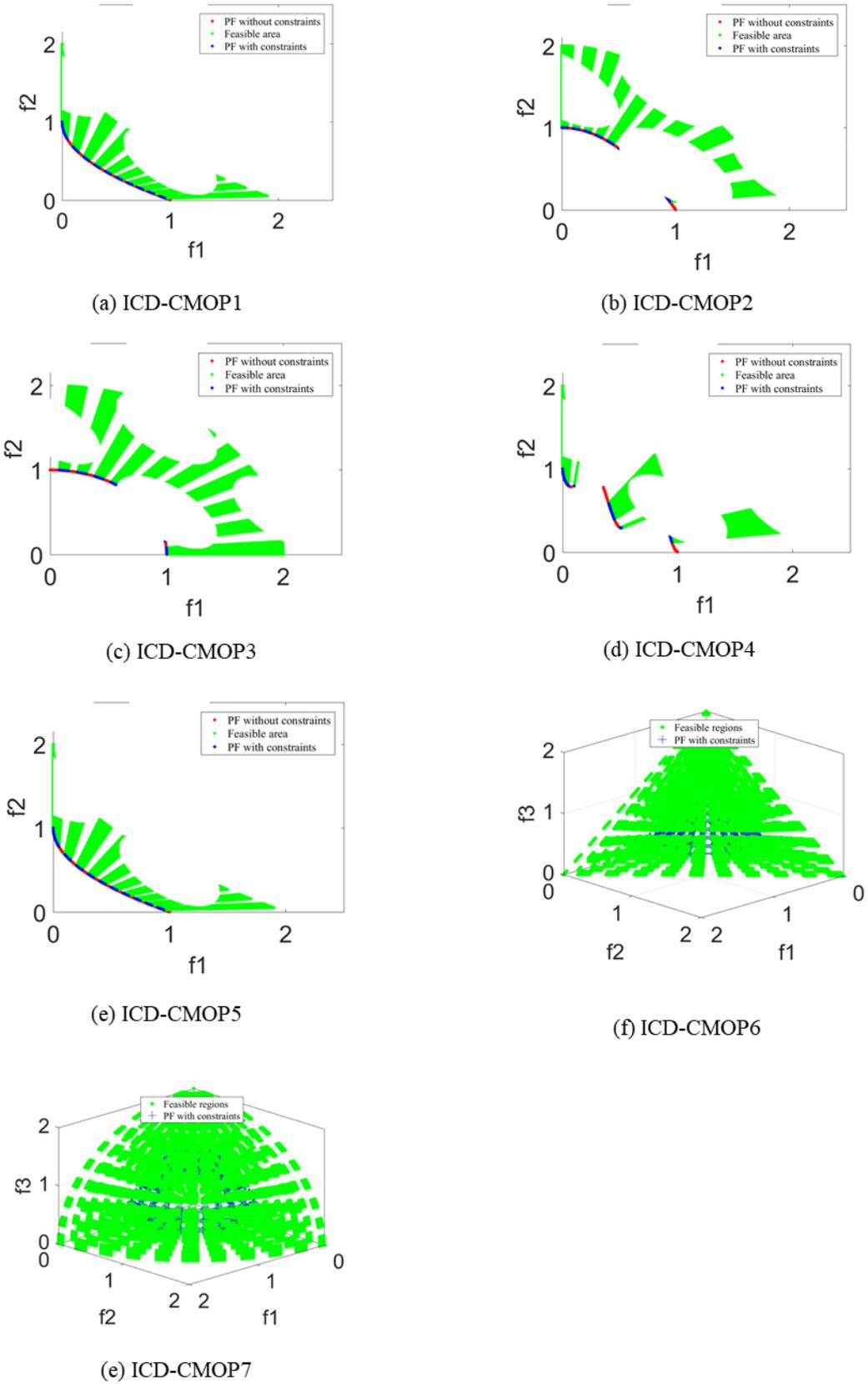


图 3-5 目标空间上测试问题 ICD-CMOP1-7 的可行区域、无约束 PF、有约束 PF 的分布情况

其范围是[0,1], 本文设定难度等级为 0.5。随着 η 、 γ 设定值的增加, 多样性难度和收敛性难度也随之增加, 对应真实帕累托前沿的断开数会更多和不可行区域也会变大。

为了更直观的观察到这七个目标不平衡且约束具约束同时具有多样性困难和收敛性困难的测试函数, 图 3-5 展示了 7 个函数在目标空间的可行区域、不可行区域、约束 PF、无约束 PF 的分布情况。具体目标函数和约束函数详见附录表格中。

为了更好地理解 M2M-IEpsilon 处理 ICD-CMOPs 的过程, 本文用例子来说明 M2M-IEpsilon 的优越性, 以 ICD-CMOP2 问题为例。如果没有使用 IEpsilon 约束处理机制, 以 CM2M 为例, 如图 3-6(a)所示, 是一个种群的初始状态, 种群中的大多数个体远离实际的 PF, 绝大部分个体都位于不可行区域。由于没有使用 IEpsilon 约束处理机制, 子种群一旦进入不可行区域之后便难以穿过不可行区。如图 3-6(b)所展示的一样, 大部分个体都陷入了不可行区域导致个体很难收敛到真实的 PF。如果没有使用 M2M 分解方法, 如图 3-7 展示了 MOEA/D-IEpsilon 的搜索过程, 图 3-7(a)是种群的初始状态, 由于没有使用 M2M 机制, 进化过程中部分个体难以穿越不可行区域且容易陷入容易满足约束的地方, 导致个体收敛到 PF 的有利子集中, 从而没办法找到整个 PF, 算法的多样性也就难以维持。如图 3-7(b)所示, 大部分个体陷入了不可行区域且个体收敛到部分 PF。当使用 M2M 方法和 IEpsilon 约束处理方法之后。首先, 通过目标空间的 5 个方向向量 v^1, \dots, v^j , 将目标空间分为了五个子区域 P_1, \dots, P_j , 图 3-8(a)是种群的初始状态, 大多数个体都位于不可行区域。通过使用这两种机制, 大多数个体跨越了不可行区, 如图 3-8(b)所示。一方面 M2M 方法有效保证个体均匀分布在每个子区域, 另一方面 IEpsilon 约束处理方法帮助种群跨越不可行区域, 共同维持多样性和收敛性的平衡, 使得个体均匀分布并收敛到整个帕累托前沿。

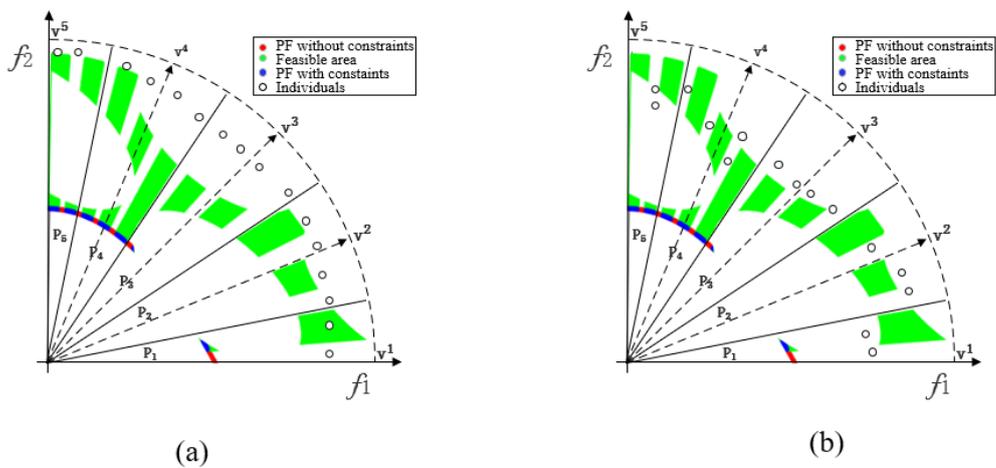


图 3-6 CM2M 方法搜索过程示例

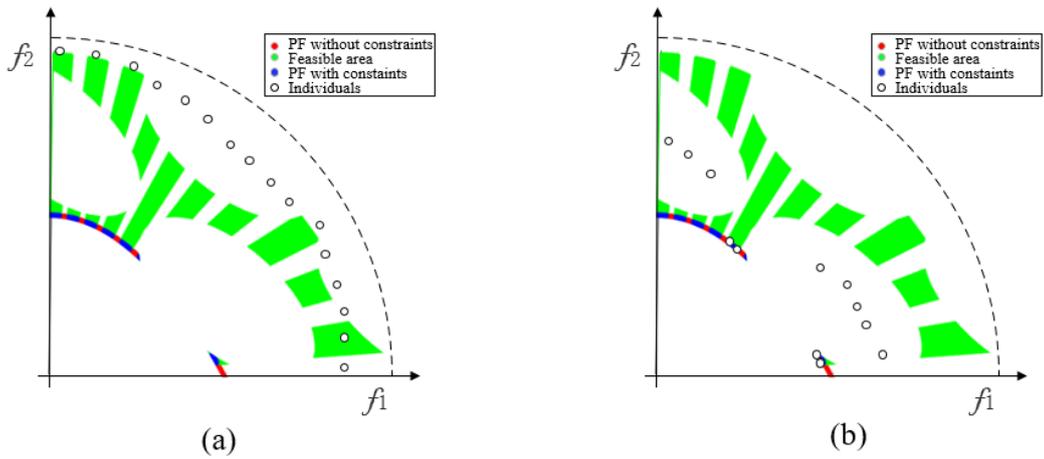


图 3-7 MOEAD-IEpsilon 方法搜索过程示例

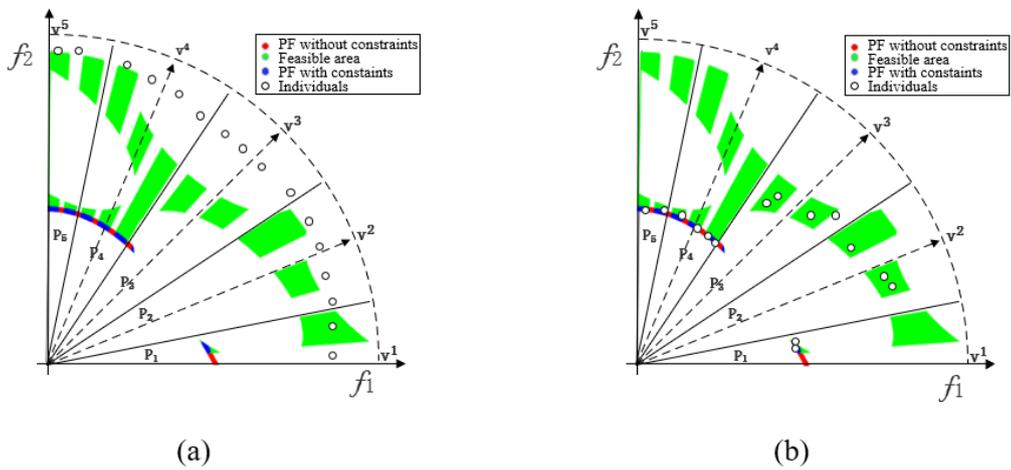


图 3-8 M2M-IEpsilon 方法搜索过程示例

3.5 本章小结

本章首先提出了一个改进的 Epsilon 方法嵌入 M2M 框架中的约束多目标进化算法 (M2M-IEpsilon)，用来解决约束同时存在多样性难度和收敛性难度的不平衡 CMOPs 问题。其次，为了有效评估所提出算法 M2M-IEpsilon 的性能，针对现有 CMOPs 只单独关注目标或者约束等单一性方面的难度，综合设计了目标不平衡且约束同时具有多样性难度和收敛性难度的 CMOPs，即 ICD-CMOPs。下文将通过实验验证 M2M-IEpsilon 在解决此类约束多目标优化问题上的有效性。

第 4 章 实验研究

本章中，对提出的算法 M2M-IEpsilon 的性能进行了综合实验研究，本章实验均在 win 10 系统下 Matlab（版本 2018b）环境中对算法进行了仿真，通过对比实验验证了提出算法的性能。

4.1 实验设置

为了评估提出 M2M-IEpsilon 算法的性能，M2M-IEpsilon 与 6 个最先进的约束多目标进化算法，包括 CM2M^[80]，CM2M2^[81]，NSGA-II-CDP^[38]，MOEA / D-CDP^[82]，PPS-MOEA / D^[83]和 MOEA / D-IEpsilon^[72]在 ICD-CMOP1-7 上进行实验测试。值得注意的是，为保证实验的公平，7 种算法使用相同处理机制时的参数设置均相同，具体参数设置如下：

- 1) 种群大小： $N = 300$ 。
- 2) 变异概率 $P_m = 1/n$ （ n 是决策变量的数目），交叉概率 $CR = 1.0$ 。
- 3) CM2M2 的参数设置：不可行权重 N_1 设置为 90，可行权重 N_2 设置为 210。
- 4) 终止条件：每个算法独立运行 30 次且当函数评估次数达到 300,000 次时停止这些算法。
- 5) 对于双目标 CMOPs， J 设置为 10，在 M2M-IEpsilon、CM2M 和 CM2M2 方法中，其中 J 是子问题的数量。
- 6) 对于三目标 CMOPs， J 设置为 15，在 M2M-IEpsilon、CM2M 和 CM2M2，其中 J 是子问题的数量。
- 7) M2M-IEpsilon 和 MOEA/D-IEpsilon 中 IEpsilon 方法的参数设置： $T_c = 800$ ， $\alpha = 0.95$ ， $\tau = 0.1$ ， $\theta = 0.05N$ 。
- 8) PPS-MOEA/D 的参数设置： $T_c = 800$ ， $\alpha = 0.95$ ， $\tau = 0.1$ ， $cp = 2$ ， $l = 20$ 。
- 9) MOEA/D-CDP 的参数设置： $T = 30$ ， $n_r = 2$ 。

4.2 比较算法

为了对所提出算法 M2M-IEpsilon 的性能进行有效评估，采用了其余 6 种 CMOEAs 进行算法实验对比，包括 CM2M^[80]、CM2M^[81]、NSGA-II-CDP^[38]、MOEA/D-CDP^[82]、PPS-MOEA/D^[83]和 MOEA/D-IEpsilon^[72]，下面对每种算法进行简要描述。

- 1) CM2M: CM2M 把一个多目标优化问题分解为一组优化子问题，这些子问题

同时进行优化。在 CM2M 中，采用一种新的边界搜索和存档方法来处理约束问题。具体地说，每个子问题对应于一个包含子种群和临时寄存器的存档。在子种群中记录的是目标值较好和违反约束较少的个体，而临时寄存器则是由之前发现的一些个体组成。

2) CM2M2: CM2M2 在 MOEA/D-M2M 框架中使用有向权重求解 CMOPs。在 CM2M2 中，采用了两种有向权重，包括可行权重和不可行权重，这些权重分别分布在可行区域和不可行区域上。利用一组可行权值来寻找一组分布良好的可行非支配解，而利用一组不可行权值来维持一组多样化良好的不可行个体。同时，在进化过程中，这些不可行权值被动态改变为更偏好目标值更好、约束违反更小的不可行个体。

3) NSGA-II-CDP: NSGA-II-CDP 在非支配排序遗传算法中嵌入了 CDP 来处理约束。NSGA-II 是最有效、最常用的 MOEAs 之一。在 NSGA-II-CDP 中，根据非支配排序以及计算拥挤距离来选择下一子代个体。CDP 用于判断任意两个个体之间的支配关系，2.2.2 小结已详细说明 CDP 如何确定两个个体的优劣关系。

4) MOEA/D-CDP: MOEA/D-CDP 在 MOEA/D 中嵌入 CDP 来处理约束。MOEA/D 是最著名的基于分解的 MOEA 方法之一，它把一个多目标优化问题分解为若干个标量优化子问题，并以协作的方式同时对每个子问题进行优化。在 MOEA/D 中，子问题的定义是用权向量来定义的，子问题之间的邻域关系通过算出权向量的欧氏距离来进行确定。同上所述，CDP 用于确定任意两个个体之间的优劣关系。

5) PPS-MOEA/D: PPS-MOEA/D 在 MOEA/D 中嵌入了推拉搜索 (PPS) 来求解 CMOPs。提出的 PPS 算法将搜索过程分为两个不同的阶段：Push 阶段和 Pull 阶段。在 Push 阶段，利用 MOEA/D 进化算法在不考虑任何约束的情况下探索搜索空间，有助于快速跨越不可行区域，从而逼近无约束帕累托前沿。此外，约束违反程度的信息也能够被探测和估计。在 Pull 阶段，采用改进的 Epsilon 约束处理机制并利用探测到的约束违反信息，把整个种群拉回到可行的非支配区域，接近有约束的 PF 中。

6) MOEA/D-IEpsilon: MOEA/D-IEpsilon 将改进的 Epsilon 约束处理方法集成到 MOEA/D 框架中处理约束。在 MOEA/D-IEpsilon 中，基于改进的约束处理方法对子问题进行了更新。

4.3 性能比较

性能指标用于衡量 CMOEAs 获得的非支配解集的质量，在第 1.3 节已经描述了现有的进化算法评价指标。在本文中，使用两种通用的指标，包括反转世代距离 (IGD) 和超体积 (HV) 两种指标去评估比较算法的性能，这两个指标可以同时衡量最终获得的非支配解集的收敛性和多样性。另外，M2M-IEpsilon 和其他 6 种 CMOEAs 之间

在 0.05 显著性水平下的 Wilcoxon 秩和检验结果也用来评估算法的性能，†和‡表示对应算法的性能分别明显低于或优于 M2M-IEpsilon，‘S-D-I’ 分别表示 M2M-IEpsilon 优于、无显著差异或劣于相应的比较 CMOEAs。

表 4-1 M2M-IEpsilon 与其他 6 种算法在 ICD-CMOP1-7 上的 IGD 结果

Test instances		M2M-IEpsilon	CM2M	CM2M2	NSGA-II-CDP	MOEA/D-CDP	PPS-MOEA/D	MOEA/D-IEpsilon
ICD-CMOP1	mean	1.864 E-02	8.498 E-02†	1.486 E-01†	3.231 E-01†	3.116 E-01†	2.972 E-01†	2.622 E-01†
	std	3.435 E-03	3.062 E-02	3.275 E-02	1.288 E-02	4.874 E-02	6.516 E-02	7.197 E-02
ICD-CMOP2	mean	1.206 E-01	2.262 E-01†	1.472 E-01	3.085 E-01†	2.730 E-01†	2.648 E-01†	2.662 E-01†
	std	8.591 E-02	6.088 E-02	6.211 E-02	2.563 E-02	7.499 E-03	5.664 E-03	1.599 E-02
ICD-CMOP3	mean	2.045 E-01	3.224 E-01†	3.984 E-01†	8.539 E-01†	4.846 E-01†	6.365 E-01†	4.438 E-01†
	std	8.403 E-02	8.045 E-02	6.471 E-02	8.117 E-02	9.174 E-02	1.847 E-01	9.821 E-02
ICD-CMOP4	mean	2.156 E-02	2.139 E-01†	8.499 E-02†	3.335 E-01†	2.855 E-01†	2.861 E-01†	2.856 E-01†
	std	5.175 E-02	7.353 E-02	6.053 E-02	2.387 E-02	1.496 E-02	1.216 E-02	4.306 E-02
ICD-CMOP5	mean	4.991 E-02	1.272 E-01†	1.713 E-01†	2.931 E-01†	3.241 E-01†	3.093 E-01†	2.840 E-01†
	std	2.054 E-02	2.968 E-02	2.306 E-02	4.749 E-02	4.252 E-03	1.437 E-02	4.158 E-02
ICD-CMOP6	mean	3.920 E-01	4.110 E-01	3.149 E-01†	8.050 E-01†	7.178 E-01†	3.517 E-01	7.258 E-01†
	std	1.192 E-01	1.182 E-01	1.016 E-02	9.639 E-03	1.550 E-02	4.736 E-02	2.651 E-02
ICD-CMOP7	mean	4.917 E-01	5.442 E-01†	5.350 E-01†	7.466 E-01†	7.190 E-01†	4.000 E-01†	7.242 E-01†
	std	2.782 E-02	4.888 E-02	2.516 E-02	6.895 E-03	9.527 E-03	5.559 E-02	6.622 E-03
Wilcoxon-Test(S-D-I)		—	6-1-0	6-1-0	7-0-0	7-0-0	6-1-0	7-0-0

表 4-1 展示了 M2M-IEpsilon、CM2M、CM2M2、NSGA-II-CDP、MOEA/D-CDP、PPS-MOEA/D 和 MOEA/D-IEpsilon 七种算法在 30 次独立运行之后的 IGD 指标的均

值和方差，黑色加粗代表该算法在其对应问题上取得了较好的 IGD 值。根据表中 Wilcoxon 测试结果表明，在所有 ICD-CMOP1-7 测试集中，M2M-IEpsilon 明显优于 NSGAI-CDP、MOEA/D-CDP 和 MOEA/D-IEpsilon。对于 ICD-CMOP6，M2M-IEpsilon 具有与 CM2M 和 PPS-MOEA/D 相似的性能，即无显著性差异，在其他 6 个测试问题中，M2M-IEpsilon 的性能显著优于 CM2M 和 PPS-MOEA/D。除 ICD-CMOP2 之外，M2M-IEpsilon 在其余的 ICD-CMOPs 上明显比 CM2M2 更好。

表 4-2 M2M-IEpsilon 与其他 6 种算法在 ICD-CMOP1-7 上的 HV 结果

Test instances		M2M-IEpsilon	CM2M	CM2M2	NSGA-II-CDP	MOEA/D-CDP	PPS-MOEA/D	MOEA/D-IEpsilon
ICD-CMOP1	mean	9.921 E-01	9.136 E-01†	8.560 E-01†	5.214 E-01†	5.508 E-01†	5.643 E-01†	6.212 E-01†
	std	4.339 E-03	3.881 E-02	1.966 E-02	2.464 E-02	8.769 E-02	1.151 E-01	1.325 E-01
ICD-CMOP2	mean	4.983 E-01	4.367 E-01†	4.315 E-01†	2.602 E-01†	3.769 E-01†	3.844 E-01†	3.662 E-01†
	std	5.057 E-02	3.027 E-02	3.351 E-02	5.458 E-02	1.931 E-02	1.236 E-02†	4.243 E-02
ICD-CMOP3	mean	3.659 E-01	2.837 E-01†	2.546 E-01†	2.393 E-01†	2.457 E-01†	2.541 E-01	2.393 E-01†
	std	7.724 E-02	6.309 E-02	2.825 E-02	8.469 E-17	2.587 E-02	4.047 E-02†	8.469 E-17
ICD-CMOP4	mean	8.152 E-01	5.844 E-01†	7.245 E-01†	4.131 E-01†	5.108 E-01†	5.037 E-01	4.782 E-01†
	std	6.995 E-02	6.767 E-02	8.048 E-02	3.134 E-02	1.676 E-02	1.044 E-02†	5.834 E-02
ICD-CMOP5	mean	9.472 E-01	8.215 E-01†	8.064 E-01†	6.138 E-01†	5.805 E-01†	5.879 E-01	6.063 E-01†
	std	2.353 E-02	4.121 E-02	1.974 E-02	5.995 E-02	0.000 E+00	7.503 E-03	5.754 E-02
ICD-CMOP6	mean	1.899 E-01	1.728 E-01†	2.058 E-01	7.089 E-03†	8.987 E-03†	4.551 E-01	1.123 E-02†
	std	6.173 E-02	5.808 E-02	9.262 E-03	7.495 E-04	3.959 E-03	3.631 E-02	1.117E-02
ICD-CMOP7	mean	3.224 E-01	2.587 E-01†	2.379 E-01†	1.410 E-01†	1.836 E-01†	5.610 E-01	1.754 E-01†
	std	3.325 E-02	3.566 E-02	2.841 E-02	6.999 E-03	8.906 E-03	3.240 E-02	7.550E-03
Wilcoxon-Test(S-D-I)		—	7-0-0	6-1-0	7-0-0	7-0-0	5-2-0	7-0-0

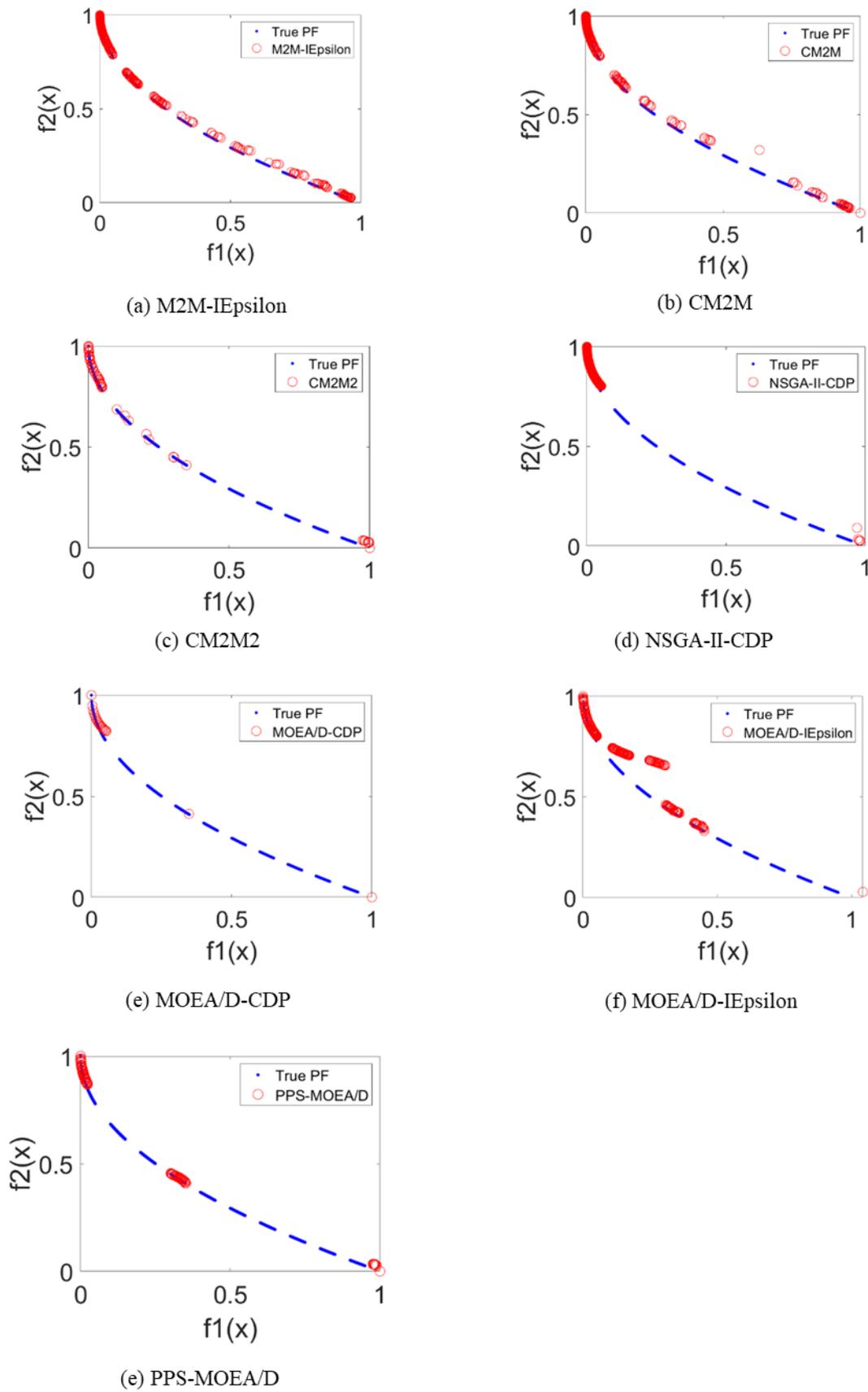


图 4-2 每种算法在 ICD-CMOP1 上获得的非支配解集

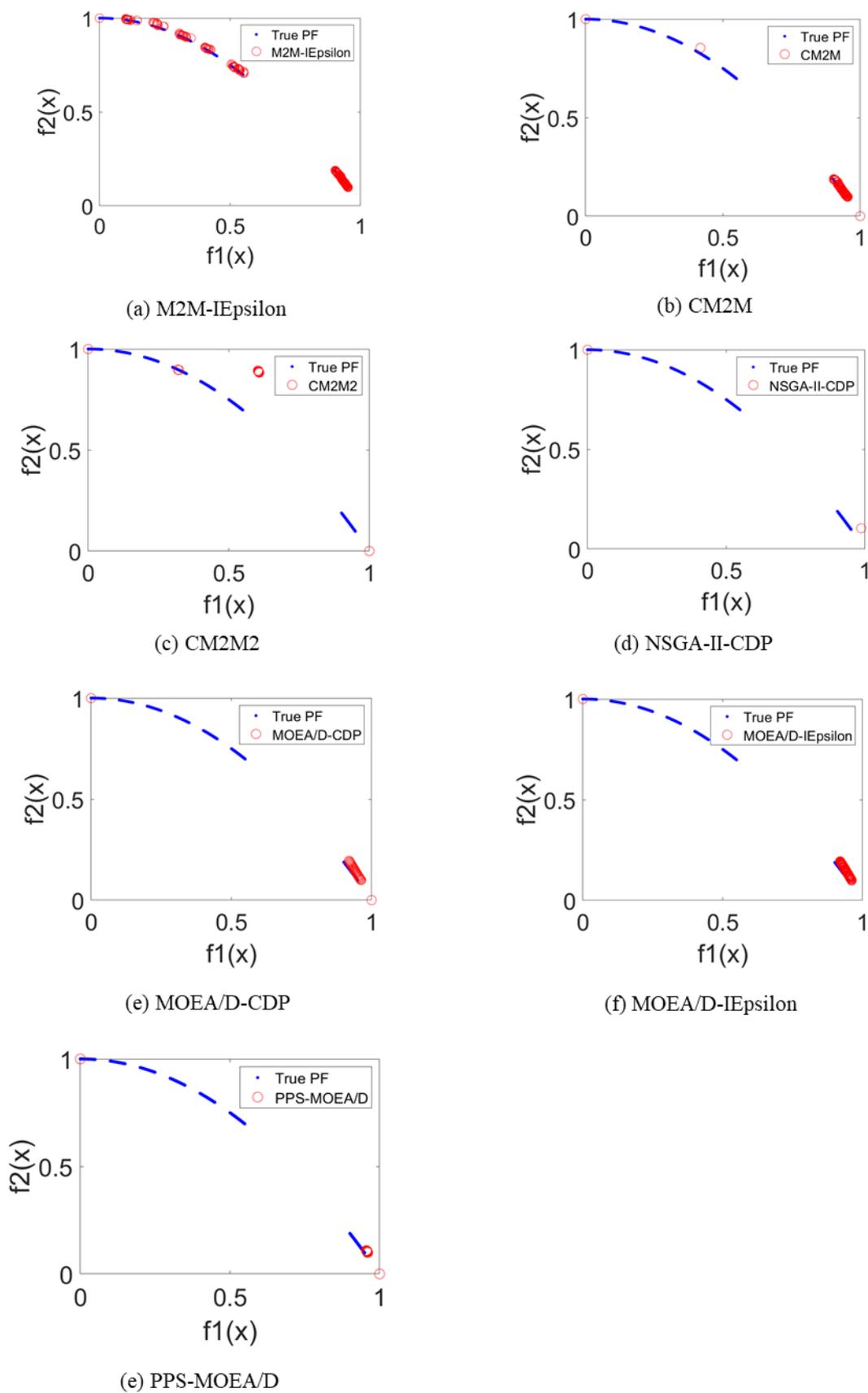


图 4-3 每种算法在 ICD-CMOP2 上获得的非支配解集

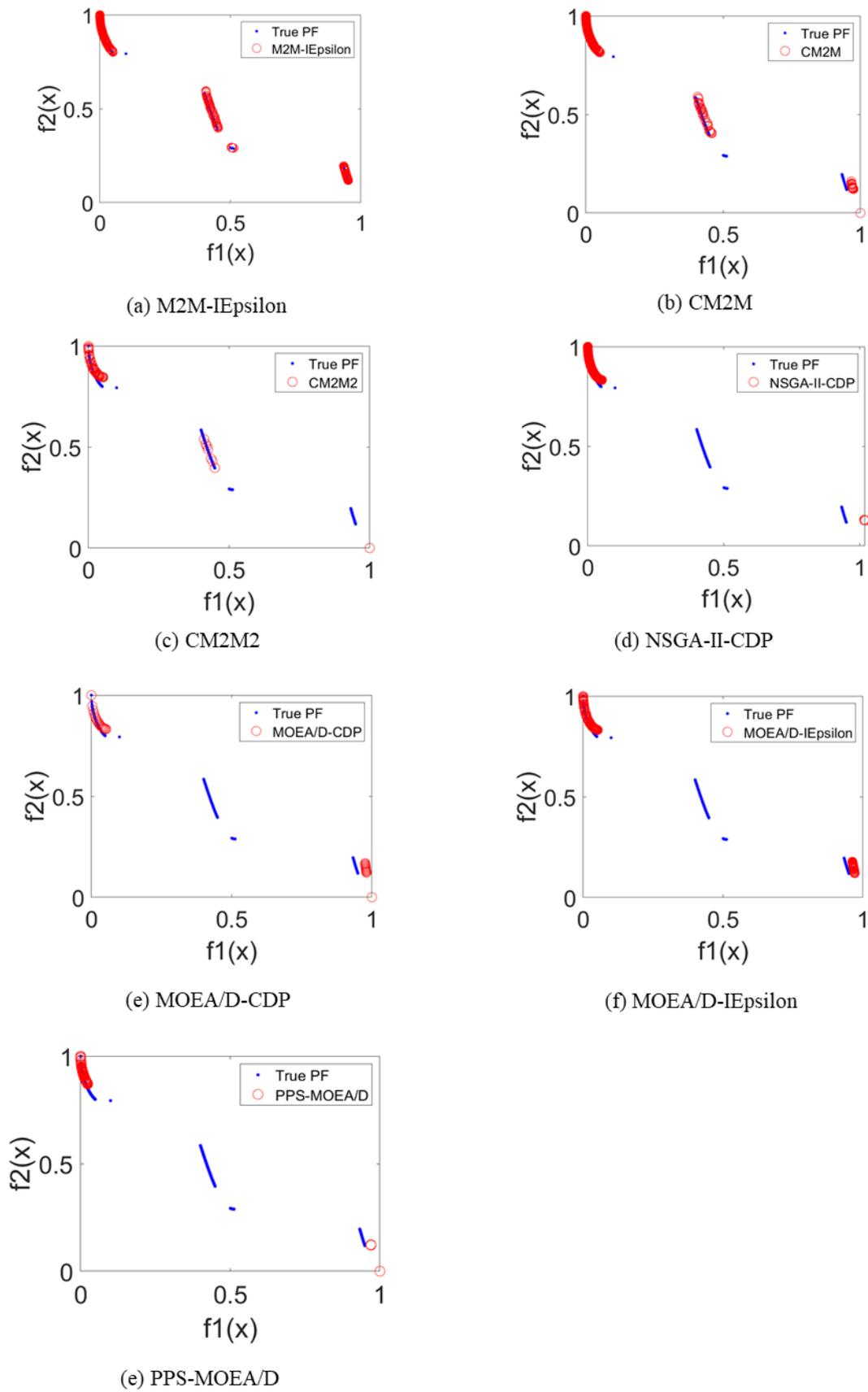


图 4-4 每种算法在 ICD-CMOP4 上获得的非支配解集

表 4-2 显示了 7 种 CMOEAs 在 30 次独立测试中 ICD-CMOP1-7 的 HV 指标的均值和方差，黑色加粗代表该算法在其对应问题上取得了较好的 HV 值。对于 ICD-CMOP1-5，M2M-IEpsilon 在 HV 指标方面明显优于其他 6 个 CMOEAs。而 Wilcoxon 测试结果表明，在 ICD-CMOP6 中，M2M-IEpsilon 显著优于 CM2M、NSGA-II-CDP、MOEA/D-CDP 和 MOEA/D-IEpsilon，而与 CM2M2 和 PPS-MOEA/D 的差异不显著。ICD-CMOP7 中，M2M-IEpsilon 显著优于 CM2M、CM2M2、NSGA-II-CDP、MOEA/D-CDP 和 MOEA/D-IEpsilon，而与 PPS-MOEA/D 无显著差异。从以上观察结果可以看出，M2M-IEpsilon 在大多数 ICD-CMOPs 上的表现优于其他 6 个 CMOEAs。

为了进一步讨论所提出算法 M2M-IEpsilon 在解决 ICD-CMOPs 上的优势，图 4-2、图 4-3 和图 4-4 分别绘制了七种算法在 ICD-CMOP1、ICD-CMOP2 以及 ICD-CMOP4 问题集上获得的 HV 中位值的非支配解集，红色圆圈代表算法获得的解，蓝色点表示真实 PF。在图 4-2(a)、图 4-3(a)以及图 4-4(a)中，可以清楚地发现，所提出的 M2M-IEpsilon 可以找到全部的真实 PFs，而其他 6 个 CMOEAs 只能收敛到部分真实 PFs。有两个可能的原因，一是 ICD-CMOPs 的目标函数是不平衡的，只有 M2M 分解方法才能有效地处理目标函数不平衡的 CMOPs。另一个原因是 ICD-CMOP2 具有多样性难度和收敛性难度约束，这使得 CM2M 和 CM2M2 难以收敛到整个真实 PFs，因为 CM2M 和 CM2M2 没有特定的约束处理机制来处理同时具有多样性和收敛性难度约束的 CMOPs。然而，所提出的 M2M-IEpsilon 具有跨越大的不可行区域并收敛到断开的 PFs 的能力，这使得它在 ICD-CMOPs 上具有最好的性能。因此，提出的 M2M-IEpsilon 在 ICD-CMOPs 上具有更好的性能。

4.4 本章小结

本章为了验证提出的混合 CMOEA—M2M-IEpsilon 在 ICD-CMOPs 测试问题集上的性能，M2M-IEpsilon 和其他 6 个最先进的 CMOEAs，包括 CM2M、CM2M2、NSGA-II-CDP、MOEA/D-CDP、PPS-MOEA/D 和 MOEA/D-IEpsilon 进行了仿真实验研究。IGD 和 HV 指标的结果、Wilcoxon 秩和检验以及上述部分 ICD-CMOPs 问题集上的帕累托可视化优化结果都表明，所提出的 M2M-IEpsilon 在大多数 ICD-CMOPs 测试问题上明显优于其他六种 CMOEA，表明了所提出的 M2M-IEpsilon 能有效处理目标不平衡且约束同时具有多样性难度和收敛性难度的 CMOPs，具有跨越不可行区域，同时提高算法的收敛性和多样性。

第 5 章 M2M-IEpsilon 算法在减速器设计上的应用

多目标优化一直是研究人员广泛贡献的一个重要领域,由于大多数工程科学问题需要同时满足多个相互冲突的目标函数,多目标约束处理也就成为工程优化研究中最活跃的领域之一。通过以上实验可以看出,提出的 M2M-IEpsilon 算法对于求解目标不平衡且约束同时具有多样性困难和收敛性困难的 CMOPs 是有效的。在本章中, M2M-IEpsilon 的性能将进一步在实际约束多目标优化工程问题上进行验证。本章围绕典型的实际工程优化问题展开研究。

5.1 减速器设计

针对现实世界的约束多目标优化问题的建模、优化是数学、计算机科学等许多领域学者研究的主要问题。大多数现实工程问题涉及多个目标且含有多个复杂约束,这种情况使得约束多目标优化成为一项具有挑战性的任务,成为理论界和工程界研究的热点。为了进一步揭示 M2M-IEpsilon 在现实问题中的适用性,选择减速器设计这一广泛使用的工程优化问题进行了测试。

减速器设计问题是由 Rao 提出^[84], 现已广泛用来验证优化算法的性能, 其结构如图 5-1 所示, 其中包括了 7 个连续决策变量和 11 个不等式约束。该问题的目标是同时优化齿轮组的重量和轴的轴向变形。设计变量 x_1 是齿轮面宽度; x_2 是齿模数量; x_3 是小齿轮的齿数; x_4 和 x_5 分别是轴 1 和轴 2 的长度, x_6 和 x_7 分别为轴 1 和轴 2 的直径。不等式约束分别是 $g_1(x)$ 为齿轮弯曲应力的上限; $g_2(x)$ 是齿轮接触应力的上界; $g_3(x)$ 和 $g_4(x)$ 是轴的轴向变形; $g_5(x)$ 和 $g_7(x)$ 是基于空间的尺寸限制; $g_8(x)$ 和 $g_9(x)$ 是根据经验对轴的设计要求; $g_{10}(x)$ 和 $g_{11}(x)$ 是齿轮轴中的应力约束。

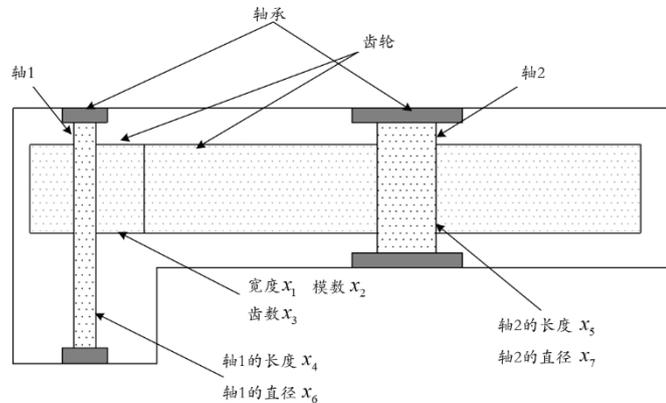


图 5-1 减速器设计问题

减速器设计问题的数学表达式如下所示：

$$\left\{ \begin{array}{l} \text{minimize } f_1(\mathbf{x}) = 0.7854x_1x_2^2\left(\frac{10}{3}x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1(x_6^2 + x_7^2) \\ \quad + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \\ \text{minimize } f_2(\mathbf{x}) = \frac{\sqrt{745\left(\frac{x_4}{x_2x_3}\right)^2 + 1.69 \times 10^7}}{0.1x_6^3} \end{array} \right. \quad (5-1)$$

约束表达式和决策变量范围分别如下公式所示：

$$\left\{ \begin{array}{l} g_1(\mathbf{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \\ g_2(\mathbf{x}) = \frac{1}{x_1x_2^2x_3^2} - \frac{1}{397.5} \leq 0 \\ g_3(\mathbf{x}) = \frac{x_4^3}{x_2x_3x_6^4} - \frac{1}{1.93} \leq 0 \\ g_4(\mathbf{x}) = \frac{x_5^3}{x_2x_3x_7^4} - \frac{1}{1.93} \leq 0 \\ g_5(\mathbf{x}) = x_2x_3 - 40 \leq 0 \\ g_6(\mathbf{x}) = \frac{x_1}{x_2} - 12 \leq 0 \\ g_7(\mathbf{x}) = 5 - \frac{x_1}{x_2} \leq 0 \\ g_8(\mathbf{x}) = 1.9 - x_4 + 1.5x_6 \leq 0 \\ g_9(\mathbf{x}) = 1.9 - x_5 + 1.1x_7 \leq 0 \\ g_{10}(\mathbf{x}) = \frac{\sqrt{\left(745\frac{x_4}{x_2x_3}\right)^2 + 1.69 \times 10^7}}{0.1x_6^3} - 1300 \leq 0 \\ g_{11}(\mathbf{x}) = \frac{\sqrt{\left(745\frac{x_5}{x_2x_3}\right)^2 + 1.575 \times 10^8}}{0.1x_7^3} - 1100 \leq 0 \\ 2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4, x_5 \leq 8.3, \\ 2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5, \end{array} \right. \quad (5-2)$$

5.2 实验结果与分析

同样的，对于减速器设计问题，M2M-IEpsilon 与 6 个最先进的约束多目标进化算法,包括 CM2M, CM2M2, NSGA-II-CDP, MOEA/D-CDP, PPS-MOEA/D 和 MOEA/D-IEpsilon 在该问题上进行了实验测试。对于所有的算法，都进行了归一化处理，种群大小设置为 100, 最大函数评估次数设置为 30000。M2M-IEpsilon 和 MOEA/D-IEpsilon 中 IEpsilon 方法的参数设置 $T_c = 240$, $\alpha_c = 0.75$, $\tau = 0.95$ 。七种算法分别独立运行了 30 次，然后计算 IGD 和 HV 值的大小。值得注意的是，对于所有现实世界的优化问题，真实的 Pareto 前沿是未知的，本文将所得到的七种算法的所有近似解集然后进行非支配排序得到非支配解集，这组非支配解集就形成了真实 Pareto 最优集，作为减速器设计问题的真实 PF。

表 5-2 给出了 M2M-IEpsilon 与其他 6 种算法的 HV 和 IGD 指标的比较结果。黑色加粗代表该算法在其对应问题上取得了较好的 HV 值和 IGD 值。从表中可以看出在 HV 指标上，M2M-IEpsilon 的 HV 值是最高的。在 IGD 指标上，M2M-IEpsilon 获得了最低的 IGD 值。两个指标结果都表明 M2M-IEpsilon 算法在该减速器设计问题上获得了第一，显著优于其他 6 种算法。

表 5-1 减速器设计问题上 HV 和 IGD 的均值 (Mean) 和标准差 (Std) 比较结果

Algorithm	HV		IGD	
	Mean	Std	Mean	Std
M2M-IEpsilon	4.2384E-01	3.6341E-05	4.1359E-03	5.5564E-04
CM2M	4.2380E-01	4.1669E-05	4.8435E-03	8.1076E-04
CM2M2	4.2250E-01	6.7143E-04	6.2048E-02	3.0149E-02
NSGA-II-CDP	4.1013E-01	1.6709E-03	9.7069E-02	1.7559E-02
MOEA/D-CDP	4.2326E-01	4.9215E-05	6.6092E-02	2.8346E-03
MOEA/D- IEpsilon	4.2342E-01	1.9452E-04	3.6184E-02	1.8989E-02
PPS-MOEA/D	4.2356E-01	1.5399E-04	2.4971E-02	1.6936E-02

为进一步展示 M2M-IEpsilon 相较于其他 6 种算法的优势，图 5-1 描绘了 7 种算法 30 次独立运行优化后得到的 HV 中位值的非支配解集。红色圆圈代表算法获得的解，蓝色点表示 7 种算法获得的解集进行联合之后进行非支配排序得到的真实 PF。可以清楚地观察到，所提出的 M2M-IEpsilon 可以找到绝大部分真实 PF，而其他 6 个

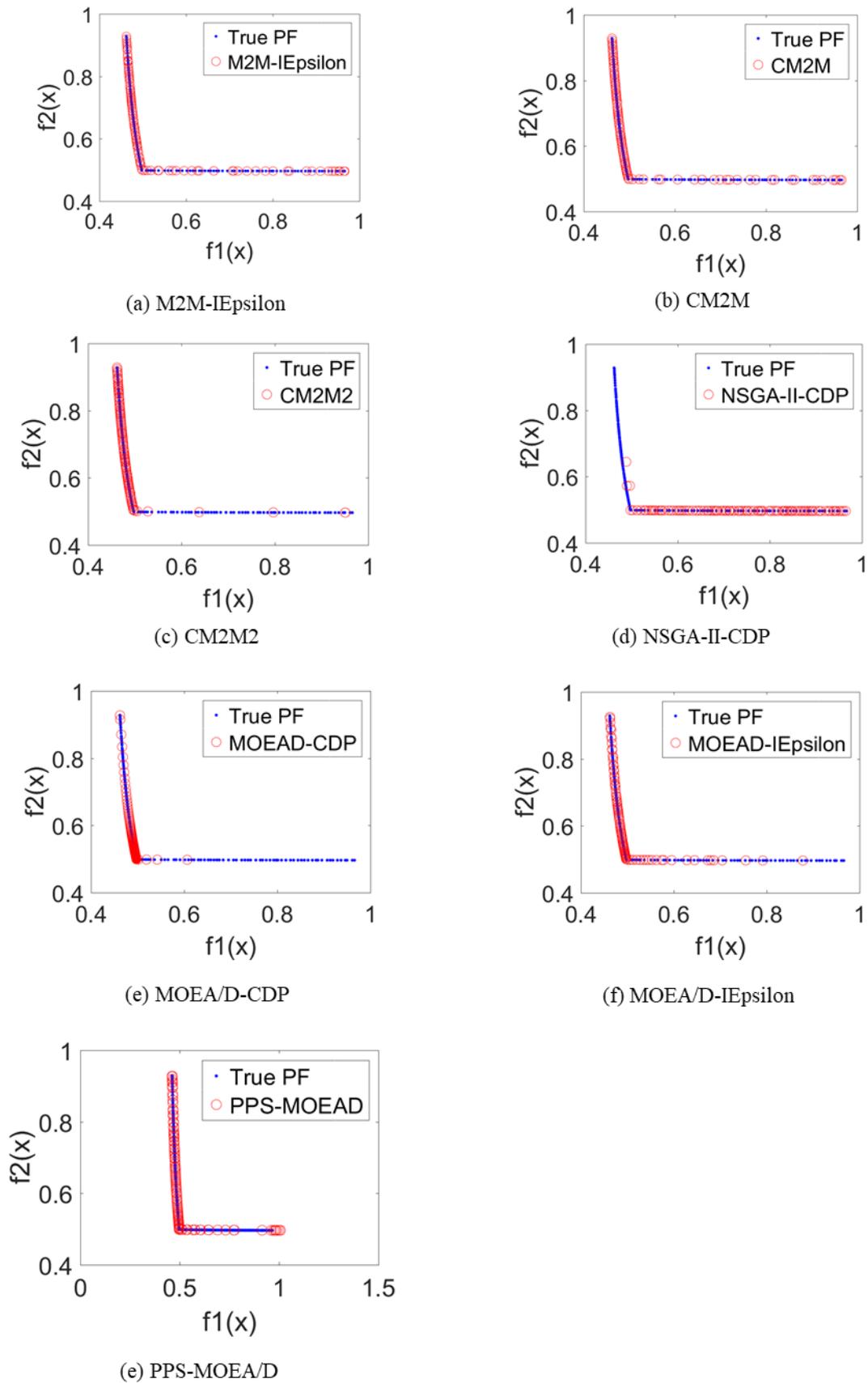


图 5-2 每种算法在减速器设计问题上获得的非支配解集

CMOEAs 只能收敛到部分真实 PF，而从每个算法获得的解在整个真实 PF 的分布性情况可以看出 M2M-IEpsilon 获得的解分布更均匀。

图 5-3 展示了七种算法在减速器设计优化问题的 IGD 指标箱线图，‘1’、‘2’、‘3’、‘4’、‘5’、‘6’、‘7’ 分别代表 M2M-IEpsilon, CM2M, CM2M, NSGA-II-CDP, MOEA/D-CDP, MOEA/D-IEpsilon 以及 PPS-MOEA/D 七种算法。从箱线图也可以观察到，M2M-IEpsilon 在减速器设计优化问题上的 IGD 指标明显优于其他算法。

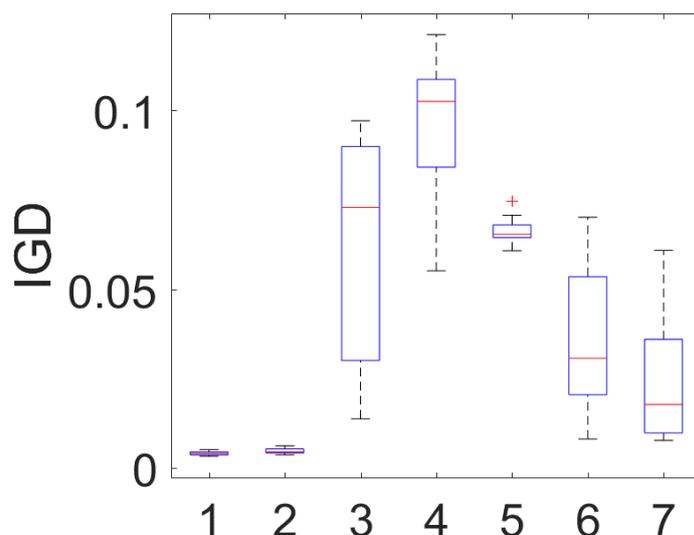


图 5-3 七种算法在减速器设计问题的 IGD 指标箱线图

基于以上分析，M2M-IEpsilon 在减速器设计问题上的性能明显要优于其他 6 中比较算法，而对于其他 CMOEAs 来说寻找沿着 PF 均匀分布的非支配解是非常具有挑战的。可以认为 M2M-IEpsilon 能够有效解决实际工程设计中出现的约束多目标优化问题，是一种非常具有潜力和竞争力的能够应用于实际工程优化的约束多目标进化算法。

5.3 本章小结

本章通过在减速器设计优化问题上的仿真实验，将所提出的算法 M2M-IEpsilon 与目前最先进的 6 种约束多目标进化算法进行了比较，结果表明了所提出算法在该问题上的 IGD 和 HV 方面具有更好或者相当的性能。验证了所提出的 M2M-IEpsilon 在求解现实工程实例的有效性，为该方法进一步扩展解决现实世界的优化问题提供了一定研究基础。

第 6 章 总结

6.1 研究总结

多目标优化问题一直是学界关注的重点,但现实世界的多目标优化不仅包含了多个相互冲突的约束而且往往包含了一个或者多个约束条件,从而导致求解约束多目标优化问题的难度进一步加深和研究学者对这一领域关注度的提升。由于约束条件的存在,势必导致对目标空间搜索难度的加深,可行区域和不可行区域之间搜索的权衡。算法需在可行性、多样性和收敛性之间保持有效平衡。这些问题的存在都给约束多目标优化的有效求解带来了极大的挑战。所以本文就这些问题提出了一种混合 CMOEA,即 M2M-IEpsilon,它将改进的 Epsilon 约束处理方法和 M2M 分解方法相结合,用于求解约束同时具有收敛性难度和多样性难度的不平衡 CMOPs。另一方面针对目标不平衡的约束多目标测试问题集的设计较少,通过使用 DAS-CMOPs 这类可扩展的测试集结合不平衡目标函数,提出了一组基准测试问题,即 ICD-CMOPs,有效评估 M2M-IEpsilon 的性能。

由于 M2M-IEpsilon 采用了 M2M 分解方法,该方法对于求解目标函数不平衡的 CMOPs 非常有效,能帮助算法在目标不平衡 CMOPs 上维持更好的多样性。M2M-IEpsilon 中嵌入的 IEpsilon 约束处理机制可以帮助 M2M-IEpsilon 的工作种群穿越不可行区域,同时提高收敛性能。为了验证这一算法的性能,在 ICD-CMOPs 上测试了 M2M-IEpsilon 和其他 6 个最先进的 CMOEAs,包括 CM2M、CM2M2、NSGA-II-CDP、MOEA/D-CDP、PPS-MOEA/D 和 MOEA/D-IEpsilon。综合实验结果表明,所提出的 M2M-IEpsilon 在大多数测试问题上明显优于其他六种 CMOEA,表明了所提出的 M2M-IEpsilon 的优越性。此外,为进一步验证 M2M-IEpsilon 的有效性,在现实世界的工程应用问题上一减压器设计上的实验结果也表明, M2M-IEpsilon 有助于提高算法在目标空间的探索以及维持算法多样性和收敛性方面的平衡。以上实验结果都表明 M2M-IEpsilon 算法比其他 6 种最先进算法都要明显优越。

6.2 研究展望

本文针对现有约束多目标进化算法在处理一些复杂约束条件上的不足,针对性的提出了一种混合约束多目标进化算法用以解决一类约束多目标测试问题集。最优化理

论提及的没有免费午餐定理所描述，只有在指定具体问题时候，才能说某一优化方法优于另一种算法。所以针对不同类型的约束难度和目标函数特性，分析不同的约束处理方法和目标函数处理方法对于约束多目标进化算法的研究都是非常必要的。另一方面随着机器学习技术的迅猛发展，这对于进化算法求解约束多目标优化也是具有积极促进作用，现有的利用进化算法进行神经网络的架构搜索（NAS）和超参数优化已经成为一个研究热点。但由于进化算法通常所需的评估次数较多，目标函数的每一次评价都需要耗费昂贵的时间和计算，减少计算的负担也已成为进化算法求解多目标优化的一个重要而困难的问题。实际的工程应用设计中，存在许多无法使用具体表达式描述的黑箱优化问题，这类问题也通常具有计算代价昂贵和运算速度慢等昂贵评价特征。所以未来的工作可以尝试将机器学习技术与 M2M-IEpsilon 相结合，利用机器学习的强大运算能力、数据挖掘能力，以解决具有昂贵的目标和约束函数的 CMOPs。

参考文献

- [1] Qu B Y, Zhu Y S, Jiao Y C, et al. A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems[J]. *Swarm and Evolutionary Computation*, 2018, 38: 1-11.
- [2] 刘锐. 智能优化算法研究及其在金融领域的应用[D]. 吉林大学, 2020.
- [3] 陈婷. 智能算法在含分布式电源配电网故障定位中的应用综述[J]. *电气技术*, 2020, 21(09): 1-6.
- [4] Li Y L, Zhan Z H, Gong Y J, et al. Differential evolution with an evolution path: A DEEP evolutionary algorithm[J]. *IEEE Transactions on Cybernetics*, 2014, 45(9): 1798-1810.
- [5] Sadollah A, Bahreininejad A, Eskandar H, et al. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems[J]. *Applied Soft Computing*, 2013, 13(5): 2592-2612.
- [6] 刘汉, 康国钦, 李凯. 基于差分进化算法设计的 MIMO 超宽带天线[J]. *华南理工大学学报(自然科学版)*, 2020, 48(03): 24-31.
- [7] 郑金华. 多目标进化算法及其应用[M]. 科学出版社, 2007.
- [8] Dorigo M, Caro G D, Gambardella L M. Ant algorithms for discrete optimization[J]. *Artificial Life*, 1999, 5(2): 137-172.
- [9] Kennedy J, Eberhart R. Particle swarm optimization[C]. *Proceedings of ICNN'95-International Conference on Neural Networks*. IEEE, 1995, 4: 1942-1948.
- [10] Li X. An optimizing method based on autonomous animats: fish-swarm algorithm[J]. *Systems Engineering-Theory & Practice*, 2002, 22(11): 32-38.
- [11] De Castro L N, Von Zuben F J. Artificial immune systems: Part I—basic theory and applications[J]. *Universidade Estadual de Campinas, Dezembro de, Tech. Rep*, 1999, 210(1).
- [12] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing[J]. *Science*, 1983, 220(4598): 671-680.
- [13] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[14] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[15] Rechenberg I. Cybernetic solution path of an experimental problem, library translation[J]. In Royal aircraft establishment, Farnborough. royal aircraft establishment library translation, 1965.

[16] Koza J R. Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems[M]. Stanford, CA: Stanford University, Department of Computer Science, 1990.

[17] Fogel L J, Owens A J, Walsh M J, *Artificial Intelligence Through Simulated Evolution*[J], John Wiley & Son, New York, NY, 1966.

[18] Such F P, Madhavan V, Conti E, et al. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning[J]. arXiv preprint arXiv:1712.06567, 2017.

[19] Mousavi-Avval S H, Rafiee S, Sharifi M, et al. Application of multi-objective genetic algorithms for optimization of energy, economics and environmental life cycle assessment in oilseed production[J]. *Journal of Cleaner Production*, 2017, 140: 804-815.

[20] Liang H, Zou J, Zuo K, et al. An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control system[J]. *Mechanical Systems and Signal Processing*, 2020, 142: 106708.

[21] 张建宇,段建龙.基于改进差分进化算法的机器人动力学参数辨识方法研究[J].北京联合大学学报,2020,34(01):48-56.

[22] Salimans T, Ho J, Chen X, et al. Evolution strategies as a scalable alternative to reinforcement learning[J]. arXiv preprint arXiv:1703.03864, 2017.

[23] Knowles J, Corne D. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation[C]. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. IEEE, 1999, 1: 98-105.

[24] Mezura-Montes E, Coello C A C. A simple multimembered evolution strategy to solve constrained optimization problems[J]. *IEEE Transactions on Evolutionary computation*, 2005, 9(1): 1-17.

[25] Hansen N. The CMA evolution strategy: a comparing review[J]. *Towards a new evolutionary computation*, 2006: 75-102.

[26] Chen H, Cheng R, Wen J, et al. Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations[J]. *Information Sciences*, 2020, 509: 457-469.

- [27] Poli R, Langdon W B, McPhee N F, et al. A field guide to genetic programming[M]. Lulu. com, 2008.
- [28] Vikhar P A. Evolutionary algorithms: A critical review and its future prospects[C]. 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC). IEEE, 2016: 261-265.
- [29] Ishibuchi H, Murata T. A multi-objective genetic local search algorithm and its application to flowshop scheduling[J]. IEEE transactions on systems, man, and cybernetics, part C (applications and reviews), 1998, 28(3): 392-403.
- [30] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on evolutionary computation, 2007, 11(6): 712-731.
- [31] Cheng R, Jin Y, Olhofer M, et al. A reference vector guided evolutionary algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5): 773-791.
- [32] Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II[J]. IEEE transactions on evolutionary computation, 2008, 13(2): 284-302.
- [33] Li K, Deb K, Zhang Q, et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition[J]. IEEE Transactions on Evolutionary Computation, 2014, 19(5): 694-716.
- [34] Ke L, Zhang Q, Battiti R. MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony[J]. IEEE Transactions on Cybernetics, 2013, 43(6): 1845-1859.
- [35] Y. Yuan, H. Xu, B. Wang, B. Zhang and X. Yao, "Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers", IEEE Transactions on Evolutionary Computation, vol. 20, no. 2, pp. 180-198, 2016.
- [36] Fan Z, Fang Y, Li W, et al. MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems[J]. Applied Soft Computing, 2019, 74: 621-633.
- [37] Zhang J, Zhu X, Chen T, et al. Improved MOEA/D approach to many-objective day-ahead scheduling with consideration of adjustable outputs of renewable units and load reduction in active distribution networks[J]. Energy, 2020, 210: 118524.
- [38] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

- [39] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[J]. TIK-report, 2001, 103.
- [40] Corne D W, Jerram N R, Knowles J D, et al. PESA-II: Region-based selection in evolutionary multiobjective optimization[C]. Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. 2001: 283-290.
- [41] Deb K, Sundar J. Reference point based multi-objective optimization using evolutionary algorithms[C]. Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. 2006: 635-642.
- [42] Bechikh S, Said L B, Ghédira K. Searching for knee regions of the Pareto front using mobile reference points[J]. Soft Computing, 2011, 15(9): 1807-1823.
- [43] Kollat J B, Reed P M. The value of online adaptive search: a performance comparison of NSGAI, ϵ -NSGAI and ϵ MOEA[C]. International Conference on Evolutionary Multi-Criterion Optimization. Springer, Berlin, Heidelberg, 2005: 386-398.
- [44] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints[J]. IEEE transactions on evolutionary computation, 2013, 18(4): 577-601.
- [45] Yang S, Li M, Liu X, et al. A grid-based evolutionary algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2013, 17(5): 721-736.
- [46] Zhang X, Tian Y, Jin Y. A knee point-driven evolutionary algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2014, 19(6): 761-776.
- [47] Van Veldhuizen D A, Lamont G B. Multiobjective evolutionary algorithm research: A history and analysis[R]. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.
- [48] Zitzler E, Thiele L. Multiobjective optimization using evolutionary algorithms—a comparative case study[C]. International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 1998: 292-301.
- [49] Czyżżak P, Jaskiewicz A. Pareto simulated annealing—a metaheuristic technique for multiple - objective combinatorial optimization[J]. Journal of Multi - Criteria Decision Analysis, 1998, 7(1): 34-47.
- [50] Brockhoff D, Wagner T, Trautmann H. On the properties of the R2 indicator[C]. Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation. 2012:

465-472.

[51] Schott J R. Fault tolerant design using single and multicriteria genetic algorithm optimization[D]. Massachusetts Institute of Technology, 1995.

[52] Ishibuchi H, Masuda H, Tanigaki Y, et al. Modified distance calculation in generational distance and inverted generational distance[C]. International Conference on Evolutionary Multi-criterion Optimization. Springer, Cham, 2015: 110-125.

[53] Beume N, Naujoks B, Emmerich M. SMS-EMOA: Multiobjective selection based on dominated hypervolume[J]. European Journal of Operational Research, 2007, 181(3): 1653-1669.

[54] Menchaca-Mendez A, Coello C A C. GDE-MOEA: a new moea based on the generational distance indicator and ϵ -dominance[C]. 2015 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2015: 947-955.

[55] Li F, Cheng R, Liu J, et al. A two-stage R2 indicator based evolutionary algorithm for many-objective optimization[J]. Applied Soft Computing, 2018, 67: 245-260.

[56] Bader J, Zitzler E. HypE: An algorithm for fast hypervolume-based many-objective optimization[J]. Evolutionary Computation, 2011, 19(1): 45-76.

[57] 李笠,李广鹏,常亮,古天龙.约束进化算法及其应用研究综述[J].计算机科学,2021,48(04):1-13.

[58] Hoffmeister F, Sprave J. Problem-independent handling of constraints by use of metric penalty functions[J]. 1996.

[59] Homaifar A, Qi C X, Lai S H. Constrained optimization via genetic algorithms[J]. Simulation, 1994, 62(4): 242-253.

[60] Joines J A, Houck C R. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's[C]. Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence. IEEE, 1994: 579-584.

[61] Ben Hadj-Alouane A, Bean J C. A genetic algorithm for the multiple-choice integer program[J]. Operations Research, 1997, 45(1): 92-101.

[62] Deb K. An efficient constraint handling method for genetic algorithms[J]. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2-4): 311-338.

[63] Takahama T, Sakai S. Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites[C]. 2006 IEEE international conference on evolutionary computation. IEEE, 2006: 1-8.

[64] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary

optimization[J]. IEEE Transactions on Evolutionary Computation, 2000, 4(3): 284-294.

[65] Wang Y, Cai Z, Zhou Y, et al. An adaptive tradeoff model for constrained evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 80-92.

[66] Mallipeddi R, Suganthan P N. Ensemble of constraint handling techniques[J]. IEEE Transactions on Evolutionary Computation, 2010, 14(4): 561-579.

[67] Hsieh M N, Chiang T C, Fu L C. A hybrid constraint handling mechanism with differential evolution for constrained multiobjective optimization[C]. 2011 IEEE Congress of Evolutionary Computation (CEC). IEEE, 2011: 1785-1792.

[68] Mani A, Patvardhan C. A novel hybrid constraint handling technique for evolutionary optimization[C]. 2009 IEEE Congress on Evolutionary Computation. IEEE, 2009: 2577-2583.

[69] Deb K, Thiele L, Laumanns M, et al. Scalable multi-objective optimization test problems[C]. Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600). IEEE, 2002, 1: 825-830.

[70] Liu H L, Gu F, Zhang Q. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems[J]. IEEE Transactions on Evolutionary Computation, 2013, 18(3): 450-455.

[71] Fan Z, Li W, Cai X, et al. Difficulty adjustable and scalable constrained multiobjective test problem toolkit[J]. Evolutionary computation, 2020, 28(3): 339-378.

[72] Fan Z, Li W, Cai X, et al. An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions[J]. Soft Computing, 2019, 23(23): 12491-12510.

[73] Deb K, Thiele L, Laumanns M, et al. Scalable multi-objective optimization test problems[C]. Congress on Evolutionary Computation. IEEE, 2002.

[74] Huband S, Hingston P, Barone L, et al. A review of multiobjective test problems and a scalable test problem toolkit[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(5): 477-506.

[75] Cheng R, Li M, Tian Y, et al. A benchmark test suite for evolutionary many-objective optimization[J]. Complex & Intelligent Systems, 2017, 3(1): 67-81.

[76] Deb K, Pratap A, Meyarivan T. Constrained test problems for multi-objective evolutionary optimization[C]. International conference on evolutionary multi-criterion optimization. Springer, Berlin, Heidelberg, 2001: 284-298.

[77] Zhang Q, Zhou A, Zhao S, et al. Multiobjective optimization test instances for

the CEC 2009 special session and competition[J]. 2008.

[78] Zhou Y, Xiang Y, He X. Constrained Multi-objective Optimization: Test Problem Construction and Performance Evaluations[J]. IEEE Transactions on Evolutionary Computation, 2020.

[79] Liu H L, Chen L, Deb K, et al. Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms[J]. IEEE Transactions on Evolutionary Computation, 2016, 21(3): 408-425.

[80] Liu H L, Peng C, Gu F, et al. A constrained multi-objective evolutionary algorithm based on boundary search and archive[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2016, 30(01): 1659002.

[81] Peng C, Liu H L, Gu F. An evolutionary algorithm with directed weights for constrained multi-objective optimization[J]. Applied Soft Computing, 2017, 60: 613-622.

[82] Jan M A, Khanum R A. A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D[J]. Applied Soft Computing, 2013, 13(1): 128-148.

[83] Fan Z, Li W, Cai X, et al. Push and pull search for solving constrained multi-objective optimization problems[J]. Swarm and Evolutionary Computation, 2019, 44: 665-679.

[84] Rao S S. Engineering optimization: theory and practice[M]. John Wiley & Sons, 2019.

攻读学位期间主要研究成果

一、发表的学术论文

(一) 已见刊或录用论文

- [1] Fan Z, **Yang Z**, Tang Y, Li W, Xu B, Wang Z, Sun F, Long Z, and Zhu G. (2021) An Improved Epsilon Method with M2M for Solving Imbalanced CMOPs with Simultaneous Convergence-Hard and Diversity-Hard Constraints. Evolutionary Multi-Criterion Optimization: 11th International Conference, EMO 2021, Shenzhen, China, March 28–31, 2021, Proceedings 11. Springer International Publishing, 2021: 248-256.
- [2] Z. Fan, Z. Wang, W. Li, Y. Yuan, Y. You, **Z. Yang**, F. Sun and J. Ruan. Push and pull search embedded in an M2M framework for solving constrained multi-objective optimization problems[J]. Swarm and Evolutionary Computation, 2020, 54: 100651.
- [3] Fan Z, Ruan J, Li W, You Y, Cai X, Xu Z, **Yang Z**, Sun F, Wang Z, Yuan Y. A Learning Guided Parameter Setting for Constrained Multi-Objective Optimization. 2019 1st International Conference on Industrial Artificial Intelligence (IAI). IEEE, 2019 1–6.

致谢

缱绻年华，时光如水，在透漏着一丝热气的盛夏和凤凰花渐次盛开的夏季迎来了毕业的铃声。此刻，落笔有千言，只化作一句感恩。感恩这段岁月中帮助我的老师、同学、家人朋友们，谢谢你们的支持和帮助，使我成长、收获许多。

学问、见识、成长这是在汕大收获的又一笔人生财富。学业之收获，在于师者授业解惑也，感谢我的两位导师唐雅娟老师和范衡老师。衷心感谢唐雅娟老师三年来的耐心细致的指导和亲切的关怀，让我得以迷茫时还能不断前行。感谢范衡老师将我纳入到课题组学习，让我有机会接触到这一研究方向，增长了自己的见识和视野。两位老师在学术科研上一直都耐心指导和帮助，学术上都是学识渊博、见识练达、敢于创新、视野开阔的学者。科研上始终治学严谨，循循善诱、精益求精，敦促我克服困难、潜移默化的影响我，耐心的帮助我修改论文并给出有利的改进建议。

特别感谢文姬师兄，帮助我确定了科研的方向，培养科研入门的技能，耐心的帮助我修改、分析论文，学习上对我的督促等，亦师亦友，也在生活、为人处世方面给了我一定引导。谢谢朱贵杰师兄、王诏君师兄、袁宇彤师兄给予的在学术调研、论文撰写等的指导，在生活之余的关心和帮助。感谢孙福赞、金鹏程、王聪、陈婧瑶、邱立靖、胡海敏、胡星晨、熊宇、罗晨林以及其他同学一起学习，一起玩乐的时光，这三年的生活才充实又有趣。谢谢龙周彬、阮洁、张章苗、邱龙凤、樊思萍、王柳等师弟师妹们不仅一起在学习上相互指导，也在工作和生活中相互照应。还有我亲爱的室友喻喻，谢谢你一直不离不弃的陪伴、照顾和关心，让我逐渐适应汕头熟悉这座城市，远在他乡也能感受到一份温暖。谢谢我的两位闺蜜毛河和黄亚萍，即使远在其他城市，即使只是匆匆一小会的相聚也能飞奔而来，在我需要你们时总能让我感受到你们的关心和爱啊，陪伴最是长情，情谊最是珍重，要继续一起 lucky。

最后，真诚的谢谢我的父母亲和姐姐。二十余载求学生涯，是你们的辛劳和汗水让我一直得以接受教育，用最无私的爱默默关心我，让我能毫无顾虑的完成我的学业，让我有机会走出大山，看到外面精彩的世界。还有可爱的景怡小侄女，在我烦恼时总能皮一下让我顿时开心不少。谢谢姨妈一家，你们的支持和帮助我才以顺利完成学业。我最爱的家人们，你们的鼓励和支持，一直都是我前进的最大动力。

作者：杨知

2021年5月9日

附录

附表 I 测试问题 ICD-CMOP1-7 目标函数及其约束条件

编号	目标函数	约束条件
1	$\begin{cases} \min & f_1(\mathbf{x}) = (1 + g(\mathbf{x}))x_1 \\ \min & f_2(\mathbf{x}) = (1 + g(\mathbf{x}))(1 - \sqrt{x_1}) \\ \text{where} & g(\mathbf{x}) = 2 \sin(\pi x_1) \sum_{i=2}^n (-0.9t_i^2 + t_i^{0.6}) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0, 1]^n \end{cases}$	$\begin{cases} c_1(\mathbf{x}) = \sin(a\pi x_1) - b \geq 0 \\ c_k(\mathbf{x}) = ((f_1 - p_k)\cos\theta_k - (f_2 - q_k)\sin\theta_k)^2 / a_k^2 \\ \quad + ((f_1 - p_k)\sin\theta_k - (f_2 - q_k)\cos\theta_k)^2 / b_k^2 \geq r \\ a = 20, b = 0 \\ p_k = [0, 1, 0, 1, 2, 0, 1, 2, 3] \\ q_k = [1.5, 0.5, 2.5, 1.5, 0.5, 3.5, 2.5, 1.5, 0.5] \\ a_k^2 = 0.4, b_k^2 = 1.6, \theta_k = -0.25\pi \\ c = 20, n = 30, x_i \in [0, 1], k = 1, \dots, 9 \end{cases}$
2	$\begin{cases} \min & f_1(\mathbf{x}) = (1 + g(\mathbf{x}))x_1 \\ \min & f_2(\mathbf{x}) = (1 + g(\mathbf{x}))(1 - x_1^2) \\ \text{where} & g(\mathbf{x}) = 10 \sin(\pi x_1) \sum_{i=2}^n \left(\frac{ t_i }{1 + e^{5 t_i }} \right) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0, 1]^n \end{cases}$	They are the same as those of ICD-CMOP1
3	$\begin{cases} \min & f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos\left(\frac{\pi x_1}{2}\right) \\ \min & f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \sin\left(\frac{\pi x_1}{2}\right) \\ \text{where} & g(\mathbf{x}) = 10 \sin\left(\frac{\pi x_1}{2}\right) \sum_{i=2}^n \left(\frac{ t_i }{1 + e^{5 t_i }} \right) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0, 1]^n \end{cases}$	They are the same as those of ICD-CMOP1

4	$\begin{cases} \min & f_1(\mathbf{x}) = (1 + g(\mathbf{x}))x_1 \\ \min & f_2(\mathbf{x}) = (1 + g(\mathbf{x}))(1 - x_1^{0.5} \cos^2(2\pi x_1)) \\ \text{where} & g(\mathbf{x}) = 1 + 10 \sin(\pi x_1) \sum_{i=2}^n \left(\frac{ t_i }{1 + e^{5 t_i }} \right) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0, 1]^n \end{cases}$	They are the same as those of ICD-CMOP1
5	$\begin{cases} \min & f_1(\mathbf{x}) = (1 + g(\mathbf{x}))x_1 \\ \min & f_2(\mathbf{x}) = (1 + g(\mathbf{x}))(1 - \sqrt{x_1}) \\ \text{where} & g(\mathbf{x}) = 2 \cos(\pi x_1) \sum_{i=2}^n (-0.9t_i^2 + t_i^{0.6}) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0, 1]^n \end{cases}$	They are the same as those of ICD-CMOP1
6	$\begin{cases} \min & f_1(\mathbf{x}) = (1 + g(\mathbf{x}))x_1x_2 \\ \min & f_2(\mathbf{x}) = (1 + g(\mathbf{x}))x_1(1 - x_2) \\ \min & f_3(\mathbf{x}) = (1 + g(\mathbf{x}))(1 - x_1) \\ \text{where} & g(\mathbf{x}) = 2 \sin(\pi x_1) \sum_{i=3}^n (-0.9t_i^2 + t_i^{0.6}) \\ & t_i = x_i - x_1x_2 \\ & n = 10, \mathbf{x} \in [0, 1]^n \end{cases}$	$\begin{cases} c_1(\mathbf{x}) = \sin(a\pi x_1) - b \geq 0 \\ c_2(\mathbf{x}) = \cos(a\pi x_2) - b \geq 0 \\ c_k(\mathbf{x}) = ((f_1 - p_k)\cos\theta_k - (f_2 - q_k)\sin\theta_k)^2 / a_k^2 \\ \quad + ((f_1 - p_k)\sin\theta_k - (f_2 - q_k)\cos\theta_k)^2 / b_k^2 \geq r \\ a = 20, b = 0 \\ p_k = [0, 1, 0, 1, 2, 0, 1, 2, 3] \\ q_k = [1.5, 0.5, 2.5, 1.5, 0.5, 3.5, 2.5, 1.5, 0.5] \\ a_k^2 = 0.4, b_k^2 = 1.6, \theta_k = -0.25\pi \\ c = 20, n = 30, x_i \in [0, 1], k = 1, \dots, 9 \end{cases}$
7	$\begin{cases} \min & f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos\left(\frac{x_1\pi}{2}\right) \cos\left(\frac{x_2\pi}{2}\right) \\ \min & f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \cos\left(\frac{x_1\pi}{2}\right) \sin\left(\frac{x_2\pi}{2}\right) \\ \min & f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin\left(\frac{x_2\pi}{2}\right) \\ \text{where} & g(\mathbf{x}) = 2 \sin(\pi x_1) \sum_{i=3}^n (-0.9t_i^2 + t_i^{0.6}) \\ & t_i = x_i - x_1x_2 \\ & n = 10, \mathbf{x} \in [0, 1]^n \end{cases}$	They are the same as those of ICD-CMOP6