

博士学位论文

求解多模态优化问题的多目标进化算法研究

**Research on Multi-objective Evolutionary Algorithm for
Solving Multi-modal Optimization Problems**

学科专业 计算机应用技术

研究方向 计算智能

作者姓名 宋 武

指导教师 蔡自兴 教授

中南大学

二〇一六年五月

中图分类号 TP391
UDC 620

学校代码 10533
学位类别 学术学位

博士学位论文

求解多模态优化问题的多目标进化算法研究

Research on Multi-objective Evolutionary Algorithm for Solving Multi-modal Optimization Problems

作 者 姓 名：宋武
学 科 专 业：计算机应用技术
研 究 方 向：计算智能
学院（系、所）：信息科学与工程学院
指 导 教 师：蔡自兴 教授

论文答辩日期_____ 答辩委员会主席_____

中南大学
二〇一六年五月

学位论文原创性声明

本人郑重声明，所呈交的学位论文是本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，除了论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，也不包含为获得中南大学或其他教育机构的学位或证书而使用过的材料。与我共同工作的同志对本研究所作的贡献均已在论文中作了明确的说明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

作者签名：_____ 日期：_____年____月____日

学位论文授权使用授权书

本学位论文作者和指导教师完全了解中南大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交学位论文的复印件和电子版；本人允许本学位论文被查阅和借阅；学校可以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，可以采用复印、缩印或其它手段保存和汇编本学位论文。

保密论文待解密后适应本声明。

作者签名：_____

导师签名_____

日期：_____年____月____日

日期：_____年____月____日

求解多模态优化问题的多目标进化算法研究

摘要：长期以来，多模态优化问题的进化求解技术一直是进化算法研究的热点。一方面，多模态优化问题与普通的单目标优化问题不同，需要求取一组而不仅是一个解。另一方面，进化算法采用全序的适应值比较形式，适合求解单目标优化问题，但是对于多模态优化问题，需要改变其进化框架以达到求解多模态优化问题的多个最优值的目的。使用带有小生境策略的进化算法求解多模态优化问题是现在的主流，取得了较多的成果，同时也带来一些参数的设置问题。多目标进化算法由于其目的在于求解一组多个目标相互冲突的 Pareto 最优解集，在求解多目标优化问题时取得了相当多的成果。多目标优化问题与多模态优化问题的求解是相似的，因此很自然的可以利用多目标进化算法求解多模态优化问题。现今国际上使用多目标进化算法求解多模态优化问题表现出了相当的观注。

本文围绕使用多目标进化算法求解多模态优化问题这一主题，在充分研究了各种性质的多模态函数性状后，首先对特殊的带有先验信息的多解问题——多解的非线性方程系统(包含多个解)提出一种多目标转换模型，并使用多目标进化算法求解。然后将其推广到一般的多模态优化问题，并设计了相关的多目标进化算法。最后针对已提出的求解多模态优化问题多目标进化算法，提出了一种基于树结构的多目标进化算法，提高了进化效率。

本文的主要研究工作和创新性成果如下：

(1) 提出一种求解多解的非线性方程系统的双目标转换方法。在此种方法中将一个多解的非线性方程系统的解映射到一个形状已知的双目标优化问题的前沿面上，并证明多解的非线性方程系统所有的解都在此双目标优化问题的前沿面上；在此基础上，可以直接使用多目标进化算法进行求解，避免在单目标框架下需要小生境半径的设置等问题，提高了其求解效率。由于多目标优化算法的结果是用有限的种群近似无限的 Pareto 最优解集，因而此方法很自然地能够对于无限多个解的非线性方程系统进行良好近似。

(2) 提出一种求解多模态优化问题的多目标转换框架。指出使用双目标转换方法求解一般的多模态优化问题时，会存在 2 个问题，即维数的遮挡问题和峰的吸引问题；对此提出一种将多模态优化问题转换为 D 个双目标优化问题的方法，并提出一种新的支配方式，用于求解多模态优化问题。在 CEC2013 的测试集上的实验表明，与国际上经典的多模态优化进化求解算法相比，此多目标优化进化方法对于多模态优化问题的求解精

度高，能够有效的定位多模态优化问题的多个最优值。此种方法在作者已知多目标转换的方法中，是唯一保证所要优化问题的所有最优值互不相同的定位在多目标空间的前沿面上的方法。

(3) 提出一种分维非支配排序树的数据结构，用于多目标优化框架的非支配排序，并将其嵌入多目标进化算法中求解多模态优化问题。此种方法具有较快的运行速度和较高的运行效率。并从理论上论证了该进化算法的时间复杂度为 $N \lg N$ 。

关键词：多模态优化问题，非线性方程系统，多目标优化问题，最优值

分类号：TP391

Research on Multi-objective Evolutionary Algorithm for Solving Multi-modal Optimization Problems

Abstract: Evolutionary algorithm for multi-modal optimization problem has been a hot topic in the research of evolutionary algorithms. On the one hand, multi-modal optimization problem is different from ordinary single objective optimization problem, and it is required to obtain not one solution but a set of solutions. On the other hand, evolutionary algorithm using fitness evaluation which is suitable for solving single objective optimization problem, but for multimodal optimization problems it need to change the evolutionary framework to solve multi-modal optimization problems. Using evolutionary algorithm with niche strategy to solve multimodal optimization problem is the main stream in this field, and has achieved many results, but parameter setting such as niche radius, is also another problem for users to define. Multi-objective evolutionary algorithms, which aims at obtain a set of Pareto optimal solution sets of multiple objectives conflict with each other, lots of achievements have been obtained in the multi-objective optimization problem. For it is similar to the solution properties of multi modal optimization problems, it is very naturally to use multi-objective evolutionary algorithm to solve multimodal optimization problems.

This dissertation focuses on using multi-objective evolutionary algorithm for solving multimodal optimization problems of this field, after a lot of research on the various properties of multi-modal function traits, in special with a priori information for multimodal optimization problem—multiple solutions of nonlinear equation system proposed a multi-objective changing model of, and using multi-objective evolutionary algorithm to solve, and then extended to the general multi modal optimization problems, and design the multi objective evolutionary algorithm. Finally, has proposed solving multi-modal optimization problems are multi-objective evolutionary algorithm for proposed a multi-objective evolutionary algorithm based on tree structure to improve the evolutionary efficiency. The main research works and innovative achievements are summarized as follows:

(1) Bi-objective transformation method which converted multiple solutions of the nonlinear equation system which is a special multi-modal optimization problems into a multi-objective problems.is presented. This method will be a

solution of non-linear system of equations solution mapping to a known shape optimization problem of the frontier and are given to demonstrate the multi-solution of a nonlinear system of equations all solutions in the the frontier of bi-objective optimization problem, and directly use multi-objective evolutionary algorithm to solve the problem. To avoid settings of niching radius in the framework of single objective optimization, the efficiency is high, and due to the multi-objective optimization algorithm results is limited populations of approximate infinite Pareto optimal solution set. This method is very natural for infinite multiple solutions of nonlinear equations of system are good approximations.

(2) A multi-objective framework for multi-modal optimization is presented. Using bi-objective transformation method for solving multimodal optimization problems will cause two problems to decrease the ability of locating multiple optima: dimension of the occlusion problem and peak attraction problem. In order to alleviate those two problems negative effect for search multiple optimums, a newly dominated relationship is presented for solving multi-modal optimization problems. Experiments are implemented on the CEC2013 benchmark, Compared to classical multi-modal optimization evolutionary algorithm compared, and multi-objective evolutionary optimization method for multimodal optimization problem, this multi-objective framework can effectively locate multimodal optimization problem of multiple optimal value.

(3) Proposed a data structure which names divide-dimensional non-dominated sorting tree. It is mainly used multi-objective optimization framework of non-dominated-sorting and embedded to the multi-objective evolutionary algorithm for solving multimodal optimization problems. This method has faster running speed and efficiency, which has the time complexity $N \lg N$.

Keywords: Multi-modal Optimization Problems; Multi-objective Evolutionary Algorithm; Multi-objective Optimization Problems; Optimum

Classification: TP391

目 录

第一章绪论	1
1.1 课题研究背景与意义	1
1.2 本文研究内容与结构安排	3
1.2.1 本论文的主要工作	3
1.2.2 本文的创新点	5
第二章 相关工作与定义	7
2.1 进化算法	7
2.1.1 进化范式	7
2.1.2 差分进化算法	8
2.2 基于进化算法求解多模态优化问题	9
2.2.1 小生镜策略	10
2.2.2 其它的单目标进化算法求解多模态优化问题	11
2.3 多目标进化算法	12
2.3.1 多目标进化算法相关定义	13
2.3.2 基于多目标进化算法的多模态优化问题的方法	14
2.4 非线性方程系统极其进化求解	15
2.4.1 基于单目标多模态优化方法	16
2.4.2 基于约束的优化方法	17
2.4.3 多目标优化方法	17
2.4.4 非线性方程与多模态优化问题	18
第三章 使用多目标进化算法定位非线性方程组的多个解	19
3.1 多目标优化方法相关工作	20
3.2 基于多目标优化算法定位非线性方程组的多个解	21
3.2.1 多目标转换函数	21
3.2.2 多目标转换函数的一个实例	23
3.2.3 不同类型的定位函数的影响	24
3.3 求解非线性方程组系统的算法流程	26
3.4 实验研究	28
3.4.1 测试平台的设计	28
3.4.2 性能指标	28
3.4.3 与 CA 方法的比较	31
3.4.4 与 CA 方法收敛行为的比较	36
3.4.5 与单目标的进化优化方法比较	39

3.4.6 不同形状的前沿面实验	40
3.5 结论	42
第四章 多模态优化问题的多目标进化求解	43
4.1 双目标转换方法	43
4.2 多目标目标转换框架	46
4.2.1 维数遮挡问题	47
4.2.2 吸引问题	48
4.3 算法整体流程与复杂度分析	49
4.3.1 算法的时间复杂度分析	50
4.4 实验	51
4.4.1 性能衡量指标	51
4.4.2 MMMOP 算法结果	52
4.4.3 MOMMOP 的支配关系比较	54
4.4.4 MOMMOP 的参数辨识	58
4.4.5 MOMMOP 与 CEC2013 的四个方法比较	61
4.4.6 MOMMOP 与经典的单目标进化算法比较	63
4.4.7 MOMMOP 与多目标优化方法的比较	65
4.6 小结	66
第五章 基于分维树的多目标进化求解技术	68
5.1 动机	68
5.2 基于分治法的双目标非支配排序树	70
5.3 分维的非支配排序树	75
5.4 时间复杂度分析	79
5.5 实验	80
5.5.1 运行速度比较	80
5.5.2 TMMOP 的实验结果	82
5.6 本章小结	83
第六章 总结与展望	85
6.1 论文工作总结	85
6.2 进一步的研究方向	86
参考文献	88
攻读博士学位期间所发表的论文	100
致 谢	101

表目录

表 2-1 不同的小生镜策略 11

表 2-2 多目标进化算法求解多模态优化问题比较 15

表 3-1 实例个体的信息..... 24

表 3-2 非线性方程系统的测试函数集..... 29

表 3-3 CA 方法与 MONES 方法实验结果对比 30

表 3-4 与单目标进化优化方法的比较..... 40

表 3-5 不同形状的前沿面的比较..... 41

表 4-1 测试函数性质 51

表 4-2 种群和函数评价次数设置 51

表 4-3 MOMMOP 在不同精度的 PR 值和 SR 值..... 52

表 4-4 比较 MOMMOP, MOMMOP1 和 MOMMOP2..... 54

表 4-5 MOMMOP 与 MOMMOP1 和 MOMMOP2 的 WILCOXON 检验 55

表 4-6 MOMMOP 与 MOMMOP1 和 MOMMOP2 的 FRIEDMAN 检验 56

表 4-7 比较 MOMMOP 不同的距离参数的 PR 值..... 59

表 4-8 比较 MOMMOP 不同的距离参数的..... 59

表 4-9 比较 MOMMOP 不同的距离参数的 FRIEDMAN 检验..... 60

表 4-10 比较 MOMMOP 不同的放缩因子的 PR 值..... 60

表 4-11 MOMMOP 不同的放缩因子的..... 61

表 4-12 MOMMOP 不同的放缩因子的 FRIEDMAN 检验..... 61

表 4-13 MOMMOP 与 CEC2013 的方法比较..... 62

表 4-14 MOMMOP 与 CEC2013 的四个方法 WILCOXON 检验 63

表 4-15 MOMMOP 与 CEC2013 的四个方法 FRIEDMAN 检验 63

表 4-16 MOMMOP 与 NCDE, NSDE, LIPS, AND R2PSO 的比较 63

表 4-17 MOMMOP 与 CEC2013 的四个方法 WILCOXON 检验 64

表 4-18 MOMMOP 与 CEC2013 的四个方法 FRIEDMAN 检验..... 65

表 4-19 MOMMOP 与多目标进化方法的比较..... 65

表 4-20 MOMMOP 与多目标方法 WILCOXON 检验..... 66

表 4-21 MOMMOP 与多目标方法 FRIEDMAN 检验..... 66

表 5-1 转换的实例的目标空间..... 69

表 5-2 用于比较运行速度测试函数集 80

表 5-3 TMMOP 在不同精度的 PR 值和 SR 值..... 82

表 5-4 TMMOP 与多目标进化算法的比较实验..... 83

图目录

图 1-1 工作思路.....	4
图 2-1 差分进化算法的框架.....	9
图 2-2 基于 Pareto 多目标进化算法的一般框架.....	14
图 2-3 非线性方程系统的求解模型.....	16
图 3-1 定位函数的关系示意图.....	21
图 3-2 CA 和 MONES 的多目标转换方式的差别.....	24
图 3-3 定位函数在目标空间的关系(纵坐标为函数值, 横坐标为决策变量值)	25
图 3-4 Pareto 前沿的不同形状(纵坐标为函数值, 横坐标为决策变量值) ..	25
图 3-5 个体间的聚集距离.....	28
图 3-6 30 次运行平均的 IGD 值进化曲线图.....	32
图 3-7 CA 方法 30 次运行目标空间最终结果分布	33
图 3-8 MONES 方法 30 次运行目标空间中最终结果分布	33
图 3-9 30 次变量空间中 MONES 最终结果分布	34
图 3-10 30 次变量空间中 MONES 最终结果分布	35
图 3-11 CA 方法 F1 函数决策空间一次运行	35
图 3-12 CA 方法 F2 函数决策空间一次运行	35
图 3-13 CA 方法 F2 函数决策空间一次运行	35
图 3-14 CA 方法 F1 函数决策空间一次运行	36
图 3-15 CA 方法 F2 函数的目标空间一次运行	36
图 3-16 CA 方法 F3 函数的目标空间一次运行	36
图 3-17 CA 方法 F3 函数的目标空间一次运行	36
图 3-18 CA 方法在 F5 函数决策空间一次运行.....	37
图 3-19 CA 方法在 F5 函数目标空间一次运行.....	37
图 3-20 CA 方法在 F6 函数目标空间一次运行.....	37
图 3-21 MONES 方法在 F1 函数目标空间一次运行.....	37
图 3-22 MONES 方法在 F1 函数目标空间一次运行.....	38
图 3-23 MONES 方法在 F3 函数目标空间一次运行.....	38
图 3-24 MONES 方法在 F4 函数目标空间一次运行.....	38
图 3-25 MONES 方法在 F5 函数目标空间一次运行.....	38
图 3-26 MONES 方法在 F6 函数目标空间一次运行.....	39
图 3-27 F3.函数不同形状的最终结果.....	41
图 4-1 原始多模态函数图像及其所对应的多目标转换的目标空间	44

图 4-2 MOMMOP 的选择原理	46
图 4-3 Vincent 函数示意图	47
图 4-4 变量组合函数	49
图 4-5 平均数量的最优值进化代数图	54
图 4-6 MOMMOP1 的一次典型运行	56
图 4-7 MOMMOP2 的一次典型运行	57
图 4-8 MOMMOP 的一次典型运行	58
图 5-1 实例的位置和支配关系示意图	69
图 5-2 支配区域关系图	71
图 5-3 节点结构	72
图 5-4 双目标非支配排序的目标空间	74
图 5-5 第一次调用节点分裂算法树的结构	75
图 5-6 最终所形成的双目标非支配排序树	75
图 5-7 实例的双目标排序树	77
图 5-8 实例的分维多目标排序树	78
图 5-9 CF1 运行平均时间比较	81
图 5-10 CF1 运行平均时间比较	81
图 5-11 CF1 运行平均时间比较	81

第一章绪论

从 Holland 提出进化算法的最早范式遗传算法的概念, 到进化算法投入到各种生产实践的方方面面, 进化算法已经存在已近半个世纪^{[1][2]}。在这半个世纪以来, 各种不同的进化范式层出不穷, 如粒子群算法^[3], 差分进化算法^[8], 蚂蚁群落算法^[7], 免疫算法^[9]等。在某种程度上可以说进化计算是由于实际的生产实践活动所推动, 如遗传算法的提出是在 Holland 设计生物自适应系统求解模式分类的问题中提出的, 而 Rosenberg 等在进行风洞实验时, 形成了进化策略。无数的文献和生产实践活动表明进化计算已经进入到了人们实际生活和科研活动的方方面面^{[1][2][3][4][5]}。

在实际生产生活中, 使用者在优化过程中往往是求取一个全局最优解或者是满意解, 但是随着实践活动的深入, 有些全局最优解在实际中由于约束或者鲁棒等条件的限制往往并不适用, 在另外一方面来讲, 一些全局最优解有助于使用者更好的发现问题的特征, 因此需要求出更多的全局最优解和局部极值解备选, 从而形成了多模态优化领域。但是传统的数学方法往往只是从一个解进行寻优, 无法一次运行定位多个最优解。从生物行为启发而形成的进化算法, 由于在算法运行时维持种群的概念, 其潜在的天然优势能够利用种群而不是一个个体定位多个最优解^{[10][11]}。因此进化算法成为了多模态优化问题的求解主流。

进化算法在进行多模态寻优时, 往往借助小生境的概念划分种群, 从而维持种群分布在整个的搜索空间, 达到定位多个最优值的目的, 形成了所谓的小生境进化算法研究方向^{[11][136][137][138][140]}。但是随着社会的发展, 现实中对所要优化问题的复杂度越来越高, 因此需要性能和效率更好的算法, 目前对于多模态函数的寻优放在单目标框架下, 一般需要对问题的最优值数目等先验经验, 从而造成算法在求解实际优化问题时性能和效率的下降^{[11][139][141]}。需要一种新的方法和框架定位多模态优化问题的多个最优值, 本文的主要工作是将多模态优化问题放在多目标优化框架下而不是在单目标的框架下, 利用多目标优化问题的特征和多目标算法对前沿面较好的搜索性能对多模态函数优化问题进行寻优。

1.1 课题研究背景与意义

《智能制造科技发展“十二五”专项规划》指出中国目前已是世界第二大经济体和制造业大国, 但自主创新能力薄弱、先进装备贸易逆差严重、高端装备与智能装备严重依赖进口, 严重制约我国制造产业健康发展, 其中国内大多数企业在生产制造过程中一定程度地应用了自动化技术, 但应用于提高产品质量、实现节能减排、提高劳动生产率的智能化技术严重缺乏。智能优化提高产品质量和生产效率, 显著减少物耗、能耗和排放, 是我国智能制造业发展的必经之路。进化算法作为智能优化算法的一个

分支，必将在我国智能制造业腾飞之路上产生重大的作用。

多模态优化 (Multi-modal optimization)不同的学科有着不同的定义,本文根据 wiki 百科的定义:多模态优化是处理优化任务时,相反于只发现一个最优值,需要发现全部或者大部分的优化问题的最优解(或者局部最优解)。多模态优化问题大量存在于实际的应用当中,如全息光栅设计^[12]、机器人路径规划^{[22] [18]}、机械和电气设计^{[13][14]}、聚类^{[17] [19][20] [21]}、经济博弈论^{[15] [16]}等诸多实际优化问题,这些问题普遍存在着不止一个全局最优解和多个局部极值解,因此需要构造一种算法,使之能够求出所有全局最优解和尽可能多的局部极值解。多模态优化问题的最优解的数目往往不止一个,在实际应用的场合中需要对这些解全部找出来,以便满足决策者使用其中一个或者全部,如在全息光栅设计中,光学工程师需要不只需要一个满足约束条件的最优或者次优解,而是需要更多的解决方案以便进一步研究和尝试。而在机械和电气设计问题中,为了消除大幅共振对系统带来的消极影响,需要找到振幅在某一特定阈值以上的所有共振频率,而这些解往往不只一个。另外还有一些特殊的场合,需要把所有的解都能找出来,以帮助使用者发现新的现象,以便更好的把握问题、分析问题的本质,如聚类问题中,一些局部次优解的分布有利于展示不同聚类问题的隐藏特点,机器人路径规划中需要找出大量的较优路径从而交给机器人进行决策。因此对于多模态优化问题的研究有着重大的实际意义和理论价值。

对于多模态优化问题的求解,传统数学方法并不适用,首先在优化时需要先验知识,包括梯度、可微可导等信息,最主要的是这些方法对初始解进行迭代寻优,不能一次运行得到多个解^[5]。而进化算法维持种群的操作,对问题直接进行编码,并不需要问题的先验知识,只需要对种群结构进行小幅度的修改,即可适合求解多模态优化问题,对于多模态优化问题的进化求解技术,产生了小生境进化算法的研究方向。由于多模态优化问题的实际意义和理论价值,许多国际会议都主办了对于多模态优化问题的进化求解技术的专题讨论会,如 IEEE 群智能会议(Swarm Intelligence Symposium, SIS), IEEE 进化计算会议(Congress on Evolutionary Computation, CEC)等,并产生了针对多模态优化问题求解的大量的研究成果。在举办这些专题讨论会之外,为了更好的研究多模态优化问题的进化求解性能,以及对不同算法进行评测分析,加快多模态寻优技术发展步伐,满足实际应用需求的步伐,一些标准化的测试及评价平台相继提出。如在 2010 年 IEEE 进化计算会议(CEC2010)上,该方法用于生成多模态优化标准测试函数库。另外 2013 年 CEC 2013 小生境技术多模态优化特别专题研讨会提供了 20 个具有不同特点和困难度的标准多模态测试函数库^[23]。此类函数库中的多模态优化问题,除了来自于工程实践的问题外,还设计了一些进化算法难以解决的变量耦合问题,基于此类函数库,CEC2013 开展了基于小生态进化算法的竞赛,最终差分进化算法得到了较好的结果。在作者撰写论文时,CEC2016 正在召开针对小生境技术求解多模态优化问题的专题讨论会,并发布了最新的测试函数集。以上表明多模态优化领域逐渐

成为优化领域一个相对较热的研究方向，并且进化计算在多模态优化问题的求解上已经成为了主流。

多模态优化问题的多个最优值一般存在如下几个特征：1 最优解集的数目存在着差异，从无限个解到有限的几个解都广泛的存在。2 各个最优解之间在空间中分布并不均匀。3 问题的维数往往较高，在高维空间中种群中个体的邻居信息和分布信息扭曲，并且由于问题维数的存在需要大量的计算资源。在单目标进化框架内需要个体之间的距离信息和分布信息划分群体的进化空间，在无法得到所求解的个数和分布等先验信息的情况下，往往会丢失一部分解，因此有必要寻找另外的框架和模型求解多模态优化问题。

由于多目标优化问题本质上是求解近似一组相互均衡且在前沿面分布均匀的无限解集，国际上一些研究者很自然的将多模态优化问题使用多目标进化算法求解。已经出现了一些工作，都取得了不错的结果，引起了学术界的相当关注，如专门刊载的进化计算相关算法的国际权威期刊《IEEE Trans Evolutionary Computation》和 MIT 的《Evolutionary Computation》近几年发表了几篇^{[24][25][26][27]}关于多目标进化算法求解多模态优化问题的相关算法，但是值得指出的是，这些方法使用的多个目标并不是相互冲突的，也不能在理论上保证其最优值就在其前沿面上。因此设计一种方法，确保其最优值在其 Pareto 的前沿面上，是求解多模态优化问题的关键。

本文首先从带有先验信息的多模态优化问题—多解的非线性方程组出发，研究了其多目标转换方法，然后将此转换方法用在了更一般的多模态优化问题上，最后开发出更为高效的多目标进化算法求解多模态优化问题转换而来的多目标优化问题。

1.2 本文研究内容与结构安排

1.2.1 本论文的主要工作

本论文针对多模态优化问题的单目标优化方法的不足，系统地研究了多模态优化问题的多目标进化算法求解技术。本论文的工作思路是连贯的，在研究了大量的各种多模态优化问题的特征的基础上，选择了带有先验信息的多模态优化问题-多解的非线性方程系统，提出了一种泛化的双目标转换函数，此函数可以理论上证明多模态优化问题的最优值都在其 Pareto 前沿面上。使用多目标进化算法对多解的非线性方程组问题进行寻优，在实验中详细的对比了已有的算法，分析了此转换方法和多目标进化算法的相关特征。在求解多解的非线性方程组的基础上，将此双目标转换函数用于更为普遍的多模态优化问题的寻优，在此过程中，发现存在着两个新的问题：非支配集的吸引性问题和维数的遮挡问题，为了解决此两个问题，设计了相关的机制和新的多目标转换函数，并使用了搜索性能较好的差分进化算法，使用了 CEC2013 的 20 个测试集进行对比实验，与当今求解多模态优化问题的进化算法相比，本文提出的方法在求

解精度和所求得的最优值数量都具有明显的优势。为了进一步将本文所提出的算法投入到实际的应用，本文针对所提出求解多模态优化问题的多目标进化算法时间复杂度较高的问题，在分析了问题的特征和多目标转换函数的信息的基础上，提出了一种新的针对多模态函数转换的多目标优化问题排序算法，采用了一种基于树结构的支配方式，有效的提高了运行效率，降低了时间复杂度。其工作思路如图 1-1 所示。

本论文的工作主要基于以下三个内容：

- 1 针对特殊的多模态优化问题—多解的非线性方程组的提出了一种双目标的转换方法，在此工作上设计多目标进化算法求解。
- 2 针对一般的多模态优化问题，在分析了双目标转换方式求解多模态优化问题的不足的基础上，提出了一种将其转换为 D 个双目标优化问题的转换方法，并设计了一种个体之间新的支配方式。
- 3 为进一步减少其时间复杂度，提高其运行效率，提出了一种基于树结构的多目标进化算法。

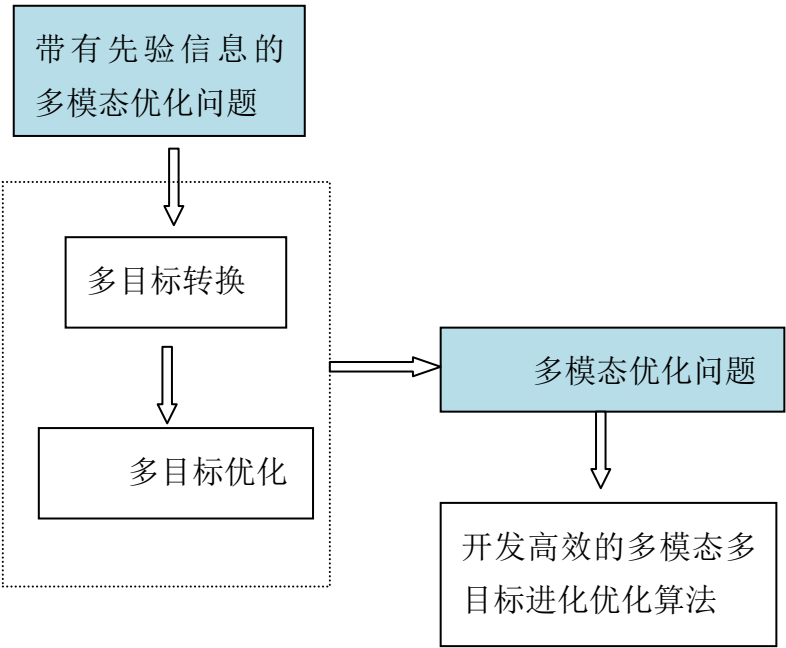


图 1-1 工作思路

论文各章节的内容安排如下：

第一章，分析了求解多模态优化问题的研究背景和意义，引出本文的工作思路，并概括本文的研究内容。

第二章，介绍当前国内外多模态优化问题的研究现状，分析了在单目标框架下的多模态优化方法的不足，指出多目标进化算法求解多模态优化问题为当前多模态优化领域的一个热点。

第三章,研究了多解的非线性方程的转换优化模型和相关进化算法,包括多模态优化模型、约束优化模型、多目标优化模型,指出其不足。提出一种双目标转换方法,通过实例分析了此双目标转换方法的特点,分析了不同形状的双目标转换方法对于多目标进化算法求解非线性方程问题的不足。设计了一个不同特征多解的非线性方程系统的测试函数集,在此基础上进行大量而充分的实验对比了同类的多目标优化方法的特征和收敛特性,并进行了对比单目标进化方法求解非线性方程系统的实验。实验结果证明了本文所提出的方法的有效性。

第四章,研究了一般的多模态优化问题的多目标优化进化方法,在此过程中,发现一些多模态优化问题存在着两个新的问题:峰的吸引性问题和维数的遮挡问题,指出种群进化过程中此 2 类问题将导致多目标进化算法在寻优过程中丢失最优值,并对此两类问题做出相关分析,设计了将一个多模态优化问题转换为 N 个双目标优化问题的多目标转换框架,以及加入个体邻域信息的支配方式,并使用了搜索性能较好的差分进化算法作为搜索引擎。在实验中,使用了 CEC2013 的多模态优化问题进化求解技术专题讨论会所提出的测试集,与当今求解多模态优化问题的进化算法相比,本文提出的方法在求解精度和所求得的最优值数量都具有明显的优势。

第五章,在探讨了以上求解多模态优化问题的多目标的进化算法的时间复杂度过于复杂的情况下,提出了一种类似于分治法的双目标非支配排序方法,定义了双目标非支配树的数据结构,使用自上而下的方式构造双目标支配树。在此基础上提出了求解多模态优化问题的分维支配树的数据结构,分析了每一次迭代的时间复杂度为 $N \lg N$,其中 N 为种群的大小,从而与维数无关,实验证明了所提出的方法的有效性

第六章,对本文的研究内容进行了系统总结,对研究中尚未解决的问题进行了归纳,并明确了下一步研究的方向。

1.2.2 本文的创新点

(1)提出求解多解的非线性方程系统这一特殊多模态优化问题的双目标转换方法。

在此种方法中将一个多解的非线性方程系统的解映射到一个形状已知的双目标优化问题的前沿面上,提出了一种泛化的双目标进化算法。避免了在单目标框架下需要小生境半径的设置等问题,其求解效率高,并且由于多目标优化算法的结果是用有限的种群近似无限的 Pareto 最优解集,此方法很自然的对于无限多个解的非线性方程系统进行良好的近似。

(2)提出求解多模态优化问题的多目标转换方法。

当使用双目标转换方法求解一般的多模态优化问题,发现存在着 2 个问题,维数的遮挡问题和峰的吸引问题,提出了一种将多模态优化问题转换为 D 个双目标优化问题的方法,并提出了一种新的支配方式,用于求解多模态优化问题。此多目标优化进

化方法求解精度高，对多模态优化问题的多个最优值定位效率高。

(3)提出了一种分维支配树的新的数据结构用于多模态优化问题所转换过来的多目标优化框架的非支配排序，并将其嵌入到多目标进化算法中求解多模态优化问题。此种方法具有较快的运行速度。

第二章 相关工作与定义

本文在多目标进化框架内求解多模态优化问题，从多模态优化问题与多目标优化问题的求解存在着相似性出发，在研究了多模态优化问题的特征的基础上，针对特殊的多模态优化问题-非线性方程系统，提出了一种泛化的转换框架，在此转换框架内，探讨了更为一般的多模态优化问题的求解方法，并设计了相关的多目标进化算法，在所提出的多目标转换框架的基础上，设计了专门针对多模态优化问题的多目标进化算法，提高其运行效率，下面将介绍使用进化算法求解多模态优化问题的相关工作及其相关定义。

2.1 进化算法

2.1.1 进化范式

进化算法(EA: Evolutionary Algorithm)^{[4] [5]}是一类模拟生物自然选择与自然进化的随机搜索算法，在人工智能领域将其归类于计算智能的范畴。由于其在问题空间中进行群体搜索，求解高度复杂的非线性问题而得到了非常广泛的关注，同时它又具有较好的通用性。在解决只有单个解的复杂系统优化问题时，不需要问题额外的经验，将优化问题作为一个黑箱子，充分展现了进化算法的优势，算法在比较适应值时采用全序的方式进行比较，对于带有多个解的特征的问题，需要改变进化算法的结构，达到求解多个解的目的。进化计算能适应不同的环境，在大多数情况下都能得到比较满意的有效解。对问题的整个参数空间给出一种编码方案，而不是直接对问题的具体参数进行处理，不是从某个单一的初始点开始搜索，而是从一组初始点搜索。搜索中用到的是目标函数值的信息，可以不必用到目标函数的导数信息或与具体问题有关的特殊知识。因而进化算法具有广泛的应用性，高度的非线性，易修改性和可并行性^[5]。

随着早期的生产实践，出现了各种形式的进化范式，其中遗传算法(Genetic Algorithms)、遗传规划 (Genetic Programming)、进化策略(Evolution Strategies)和进化规划(Evolution Programming)是进化算法的四种典型方法^{[3][5]}。遗传算法是其中最为成熟的方法，广泛的应用于各种实际问题。这四种进化范式有着联系，都是借鉴于生物模型，从一组初始点开始搜索，但是也有着区别，首先遗传算法的主要基于概率的选种、交配和突变，而其它的几种方法，进化机制源于选种和突变。就适应度的角度来说遗传算法用于选择优秀的父代，认为这些优秀的父代才能产生更优秀的子代，而进化规则和进化策略则用于选择子代(优秀的子代才能存在)。遗传算法与遗传规划强调的是父代对子代的遗传链，而进化规则和进化策略则着重于子代本身的行为特性，即

行为链。遗传算法采用二进制编码，在迭代过程中需要有编码—解码过程，而进化规则和进化策略一般采用实数编码，省去了迭代过程中的解码过程，其直接对应于连续优化问题的变量空间的适应值景观，而对于非数值优化需要做一些改动，故而较为适合求解连续优化问题，但是随着进化算法的发展，遗传算法采用实数编码也较为常见。另外进化策略可以确定机制产生出用于繁殖的父代，而遗传算法和进化规则强调对个体适应度和概率的依赖。

除了以上所提到的几种进化范式外，近二十多年来，出现了一些更为优秀的进化算法范式(Evolutionary paradigm)，包括蚂蚁群落算法^[7]、粒子群算法^[6]、差分进化算法^[8]、萤火虫算法^[49]、人工免疫算法^[9]等，这几种进化算法一般将其归结为群体智能的范畴，有着较强的生物协同机制，如蚂蚁算法是模拟蚂蚁寻找食物过程中信息素的挥发机制，而粒子群模拟鸟群的捕食行为。对比如遗传算法，其实现更加简单，收敛速度更快等特点，由于其独特的搜索特征吸引大量的研究者关注，产生了大量的研究工作。由于多模态优化问题的求解意义和价值，这些算法被大量的用于求解多模态优化问题，其中由于差分进化算法无论是在鲁棒性、可实现性还是进化参数的数量来说都有着较大的优势，特别是在各类问题如全局优化问题（CEC2015）、约束优化问题（CEC2010）等，都获得了不错的结果，都有着广泛的应用。

2.1.2 差分进化算法

差分进化算法最初由 R.Storn 在求解切比雪夫不等式时提出，是一种非常有效的针对实数的优化问题的群集智能算法，由于其高效性，在进化社区非常频繁的用于求解优化问题，包括多模态优化问题^[31]，大规模全局优化问题^[29]，多目标优化问题^[28]，约束优化问题^[30]等。差分进化在搜索变量空间的化过程中，维持父种群和子种群两个种群，种群的代沟模型是子个体与父个体进行比较、选择生成下一代种群的个体，实行贪婪选择，是一种稳态的进化算法。需要指出的是，在多目标进化算法中，由于使用偏序选择，使用的是群体替代模型，而不是稳态选择。差分进化算法在单目标进化时，种群代沟模型如图 2-1 所示。初始种群为决策空间 S 内的均匀随机数产生的个体 $X_{i,0}$ 构成，在进化的过程中每一代 G 的每个个体 $X_{i,G}$ 称为目标矢量(target vector)。DE 根据父代个体间的矢量进行变异(Mutation)、交叉(crossover)和选择(selection)操作，其中变异操作产生变异矢量 (mutant vector) $V_{i,G} = (v_{i1,G}, v_{i2,G}, \dots, v_{im,G})$ ，变异矢量与目标矢量交叉生成试验矢量 (trial vector) $U_{i,G} = (u_{i1,G}, u_{i2,G}, \dots, u_{im,G})$ ，试验矢量再与目标矢量通过选择操作产生下一代个体 $X_{i,G+1}$ 。

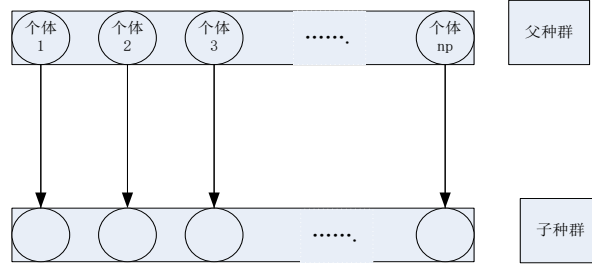


图 2-1 差分进化算法的框架

差分进化算法的变异操作通常有如下 5 种策略实现[8]。

“DE/rand/1”:

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \quad (2-8)$$

“DE/best/1”:

$$V_{i,G} = X_{best,G} + F \cdot (X_{r1,G} - X_{r2,G}) \quad (2-9)$$

“DE/current to best/1”:

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r1,G} - X_{r2,G}) \quad (2-10)$$

“DE/best/2”:

$$V_{i,G} = X_{best,G} + F \cdot (X_{r1,G} - X_{r2,G}) + F \cdot (X_{r3,G} - X_{r4,G}) \quad (2-11)$$

“DE/rand/2”:

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) + F \cdot (X_{r4,G} - X_{r5,G}) \quad (2-12)$$

其中，下标 $r1, r2, r3, r4, r5$ 为 5 个互不相等并且不等于 i 的处于 $[1, NP]$ 范围内的均匀随机整数， $F \in [0, 2]$ 为差分矢量的尺度因子， $X_{best,G}$ 为第 G 代种群中具有最佳适应度的个体。上述 5 种策略中，“DE/rand/1”是由于其实现简单，全局搜索能力强从而成为使用得最多的一种。在本文中，求解多模态优化问题时将使用 DE/rand/1 作为搜索引擎。

2.2 基于进化算法求解多模态优化问题

使用进化算法求解多模态函数的研究，不但对于多模态函数的求解产生了较好的效果，并且在某种程度上大大丰富了进化算法的基础理论和研究范围，形成了所谓的小生态进化算法的研究方向。为了使原始的进化算法在多模态优化问题的适应值景观下进行搜索，非常多的研究者设计了不同的技术，使进化种群趋向于不同的搜索方向，从而获得种群的较好的分布，使算法能够一次运行定位多个解。

在 1970 年 Cavicchio^[32]提出的与选择机制是小生镜遗传算法的第一个实现，非常多的小生镜策略被提出来求解多模态优化问题。下面将简单介绍几个比较流行的方法。

2.2.1 小生镜策略

拥挤 (Crowding)^{[32] [33] [34]}策略是最早和最简单的用来求解多模态优化问题的小生镜技术。主要思想是用适当的技术替换当前的种群个体,保持种群的分布。另外由于受限竞争选择非常相似于拥挤策略,本文将其归为一类。De Jong 在 1975 年首次提出拥挤策略的方法,此种方法允许相似的个体竞争有限的种群资源,其具体做法是子代个体与种群中父代个体进行比较,将最相似的父代个体进行比较,并进行替换操作。拥挤因子 CF 被用来计算父代种群中的个体大小。此方法的运算速度较快,但是经常出现替代的错误。针对原始的拥挤策略经常发生替代错误,Mahfound^[34]和 Mengshoel^[44]分别提出了改进的办法,但是仍然避免不了替代错误的发生。

适应值共享^[35]最先由 De Jong 提出,主要思想是根据个体的相似度将种群分割从不同的子种群,降低种群中相似个体的适应值,修改适应值函数曲面形状来限制相似个体在种群中的过度增长定位函数的多个解。修改的曲面用共享适应值描述,按照共享适应值决定个体的选择概率^[35]。适应值共享的时间复杂度为 N^2 。

不同于适应值共享策略,清洗策略^[36]采用在每一个小生镜中,清除掉最差的个体,保存最好的个体。算法在运行之前,预先定义小生镜的半径,然后对种群根据适应值进行排序,将最好的个体从种群中选择出来,在其选择出来的个体的小生镜半径之内的所有个体,直到种群中不存在着其它的个体。清洗策略在进化过程中能够保持小生镜中最好的个体,但是其收敛速度较慢,并且不能较好的定位多模态优化问题的多个最优值。

借助于自然演化的概念,物种演化 (Speciation)^[37]使用物种种子的概念求解多模态优化问题,在进化过程中,每一个小生镜的中心称之为物种种子,借助于小生镜半径的概念,将小生镜半径之内的个体称之为一个群体,以此方式种群中的个体被划分为若干组。此外借助于历史上的记忆,在划分物种时利用种群曾经所搜索到的个体的拓扑结构划分物种可以在某种程度上避免参数的引入^[38]。

除了以上的小生镜策略外还有标签技术^[39],动态聚类技术^[40],合作进化共享^[41]序列小生态技术^{[42] [43]}等相继提出。为了更清晰的表示概念,表 2-1 总结这些小生镜策略的时间复杂度以及特征等。除此之外,这些小生境策略被嵌入到无数的进化范式中求解多模态优化问题,包括遗传算法^[44]、人工免疫系统^[45]、粒子群优化^[46]、差分进化算法^[47]、进化策略^[48]、萤火虫算法^[49]等。另外对于多模态优化问题的收敛性研究,也已经取得了一些成果^{[136] [137] [138] [139] [140] [141] [142]}。

表 2-1 不同的小生镜策略

方法	小生镜参数	特征	缺点	时间复杂度
拥挤策略	拥挤因子	简单，易运行	求解性能较差	CN
适应值共享	共享半径	能够维持稳定的小生镜	小生镜半径较难 以获得	N^2
清洗策略	清洗半径	简单并且能够获得小生 镜中较好的个体	收敛速度较慢	N^2
物种形成	小生镜半径	能够维持稳定的小生镜	小生镜半径较难 以获得	N^2

2.2.2 其它的单目标进化算法求解多模态优化问题

下面简单介绍最近几年来单目标进化算法进化算法求解多模态优化问题的研究工作。Preuss^[50] 提出了适应性小生境协同矩阵进化策略(CMA-ES)，在 CMA-ES 中^[51] 采用了局部搜索算法，另外所提出的方法包括 2 个部分。第一个是文中所提出的最近的聚类 (nearest-better clustering)，第二个是所定义的的高水平策略。Li^[51]在研究了粒子群算法的能力的基础上使用环状拓扑的结构求解多模态优化问题(PSO)^[52] 根据文献的实验和分析，此种算法能够维持较为稳定的小生境能力。Epitropakis 等^[53] 使用了 2 个变异策略的差分进化算法，提高了差分进化算法的形成小生境，其中最近邻居的本征向量被使用为基向量。Qu 等^[54]提出了基于邻居变异的差分进化算法，此种进化算法使用了三种小生境策略求解多模态优化问题。同时 Qu 等^[55] 提出了局部搜索粒子群算法中粒子的局部最优策略，并将其应用于 3 个采取不同小生境策略的粒子群算法求解多模态优化问题，证明此方法能够提高粒子群的搜索和收敛能力，另外 Qu^[56]采取了邻居信息改进原始粒子群的飞行速度，另外采取了一个基于局部距离的粒子群算法求解多模态优化问题。最近 Biswas^[57]提出了父亲中心的归一化变异与基于近似度拥挤距离的差分进化算法，在进化过程中开发邻居信息，从而使搜索导向多个最优值。

另外使用外部集策略在求解多模态优化问题时引起了一些关注，这是因为外部集动态调节能够适应不同的多模态优化问题，避免使用固定的种群定位个数相差较大的多模态优化问题的最优值。其中这些策略有：在差分进化算法的基本的变异策略的基础上，Epitropakis^[58]整合了 2 个新的机制，第一个机制是适应性差分进化算法（类似于 JADE^[59]），第二个机制是动态的档案集策略。Molina^[62]提出了小生境变化的网格优化方法，此方法使用了清洗技术协同适应性小生境以及外部集保存所找到的当前最好的解。Xu^[63]引入吸引盆估计遗传算法，使用探测多模态技术去估计小生境的半径，并使用所估计的小生境半径形成种群，另外一个种子外部集策略去探测不同的峰。

多模态优化问题一般存在如下几个特征：1 最优解集的数目存在着差异，从无限

个解(非线性方程系统问题经常遇到无限个解得情况)到有限的几个解都广泛的存在。

2 各个最优解之间在优化空间中分布并不均匀。3 问题的维数往往较高,在高维空间中种群中个体的邻居信息和分布信息扭曲,并且由于问题维数的存在需要大量的计算资源。在单目标进化框架内需要个体之间的距离信息和分布信息划分群体的进化空间,在无法得到所求解的个数和分布的情况下,往往会丢失一部分解,从而使得算法求解性能变弱。因此有必要寻找另外的框架和模型求解多模态优化问题。由于多目标优化问题本质上是求解近似一组相互均衡且在前沿面分布均匀的无限解集,因此多模态优化问题天然的适合多目标进化算法求解。

如何有效地提高多模态优化问题的求解性能具有重大的科学与实践意义,而寻找一种方法,这种方法能够在不需要计算个体在问题空间中的分布的情况下,使种群中潜在的最优个体都能够通过选择算子保存下来继续参与迭代,是提高多模态优化问题的求解性能的关键。本论文将多模态优化问题的最优值映射到多目标优化问题的前沿面上,从而无需知道最优值数目和分布信息,避免丢失解的情况。另外本项目是将多模态优化问题转换为多目标优化问题,在保持种群分布时,只需要计算前沿面的分布状况,不需要计算种群中解个体的邻居信息,有望有效提高高维复杂问题的计算效率。

2.3 多目标进化算法

无论在科学研究或工程应用上,现实问题常常不存在着一个目标,而是多个目标共存。在理想情况下,最小化每个目标,但是此最优值是不存在的,因为这些目标耦合在一起,相互冲突,当考虑了其中一个子目标的“利益”时,其它的子目标的“利益”必然会受到损失^[1]。由于多个目标存在的实际问题非常多,因此和进化算法的出现类似,应用推动了多目标进化算法(Multi-objective Evolutionary Alogrithm, 简称为 MOEAs)的出现和发展。一般情况下, MOEAs与遗传算法的操作相同,通过对问题空间进行编码,迭代搜索,由于采用了经济学家Pareto所提出的前沿面的概念,使用Pareto支配的偏序选择,驱动种群向前沿面而不是一个点搜索,从而可以达到一次运行得到多个解的效果。

从Schaffer提出的向量评估遗传算法(VEGA: vector evaluated genetic algorithm)可以说是第一个真正意义上的MOEA后, MOEA有了很大的发展。许多种类的MOEAs不断出现。Coello Coello^[70]将MOEAs的发展分成两个阶段:第一阶段为上个世纪末,其中有VEGA^[64]^[65]、MOGA、NSGA以及NPGA。此阶段主要研究如何将进化算法用来求解多目标优化问题,故其算法相对来说比较简单。第二阶段为这个世纪至今,这期间代表性的MOEAs 有SPEA^[68]及其改进版SPEA2^[69]、PAES^[66]^[67]、NSGA的改进版NSGA-II^[63] [12,13]、PESA^[66]及PESA II^[67]。此阶段由于MOEAs相对较为成熟,故效率的衡量是算法设计者主要考虑的问题,其主要特征是精英保留机制(elitism)等有效方法

的引入。另外需要指出的是现在多目标优化问题的目标数目大于6维时（极大多目标优化问题），由于选择压等因素，导致了无法收敛于前沿面的情况出现，针对此问题出现了一些在极大多目标优化环境下的多目标进化算法，相关文献参考。虽然取得了一些进步，但是在极大目标下的优化仍然是对进化社区的一个挑战^{[87][88][89]}。

多目标进化算法正是因为它具有广泛的应用前景而发展起来的。如VEGA就是为了解决机器学习中的多模式分类判别问题而提出来的，另外大量的实际问题如环境与资源配置^[71]，电子与电气工程^[72]，通信与网络^[73]，机器人^[74]，航空航天^[75]，市政建设^[76]，交通运输^[77]，机械设计与制造^[78]，管理工程^[79]，金融^[80]等等。另外由于多目标优化问题在分布度保持上的特有的特征，以及偏序比较的概念，能够很好地解决单目标优化问题不好提取最优值的特征的一些问题，存在着很多的工作使用多目标优化算法求解这类问题^{[91][90][92]}。

2.3.1 多目标进化算法相关定义

由于在本文中所要用到的多目标优化的相关概念较多，因此在介绍多目标进化算法的同时，将给出相关的概念。一个多目标优化问题采用以下形式：

$$\begin{aligned} \underset{\vec{x}}{\text{minimize}} f(\vec{x}) &= (f_1(\vec{x}), K, f_M(\vec{x})) \\ g_i(X) &\geq 0 \quad i=1, 2, L, k \\ h_i(X) &= 0 \quad i=1, 2, L, l \end{aligned} \quad (2-13)$$

$\vec{x} = (x_1, K, x_D) \in X \subset \mathbb{R}^D$ 为决策向量， $x_i (i \in \{1, K, D\})$ 是， D 为决策变量的个数， X 为决策空间， $f(\vec{x}) \in Y \subset \mathbb{R}^M$ 为目标空间， $f_i(\vec{x}) (i \in \{1, K, M\})$ 第 i 个目标问题， M 为目标个数。 $g(X)$ 为不等式约束， $g_i(X)$ 为第 i 个约束， k 为不等式约束的个数。 $h(X)$ 为等式约束， $h_i(X)$ 为第 i 个等式约束， l 为等式约束的个数。

定义 2.1. 个体的Pareto支配关系：设 p 和 q 是进化群体Pop中的任意两个不同的个体，我们称 p 支配 q ，则必须满足下列二个条件：

- 1对所有的子目标， p 不比 q 差，即 $f_k(p) \leq f_k(q) (k=1, 2, L, r)$ ；
- 2至少存在一个子目标，使 p 比 q 好。即 $\exists l \in \{1, 2, L, r\}$ ，使 $f_l(p) < f_l(q)$ 。

其中 r 为子目标的数量。此时称 p 为非支配的(non-dominated)， q 为被支配的(dominated)。表示为 $p \phi q$ ，其中“ ϕ ”是支配关系(dominate relation)。

定义2.2. Pareto最优解：给定一个多目标优化问题 $\vec{\text{Min}} f(X)$ ，称 $X^* \in \Omega$ 是最优解，若 $\forall X \in \Omega$ ，满足下列条件：

$$\begin{aligned} \bigwedge_{i \in I} (f_i(X) = f_i(X^*)) \\ \text{至少存在一个 } j \in I, \quad I = \{1, 2, L, r\}, \text{ 使: } f_j(X) > f_j(X^*) \end{aligned} \quad (2-14)$$

其中 Ω 是满足式(1)和式(2)的可行解集，即：

$$\Omega = \{X \in \mathbb{R}^n \mid g_i(X) \geq 0, h_i(X) = 0; (i=1, 2L, k; j=1, 2L, l)\}$$

定义 3. Pareto最优面: 给定一个多目标优化问题 $\text{Min } \vec{f}(X)$ 和它的最优解集 $\{X^*\}$, 它的Pareto最优面定义为:

$$PF^* = \{\vec{f}(X) = (f_1(X), f_2(X), \dots, f_r(X)) \mid X \in \{X^*\}\} \quad (2-15)$$

多目标进化算法求解多目标优化问题时, 产生非常多的框架, 其中采用得最多的一种框架如图2-1所示, 在空间中产生初始种群, 使用进化操作(如交叉算子, 变异算子), 产生子代种群。针对每一代进化群体, 寻找出其当前最优个体(即当前最优解), 我们称一个进化群体的当前最优解为非支配解(non-dominated solution), 或称为非劣解(non-inferior solution); 所有非支配解的集合称之为当前进化群体的非支配集(NDS: non-dominated solutions), 找出非支配集的过程一般称为非支配排序, 在进化过程中, 驱动种群的非支配集NDS不断逼近真正的最优解集, 最终达到所谓的Pareto最优面, 即使 $NDS^* \subseteq \{X^*\}$, NDS^* 为算法运行结束时所求得的非支配集。

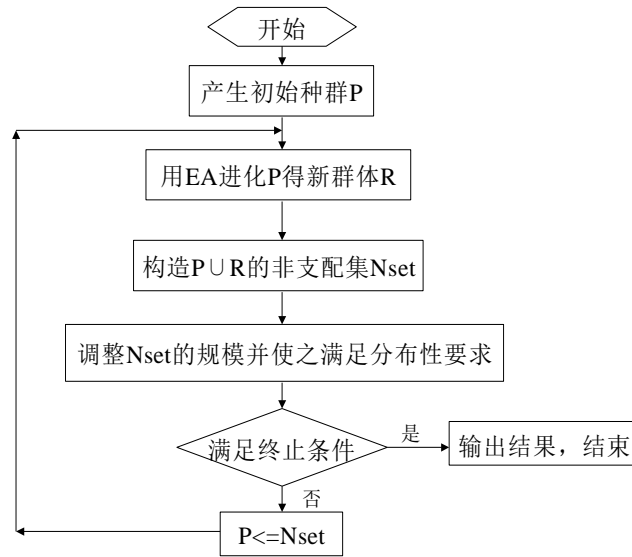


图 2-2 基于 Pareto 多目标进化算法的一般框架

2.3.2 基于多目标进化算法的多模态优化问题的方法

由于多目标进化算法的最终输出是一组在前沿面分布良好的 Pareto 最优解集, 与求解多模态优化问题相似, 很自然的可以联想到使用多目标进化算法求解多模态优化问题, 现在国际上的已有一些关于使用多目标进化算法求解多模态优化问题的方法。其一般特征是将多模态优化问题原来的目标函数不变, 将其作为转换过来的多目标优化问题的第一个目标, 而第二个目标为种群中解的特征或者分布情况等, 下面将简要介绍这些工作:

Deb.K 等^[29]首先提出了使用求解多模态函数的双目标进化算法, 将一个多模态函

数形式化为一个双目标优化问题，第一个目标为原多模态优化问题的适应值，第二个目标为种群中的个体在变量空间中的梯度信息。另外使用了非支配排序算法求解此问题，取得了不错的结果。需要指出的是此算法需要在预先探测解的梯度的情况。

Basak.A^[30]等提出了使用双目标差分进化算法求解多模态优化问题(简称为MOBiDE)，在MOBiDE中使用原目标函数为第一个目标，而第二个目标为种群中个体的平均距离，另外使用了非支配排序的选择方式，当前层的非支配集数目大于种群数目时使用了超空间指标选择适应值较大的个体。MOBiDE的主要特征是不需要问题的梯度等先验信息。

YAO[31]提出了多种群双目标遗传算法求解多模态函数优化(BMPGA)，此算法采用了三种机制求解多模态优化问题，首先使用了双目标机制提高分布，其次聚类方法将种群进行分割，最后使用多种群的方法强化分布和开发潜在最优解的区域。

需要指出的是这些工作无法保证多模态优化问题的最优值一定在其前沿面上，因此求解效率并不强，另外需要一些问题的先验信息，比如BMPGA的聚类数目，DEB的方法中个体的梯度信息。本论文的工作与之不同的是，将多模态优化问题完全视为一个多目标优化问题进行求解，而不是其中的一个目标，其转换过来的多目标优化问题的前沿面形状已知，从而避免了有关于问题特征参数引入。

表 2-2 多目标进化算法求解多模态优化问题比较

算法	是否带有问题先验信息	第一个目标	第二个目标	时间复杂度
Deb的多目标方法	是	原多模态函数	梯度信息	N^2
MOBiDE	否	原多模态函数	个体之间的平均距离	N^2
BMPGA	是	原多模态函数	梯度信息	N^2

2.4 非线性方程系统极其进化求解

非线性系统方程组(Nonlinear Equation Systems, 简称NES)在电子工业设计^{[93][94]}, 工材料^[95], 机器人^{[96][77]}等广泛存在, 为表述方便, 一个非线性系统方程组其形式如下:

$$\begin{cases} e_1(\vec{x}) = 0 \\ e_2(\vec{x}) = 0 \\ \vdots \\ e_M(\vec{x}) = 0 \end{cases} \quad (2-16)$$

$\vec{x} = (x_1, \dots, x_D) \in S$ 为决策向量, S 为决策向量空间以约束的形式表述: $L_i \leq x_i \leq U_i$,

$1 \leq i \leq D$, L_i 和 U_i 是决策向量 x_i 的上界和下界, 而 $e_i(\vec{x}) (i \in \{1, \dots, M\})$ 代表第 i 个方程, M 是方程的数目。一般情况下方程组的数目不止 1 个, 如果 $\forall i \in \{1, K, M\}$, 则 $e_i(\vec{x}^*) = 0$, \vec{x}^* 是 NES 的一个解。

许多经典的数学方法如牛顿法^{[56][57][58]}等被用于求解非线性方程, 但是这些方法^{[60][61]}需要梯度, 可微可导, 严重依赖于初始点的选择, 并且容易陷入到局部最优点。另外这些方法, 只是单个解的迭代, 难以得到多个最优解, 进化算法的群体操作使得多解的非线性方程成为可能。对于非线性方程组的进化求解技术, 国际上已有一些成果, 一般情况下需要将非线性方程组转换位一个优化问题, 在此优化问题上进一步设计算法求解。下面简单介绍相关的工作:

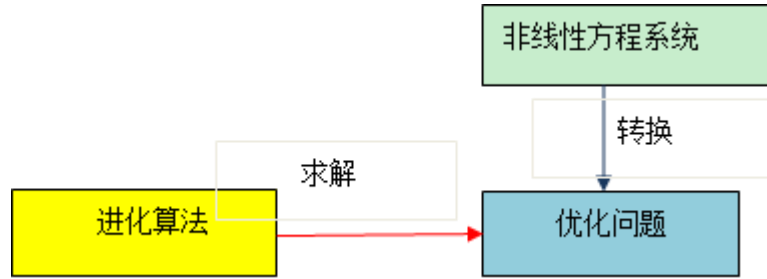


图 2-3 非线性方程系统的求解模型

2.4.1 基于单目标多模态优化方法

大多数研究是将各个方程的绝对值或者平方相加, 组成了一个全局优化的模型, 当非线性方程组具有多个解时, 就成为了多模态优化模型, 在此方法下, 非线性系统方程被转换为如下形式:

$$\begin{aligned} & \text{minimize } \sum_{i=1}^M |e_i(\vec{x})| \\ & \text{minimize } \sum_{i=1}^M e_i^2(\vec{x}) \end{aligned} \quad (2-17)$$

在此转换方法下, Wu^[98]展示了一个基于子空间的精英 GA, Mhetre^[99]等提出了使用 GA 能够有效的解决非线性系统方程的问题, Noriega^[100]应用演化策略计算所有元素的导向问题其可归结为一个 NES 问题, Wang^[101]引入了免疫遗传算法, Xie^[102]组合了使用社会情感算法组合 Metropolis 规则进行优化, 最近粒子群算法吸引了很多注意, 其中协同链接导向方法与粒子群算法求解非线性方程问题, Jaberipour^[108]改进了位置和更新速度, Voglis^[109]建议了微量的损失信息进行求解。

值得注意的是, 这些方法往往只是集中于求解非线性系统方程的一个解, 并没有加入分布机制, 为了定位非线性系统方程的多个解, Brits^[110]等采取了标准的 PSO 协

同小生境策略求解多个解的非线性系统方程问题。Hirsch^[111]等利用连续的随机启发式方法最小化每个方程的解，算法不断的重启去优化改进的目标函数，另外设计了一个排他域机制避免算法重新搜索已经发现到的最优解。Pourjafari^[112]提出了两阶段根发现者方法，这个方法采用杂草算法作为搜索引擎，另外也使用了排他域的方法求解 NES 问题。

尽管以上的方法拥有发现非线性系统方程多个解的能力，但是这些方法依赖于问题的参数，另外还需要一些先验知识，比如小生境的个数以及聚类半径等。

2.4.2 基于约束的优化方法

Kuri-Morales^[113]将一个 NES 转换为一个如下的约束优化问题：

$$\begin{cases} \text{minimize } \sum_{i=1}^M |e_i(\mathbf{x})| \\ \text{subject to } e_i(\mathbf{x}) \geq 0, i=1, \dots, M \end{cases} \quad (2-18)$$

在此优化问题中，最小化每个方程的绝对值之和，其约束是每个方程都需要大于 0。完成以上的转换以后，设计了一个罚函数策略和一个简单的遗传算法去搜索此约束优化问题的最优值。与以上的转换不同的是，Pourrajabian^[114]等将 NES 的每个方程转换为等式约束，其形式如下：

$$\begin{aligned} & \text{minimize } \sum_{i=1}^M |e_i(\mathbf{x})| \\ & \text{subject to } e_i(\mathbf{x}) = 0, i=1, \dots, M \end{aligned} \quad (2-19)$$

转换后，设计了一个拉格朗日罚函数策略和遗传算法协同求解非线性方程组。类似于非线性系统方程组的单目标转换，此方法因为没有加入额外的机制，并不能保证获得非线性系统方程组的多个解。

2.4.3 多目标优化方法

除了以上两种方法外，Grosans 和 Abraham^[115]提出了一个与以上工作完全不同的方法，将非线性方程系统转换为一个多目标优化问题，其转换的方式是将每一个方程看成一个目标。其转换形式如下：

$$\begin{cases} \text{minimize } |e_1(\mathbf{x})| \\ \text{minimize } |e_2(\mathbf{x})| \\ \vdots \\ \text{minimize } |e_M(\mathbf{x})| \end{cases} \quad (2-20)$$

非线性方程组转换为一个 M 个目标的多目标优化问题，其方程组的数目为多目标

优化问题的目标数目，CA 方法能够有效的处理非线性方程组，由于此方法是惟一的一个使用多目标进化算法求解非线性方程系统的方法，在第三章将会详细讨论和实验对比本文所提出的多目标转换方法的异同。

2.4.4 非线性方程与多模态优化问题

非线性方程与多模态优化问题并不是同一类问题，由于非线性方程具有多解性，与多模态优化问题的求解相一致，可以将其视为一个特殊的多模态优化问题，但是还是有所不同，其最优值具有先验信息，多解的非线性方程转换的多模态优化问题的最优值为已知的，其值为 0；而普通的多模态优化问题的值通常是不可得的。

第三章 使用多目标进化算法定位非线性方程组的多个解

非线性系统方程组 (Nonlinear Equation Systems, 简称 NES) 其形式如下:

$$\begin{cases} e_1(\vec{x}) = C \\ e_2(\vec{x}) = C \\ \vdots \\ e_M(\vec{x}) = C \end{cases} \quad (3-1)$$

$\vec{x} = (x_1, \dots, x_D) \in S$ 为决策向量, S 为决策向量空间以约束的形式表述: $L_i \leq x_i \leq U_i$, $1 \leq i \leq D$, L_i 和 U_i 是决策向量 x_i 的上界和下界, 而 $e_i(\vec{x}) (i \in \{1, \dots, M\})$ 代表第 i 个方程, M 是方程的数目。一般情况下方程组的数目不止 1 个, 如果 $\forall i \in \{1, \dots, M\}$, 则 $e_i(\vec{x}^*) = 0$, \vec{x}^* 是 NES 的一个解。

在使用进行算法进行求解时, 一般情况下需要将非线性系统方程组转换为各种不同的优化模型并加以优化, 其中一种方法将其转换为多模态优化模型。此多模态优化问题有着明显的特征, 其转换过来的多模态优化问题的最优值已知的, 因为必须满足每个方程, 因此其值为 0, 而普通的多模态优化问题的最优值未知, 本章使用多目标进化算法求解多解的非线性方程组的动机是: 非线性方程组的多模态转换模型的最优值带有一定的先验信息, 是非常好研究多模态优化问题的多目标优化方法的优化问题, 其次运行多目标优化方法的最终结果就是求取一组相互均衡的而不是一个解, 可以对数量无限的多目标优化问题的 Pareto 进行良好近似, 而非线性方程组还带有自身的特性, 一些特殊的非线性方程组其解的数量是无限的, 可以很自然的使用多目标优化方法求解非线性方程组。

本章将对非线性方程组的多模态优化模型进行探讨, 提出了一种多目标的转换方法, 将非线性方程组转换为一个双目标优化问题。提出了使用多目标进化算法定位非线性方程组的多个解 (Locating Multiple Optimal Solutions of Nonlinear Equation Systems Based on Multiobjective Optimization, 简称为 MONES), MONES 使用多目标转换函数将非线性方程转换为一个双目标的优化问题, 其解在转换而来的多目标优化问题的前沿面上, 并使用多目标进化算法进行求解。此种方法具有如下特征:

1 MONES 无需先验知识包括解的个数、非线性方程组的数目等可以求得非线性方程组的多个解。

2 具有多个解的非线性方程组的所有解都在所转换过来的多目标优化问题的前沿面上。

3 其前沿面的形状是已知的, 在其转换过来的多目标优化问题的目标空间上为线段“ $y=1-x$ ”。

4 由于多个解在其前沿面上，因此可以直接使用多目标进化算法的相关方法进行定位。

5 MONES 能够一次运行近似无限个解的非线性方程组。

本节的内容组织如下：3.2 节介绍本文所提出的算法，并使用了实例与已存在的多目标进化算法求解非线性方程系统进行分析，3.3 节为了证明本文提出算法的有效性，将详细给出实验的对比，最后得出结论。

3.1 多目标优化方法相关工作

使用多目标进化算法求解非线性方程系统的工作较少，Crina Grosans 和 Abraham^[80]（为了表述方便简称为 CA 方法）第一个提出了使用多目标进化算法求解非线性方程系统，其工作非常简单和直观，将非线性方程系统转换为一个多目标优化问题，其转换的方式将每一个方程看成多目标优化问题的一个目标，然后使用多目标进化算法进行求解，其转换形式如下：

$$\begin{cases} \text{minimize } |e_1(\mathbf{x})| \\ \text{minimize } |e_2(\mathbf{x})| \\ \quad \quad \quad \mathbf{M} \\ \text{minimize } |e_M(\mathbf{x})| \end{cases} \quad (3-2)$$

以此方式非线性方程组转换为一个 M 个目标的多目标优化问题，其方程组的数目为多目标优化问题的目标数目，根据他们的实验结果，CA 方法能够有效的处理具有多个解的非线性方程系统问题，其求解效率超出了 Effati^[112]的方法和其它的一些经典的数学方法的性能。

注意到 CA 方法将每个方程都作为一个目标，在本文第二章相关工作中指出，当多目标优化问题的目标数目大于 6 个目标时（极大目标），非常难以收敛，极大目标下的多目标进化优化对于多目标优化社区本来就是一个非常巨大的挑战^{[87] [88] [89]}。因此可以简单的得出当方程数目增长时，CA 方法趋近于无效。另外可以看到，当将每个方程都作为一个目标，而非线性方程系统的最优值由于满足每个方程，因此以公式 3-2 的转换方式，每个方程也为 0，也就是说所有的目标也都是 0，此 0 点或者说极大目标优化问题的理想点是存在的，并且所有的解都对应于 1 个点，在此转换目标的前沿面，并不存在着冲突。由于此方法是第一个将多目标进化算法引入到非线性方程组的求解，本节将在实验部分详细对比本文所提出的多目标转换方法，下面将介绍本文所提出的双目标转换方法。

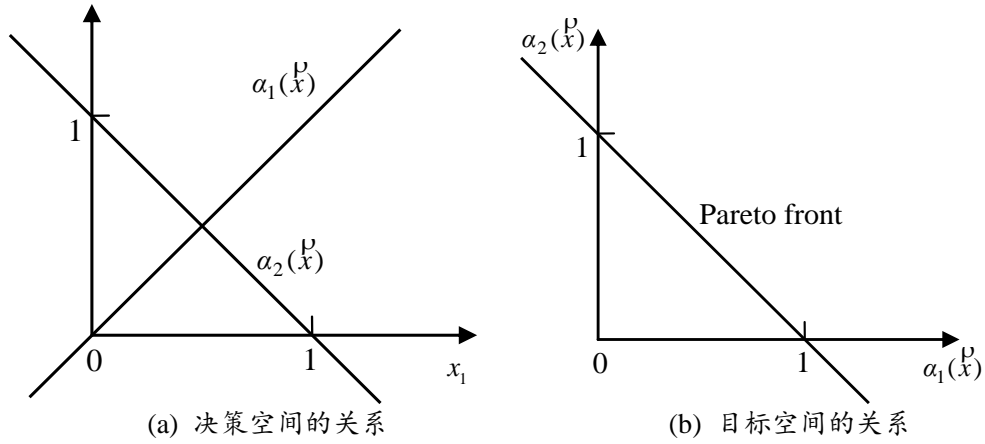


图 3-1 定位函数的关系示意图

3.2 基于多目标优化算法定位非线性方程组的多个解

本文的动机是将多模态优化问题的最优值映射到多目标优化问题的前沿面上，从而达到其最优值在 Pareto 支配概念上无法比较的目的，并使用多目标进化算法进行求解，在此过程中达到对于多解的非线性方程组这一特殊多模态优化问题进行深入研究的目的，探讨其求解性质。本文所提出的多目标的转换模型由两部分组成，一部分称之为系统函数，另一个为定位函数，然后提出一个实例展示本文所提出的方法和 CA 方法的不同，最后分析了不同形状的定位函数的性质。

3.2.1 多目标转换函数

在本小节中将介绍本文所提出的双目标转换模型，将非线性方程组转换为一个双目标优化问题，此双目标优化问题由系统函数和定位函数组成，并给出非线性方程组的所有解都在其多目标优化问题的前沿面上。一个定位函数 $\alpha(\vec{x})$ 具有如下形式：

$$\begin{cases} \text{minimize } \alpha_1(\vec{x}) = x_1 \\ \text{minimize } \alpha_2(\vec{x}) = 1 - x_1 \end{cases} \quad (3-3)$$

$\vec{x} = (x_1, \dots, x_D) \in S$ 为决策向量， x_1 为向量 \vec{x} 的一个决策向量。

定理 3.1 任意的向量 \vec{a} 和 \vec{b} ， $\vec{a}, \vec{b} \in S$ ，必然有 \vec{a} 和 \vec{b} 互不支配。

证明：假设 $\vec{a} \prec \vec{b}$ ，那么必然有 $\alpha_1(\vec{a}) < \alpha_1(\vec{b})$ 和 $\alpha_2(\vec{a}) < \alpha_2(\vec{b})$ ，根据定义 $\alpha_2(\vec{a}) = 1 - \alpha_1(\vec{a})$ 和 $\alpha_1(\vec{b}) = 1 - \alpha_2(\vec{b})$ 。存在 $1 - \alpha_1(\vec{a}) < 1 - \alpha_2(\vec{b})$ ，有 $\alpha_1(\vec{a}) > \alpha_2(\vec{b})$ 。矛盾，因此两者之间不存在着支配关系。

此定理也可以从函数图 2-2 看出， $\alpha_1(\vec{x})$ 单调递增，而 $\alpha_2(\vec{x})$ 单调递减，两者在变量空间中处处矛盾。从图中也可以看出 $\alpha(\vec{x})$ 在目标空间中的解析式为线性函数“ $y=1-x$ ”。

一个系统函数 $\beta(\vec{x})$ 具有如下形式：

$$\begin{cases} \text{minimize } \beta_1(\vec{x}) = \sum_{i=1}^M |e_i(\vec{x})| \\ \text{minimize } \beta_2(\vec{x}) = M * \max(|e_1(\vec{x})|, \dots, |e_M(\vec{x})|) \end{cases} \quad (3-4)$$

$\vec{x} = (x_1, \dots, x_D) \in S$ 为决策向量, $e_1(\vec{x}), \dots, e_M(\vec{x})$ 是一个非线性系统方程组的 M 个方程。

定理 3.2 如果一个向量 \vec{x} 为非线性系统方程组的一个解, 必然有 $\beta_1(\vec{x}) = \beta_2(\vec{x}) = 0$ 。

证明: 如果 \vec{x} 是非线性系统方程组的解, 有 $\forall e_i(\vec{x}) = 0, i \in \{1, \dots, M\}$, 从而得到 $\beta_1(\vec{x}) = \beta_2(\vec{x}) = 0$ 。

需要指出的是在系统函数中, $\beta_1(\vec{x})$ 所有方程组的绝对值之和, 而 $\beta_2(\vec{x})$ 是最大的一个方程的 M 倍。当非线性方程组存在着多个解时, $\beta_1(\vec{x})$ 和 $\beta_2(\vec{x})$ 为一个最优值为 0 的多模态函数优化问题。

组合定位函数和系统函数成为多目标转换函数, 对于一个非线性系统方程组, 其形式如下:

$$\begin{cases} \text{minimize } f_1(\vec{x}) = \alpha_1(\vec{x}) + \beta_1(\vec{x}) \\ \quad \quad \quad = x_1 + \sum_{i=1}^M |e_i(\vec{x})| \\ \text{minimize } f_2(\vec{x}) = \alpha_2(\vec{x}) + \beta_2(\vec{x}) \\ \quad \quad \quad = 1 - x_1 + M * \max(|e_1(\vec{x})|, \dots, |e_M(\vec{x})|) \end{cases} \quad (3-5)$$

下面将给出证明, 非线性方程组的所有解都在公式 3-10 函数的前沿面上。

定理 3.3 所有的非线性系统方程组的解在其多目标转换函数的前沿面上。

证明: 假设 \vec{x}^* 为一个非线性方程组的解, 而 \vec{x}' 为一个非线性方程组的任意决策变量, 有 $\vec{x}' \neq \vec{x}^*$ 。由于 \vec{x}^* 为方程组的一个解, 根据定理 2.2 必然 $\beta_1(\vec{x}^*) = \beta_2(\vec{x}^*) = 0$, 那么 $f_1(\vec{x}^*) = \alpha_1(\vec{x}^*)$ 和 $f_2(\vec{x}^*) = \alpha_2(\vec{x}^*)$, 而 \vec{x}' 有 $\beta_1(\vec{x}'), \beta_2(\vec{x}') \geq 0$, $f_1(\vec{x}') = \alpha_1(\vec{x}') + \beta_1(\vec{x}')$ 和 $f_2(\vec{x}') = \alpha_2(\vec{x}') + \beta_2(\vec{x}')$, 根据 $f_1(\vec{x}') \leq f_1(\vec{x}^*)$ 重写有 $\alpha_1(\vec{x}') + \beta_1(\vec{x}') \leq \alpha_1(\vec{x}^*)$, 存在 $\alpha_1(\vec{x}') - \alpha_1(\vec{x}^*) \leq -\beta_1(\vec{x}')$, 推出 $\alpha_1(\vec{x}') - \alpha_1(\vec{x}^*) \leq 0$ 而根据 $f_2(\vec{x}') \leq f_2(\vec{x}^*)$ 重写有 $\alpha_2(\vec{x}') + \beta_2(\vec{x}') \leq \alpha_2(\vec{x}^*)$, 存在 $\alpha_2(\vec{x}') - \alpha_2(\vec{x}^*) \leq -\beta_2(\vec{x}')$, 推出 $\alpha_2(\vec{x}') - \alpha_2(\vec{x}^*) \leq 0$, 得到 \vec{x}^* 在 $\alpha(\vec{x})$ 上都要小于 \vec{x}' , 与定义相矛盾。故不存在个体在此多目标优化问题上支配非线性方程组的解, 从而所有的非线性系统方程组的解在其多目标转换函数的前沿面上。

下面将进一步的指出非线性方程组的解特征。

定理 3.4 所有的非线性系统方程组的解在多目标转换问题的目标空间“ $y=1-x$ ”上。

证明: 假设 \vec{x}^* 为一个非线性方程组的解, 根据以上定理 $\beta_1(\vec{x}^*) = \beta_2(\vec{x}^*) = 0$, \vec{x}^* 为此双目标优化问题的 Pareto 最优解, $\alpha_1(\vec{x}^*) = x_1^*$ 和 $\alpha_2(\vec{x}^*) = 1 - x_1^*$, 必为 $y=1-x$ 。得证

由上一个多解的非线性方程系统的解都转换到了所构造的多目标优化问题的前沿面上, 但是需要指出的是并不代表非线性方程系统已经得到解决, 其多目标优化问题

的前沿面所对应的变量空间并不清楚。下面将给出一个实例用以分析 CA 方法与本文的方法的区别。

3.2.2 多目标转换函数的一个实例

由于 CA 方法也使用了多目标进化算法，为了更好的分析 MONES 的特征，下面将给出一个具体的多解的非线性方程组的实例：

$$\begin{cases} e_1(\mathbf{x}) = x_1^2 + x_2^2 - 1 = 0 \\ e_2(\mathbf{x}) = x_1 - x_2 = 0 \\ -1 \leq x_1, x_2 \leq 1 \end{cases} \quad (3-6)$$

在此非线性系统方程中，第一个方程为一个单位圆的解析式，而第二个方程为一个线段，方程组具有两个解 $\mathbf{x}_1^* = (0.707, 0.707)$ 和 $\mathbf{x}_2^* = (-0.707, -0.707)$ 。

根据 CA 方法其转换过来的解析式为：

$$\begin{cases} \text{minimize } f_1(\mathbf{x}) = |x_1^2 + x_2^2 - 1| \\ \text{minimize } f_2(\mathbf{x}) = |x_1 - x_2| \end{cases} \quad (3-7)$$

相反与 CA 方法，本文的方法其转换过来的解析式为：

$$\begin{cases} \text{minimize } f_1(\mathbf{x}) = x_1 + |x_1^2 + x_2^2 - 1| + |x_1 - x_2| \\ \text{minimize } f_2(\mathbf{x}) = 1 - x_1 + 2 * \max(|x_1^2 + x_2^2 - 1|, |x_1 - x_2|) \end{cases} \quad (3-8)$$

在其变量空间中，均匀采样 441 个点，如图所示。这些均匀采样的点按照 CA 方法和 MONES 方法转换过来以后，其在目标空间中如图所示。红色标记的为最优点。可以看到 CA 方法 2 个最优值相互重叠，而随着进化的进行，重叠的区域将被无理由的替代掉，在而本文的方法其最优值在目标空间都不相同，并且都不被其它的点所支配，有理由相信本文所提出的方法有着更好的适应性，在此有待实验进一步验证。

为了更进一步的分析其支配结构，在决策空间中采样 6 个点，分别为 A, B, C, D, E, F 其变量空间位置和转换的多目标优化问题的目标空间值如表 3-1 所示，其变量空间分布如图 3-3 所示，从图中可以看出 B 点和 E 点更靠近最优值。在 CA 方法中，B 和 D 点由于不被任何的其它点所支配，因此为 Pareto 最优值，而本文所提出的方法，D 点的适应值在支配关系的层面上是最差的，B 点和 E 点为 Pareto 最优值，更靠近理论上的最优值，从而保存了对潜在最优值区域搜索的可能性。从此实例上可以看出本文所提出的方法更有合理性。

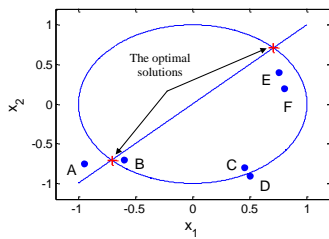
1) 首先 MONES 是将此非线性方程组先转换为两个多模态函数优化问题，再将其嵌入到多目标转换模型中，其目标数目是 2。而 CA 方法将非线性方程组转换为 M 个目标的优化问题，当方程数目很多的时候就将遇到维数灾难的问题 (curse of dimensionality)。

2) 在本节所提出的实例上可以得到：非线性方程组的所有解在 CA 方法所转换过来的目标空间中是重叠的。MONES 为了将多个最优值映射到一个多目标优化问题的前沿面上，每个最优值是不相同的。

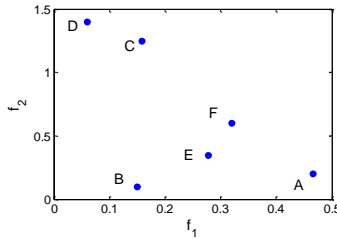
3) 由于转换方式不同，其搜索的行为也是不一样的。首先由于其前沿面不是一个点，种群趋向于前沿面的搜索，而 CA 方法由于转换之后，Pareto 前沿面有一个点，种群在早期由于多个目标评价的影响趋向于搜索整个解空间，随着进化的进行，较优的个体被种群所搜索到，趋向于搜索较优的个体。

表 3-1 实例个体的信息

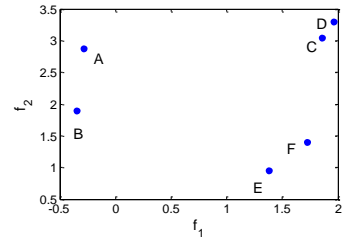
个体	位置		目标函数 (CA)		目标函数 (MONES)	
	x_1	x_2	f_1	f_2	f_1	f_2
A	-0.95	-0.75	0.4650	0.2000	-0.285	2.8800
B	-0.60	-0.70	0.1500	0.1000	-0.350	1.9000
C	0.45	-0.80	0.1575	1.2500	1.8575	3.0500
D	0.50	-0.90	0.0600	1.4000	1.9600	3.3000
E	0.75	0.40	0.2775	0.3500	1.3775	0.9500
F	0.80	0.20	0.3200	0.6000	1.7200	1.4000



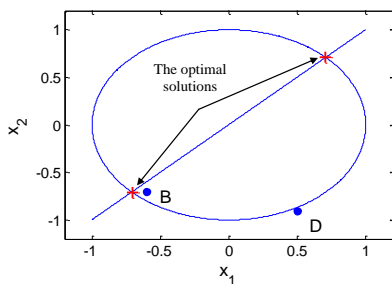
(a) 决策空间的分布



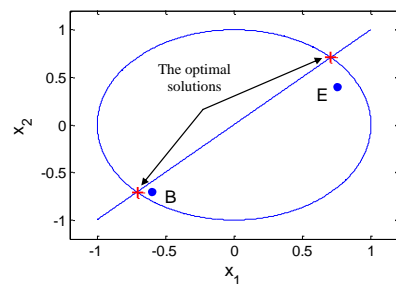
(b) CA 方法的目标空间



(c) MONES 方法的目标空间



(d) CA 中的非支配个体



(e) MONES 中的非支配个体.

图 3-2 CA 和 MONES 的多目标转换方式的差别.

3.2.3 不同类型的定位函数的影响

在本文所提到的转换方法中，前沿面的形状是由定位函数 $\alpha(\vec{x})$ 确定，在前文所讨论的 $\alpha(\vec{x})$ 为线性函数，而对于不同形状的定位函数是否更有效，是本小节所要解决的问题。

题, 定义一个定位函数 $\alpha(\vec{x})$ 如下:

$$\begin{cases} \text{minimize } \alpha_1(\vec{x}) = x_1 \\ \text{minimize } \alpha_2(\vec{x}) = 1 - x_1^\gamma \end{cases} \quad (3-9)$$

在方程中如果 $x_1 \geq 0$, $\alpha_1(\vec{x})$ 是严格的递增函数, $\alpha_2(\vec{x})$ 是严格的递减函数, 那么其在变量空间中处处冲突, 对于变量空间中的任意个体都存在互不支配的关系。如果 $0 < \gamma \leq 1$, the Pareto 前沿面是凸的, 如 $\gamma > 1$, Pareto 前沿面为凹的, 图 3-5 (b) 展示了 5 个不同的 α 值, 当 $0 \leq x_1 \leq 1$. 需要注意的是 x_i ($i \in \{1, \dots, D\}$) 并不常常在实际问题中, 比如本文的测试实例都是在 $x_i \in [-1, 1]$ 的区间内如果 $-1 \leq x_1 < 0$, 根据 γ 的不同将会发生以下几种情况:

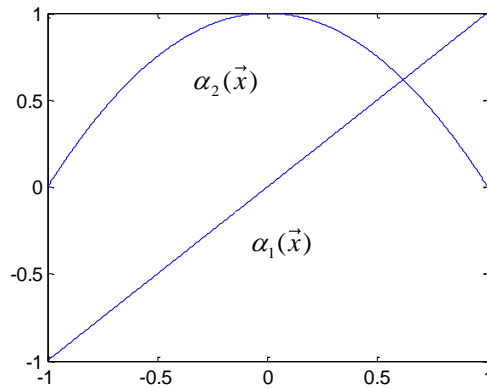


图 3-3 定位函数在目标空间的关系 (纵坐标为函数值, 横坐标为决策变量值)

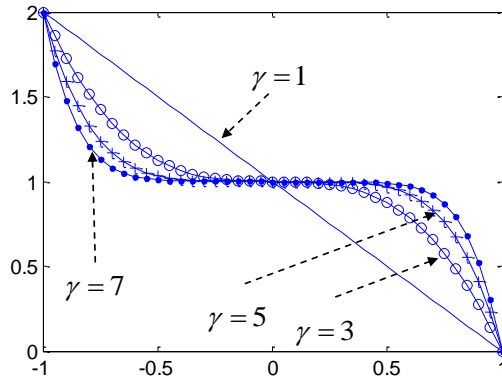


图 3-4 Pareto 前沿的不同形状 (纵坐标为函数值, 横坐标为决策变量值)

1) γ 是小数(如 $1/2$ 和 $17/5$): x_1^γ 则为复数(如 $(-0.5)^{1/2}$ $(-0.8)^{17/5}$) 对于复数构成的多目标前沿面的形状是无法知道的, 无疑会增加求解难度。

2) γ 正的偶数(2, 4, 6): $\alpha_1(\vec{x})$ 与 $\alpha_2(\vec{x})$ 将在某些区域不发生冲突, 可以从图看出 ($\gamma=2$) 当 $-1 \leq x_1 < 0$, $\alpha_1(\vec{x})$ 和 $\alpha_2(\vec{x})$ 是严格单调递增的而在此之外的区间并不冲突, 两个函数协调发展。违反了本文的定理 1, 从而造成求解困难。

3) γ 正的奇数 (1, 3, 5): $\alpha_1(\vec{x})$ 与 $\alpha_2(\vec{x})$ 相冲突。

从以上的讨论可以看出当时, γ 为正的奇数, 符合本文所提出的构造函数定义,

在图 3-5 中给出了 $\gamma=1, 3, 5$, 不同的前沿面的形状, 可以看到前沿面的某些区域几乎是水平和垂直状态, 对于多目标优化算法而言较难以搜索到, 因此得到具有线性形状的 $\alpha(\vec{x})$ 更具备合理性, 我们将在实验中将给出充分的实验论证。

3.3 求解非线性方程组系统的算法流程

为了显示本文所提出的方法的普适性, 将算法嵌入到经典的 NSGA-II 中的框架中, 其算法如下:

求解非线性方程系统多个解的多目标进化算法

输入: 多解的非线性方程组
 输出: 非线性方程组的多个解

$G=0$; // G is the generation number
 Randomly generate an initial population P_G of size N from the decision space;
 Evaluate each individual in P_G based on equation 3-5;
 Implement the simulated binary crossover and polynomial mutation to generate the offspring population Q_G ;
 Evaluate each individual in Q_G based on equation 3-5;
 $H_G = P_G \cup Q_G$;
 Pareto_soring: Divide H_G into k different nondomination levels ND_1, ND_2, \dots, ND_k according to a fast nondominated sorting;
 $P_{G+1} = \emptyset$;
 $i=1$;
 While $|P_{G+1}| < N$
 $P_{G+1} = P_{G+1} \cup ND_i$;
 $i = i + 1$;
 End While
 $P_{G+1} = P_{G+1} \setminus ND_{i-1}$;
 Delete $(|ND_{i-1}| + |P_{G+1}| - N)$ individuals with the smallest crowding-distance values in ND_{i-1} ;
 $P_{G+1} = P_{G+1} \cup ND_{i-1}$;
 $G = G + 1$;
 If the stopping criterion is satisfied, stop and output the final population, otherwise go to step 4.

其中 Pareto_soring 为非支配排序算法, 其作用是将种群中的个体分为不同的层数, 算法如下:

非支配排序算法

输入：种群 Pop

输出：种群 K 个层次 ND_1, K, ND_k

ParetoSort(Pop)

for each $p \in \text{Pop}$ //第一部分：计算 n_i 和 s_i

for each $q \in \text{Pop}$

if (p dominated q) then

$sp = sp \cup \{q\}$

else if (q dominated p) then

$np = np + 1;$

end for q

if ($np=0$) then

$P1 = P1 \cup \{p\}$

end for p

$i=1;$

while ($P_i \neq \emptyset$) //第二部分：求 ND_1, K, ND_k

$H = \emptyset;$

for each $p \in P_i$

for each $q \in sp$

$nq = nq - 1;$

if ($nq=0$) then $H = H \cup \{q\}$

end for p

$i = i + 1;$

$P_i = H;$

end for while

end for sort

在算法的主流程中，与第二代的非支配排序多目标进化算法（NSAG-II）是相同的，其分布度的保持策略使用了拥挤距离的概念。由于拥挤距离计算速度较快且被多目标进化算法采用较多，故本文算法仍然采用拥挤距离保持分布，下面将简单介绍拥挤距离的定义。

存在子目标 f_1 和 f_2 的双目标优化问题，个体 i 的聚集距离是图中虚线四边形的长与宽之和。设 $P[i].\text{distance}$ 为个体 i 的聚集距离， $P[i].f_j$ 为个体 i 在子目标 j 上的函数值，则图 4-6 中个体 i 的聚集距离为：

$$P[i].distance = (P[i+1].f_1 - P[i-1].f_1) + (P[i+1].f_2 - P[i-1].f_2) \quad (3-10)$$

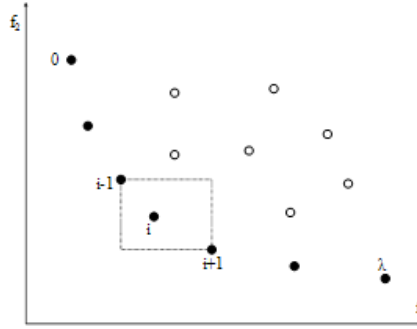


图 3-5 个体间的聚集距离

当有 r 个子目标时个体 i 的聚集距离为:

$$P[i].distance = \sum_{k=1}^r (P[i+1].f_k - P[i-1].f_k) \quad (3-11)$$

3.4 实验研究

3.4.1 测试平台的设计

由于非线性方程组并没有一个统一的具有不同函数性质的比较平台，在本文中 7 个具有不同特征的测试函数被用来构造实验平台，用来研究进化算法求解多解的非线性方程组的有效性。7 个测试函数中，前面 5 个根据不同的特性和维数构造出来，最后两个是来自于神经科学的一个实际的测试实例。这些测试函数具有不同的特征。

1) F1 和 F2 具有 2 个解。其中 F1 是本节中所提出来的实例。F2 包括 10 个决策变量，主要是用来考察在高维空间中的有效性。从本质上来说，F1 是 F2 一种泛化，在 x_1-x_2 空间中具有相似的结构。

2) F3 和 F4 的解多于 2 个。其中 F3 有 11 个解，而 F4 有 15 个解。在这两个测试函数上，有的解在变量空间中非常的相近，导致算法很难进行定位。

3) F5, F6, 和 F7 拥有无限的解，使用进化算法不可能取得这些非线性方程系统的所有解集，因此只能取得一个近似集合。另外 F6 被用来展示 CA 的限制，它包含了 6 个方程，因此按照 CA 的方法具有 6 个组的所有解，因此只能在一次运行中找到一个近似的解集去表达数目无限的解集。

3.4.2 性能指标

在非线性方程组中，对于实验的评价与多模态优化问题有所不同，由于存在着解

表 3-2 非线性方程系统的测试函数集

	测试函数	决策变量 个数	决策 空间	线性方程 个数	非线性 方程个数	解的个数
F1	$x_1^2 + x_2^2 - 1 = 0$ $x_1 - x_2 = 0$	2	$[-1,1]^2$	1	1	2
F2	$\sum_{i=1}^D x_i^2 - 1 = 0$ $ x_1 - x_2 + \sum_{i=3}^D x_i^2 = 0$	20	$[-1,1]^{20}$	0	2	2
F3	$x_1 - \sin(5 * \pi * x_2) = 0$ $x_1 - x_2 = 0$	2	$[-1,1]^2$	1	1	11
F4	$x_1 - \cos(4 * \pi * x_2) = 0$ $x_1^2 + x_2^2 = 1$	2	$[-1,1]^2$	0	2	15
F5	$x_1 + x_2 + x_3 - 1 = 0$ $x_1 - x_2^3 = 0$	3	$[-1,1]^3$	1	1	infinite
F6	$x_1^2 + x_3^2 = 1$ $x_2^2 + x_4^2 = 1$ $x_5 x_3^3 + x_6 x_4^3 = 0$ $x_5 x_1^3 + x_6 x_2^3 = 0$ $x_5 x_1 x_3^2 + x_6 x_4^2 x_2 = 0$ $x_5 x_1^2 x_3 + x_6 x_2^2 x_4 = 0$	6	$[-1,1]^6$	0	6	Infinite
F7	$(x_k + \sum_{i=1}^{D-k-1} x_i x_{i+k}) x_D - c_k = 0, 1 \leq k \leq D-1$ $\sum_{l=1}^{D-1} x_l + 1 = 0$	20	$[-1,1]^{20}$	1	19	Infinite

集无限的问题，在一次进化的过程中无法得到其无限的群体。在本文中，使用了 2 钟指标评价非线性方程组的有效性，第一个指标借用的多目标进化算法的反向世代距离 (inverted generational distance, 简称为 IGD)，其定义如下：设 IP 为种群在目标空间的解集，以及 IP^* 最优解在在目标空间的解集其中 $IP^* = \{\bar{v}_1, K, \bar{v}_{|IP^*|}^1\}$ 。IGD 的大小以如下

方式进行计算：

$$IGD(IP, IP^*) = \frac{\sum_{i=1}^{|IP^*|} d(\bar{v}_i^1, IP)}{|IP^*|} \quad (3-12)$$

$d(\bar{v}_i^1, IP)$ 是 \bar{v}_i^1 和 IP 中的点的最小欧几里得距离，值得指出的是如果解集是无限的 (比如 F5 和 F6)， IP^* 在前沿面分布均匀的解集。在本文中， IP^* 中点的个数设置为 100，

表 3-3 CA 方法与 MONES 方法实验结果对比

Test instance	Status	IGD		NOF	
		CA	MONES	CA	MONES
F1	Best	9.97E-01	1.47E-04	1	2
	Mean	9.99E-01	2.01E-04	1	2
	Worst	1.00E+00	3.77E-04	1	2
	Std Dev	4.09E-04	4.74E-05	0.00E+00	0.00E+00
F2	Best	9.90E-01	2.06E-04	1	2
	Mean	9.99E-01	4.44E-04	1	2
	Worst	1.00E+00	9.25E-04	1	2
	Std Dev	1.98E-03	1.95E-04	0.00E+00	0.00E+00
F3	Best	5.51E-01	1.11E-03	1	11
	Mean	1.16E+00	2.12E-03	1	11
	Worst	1.30E+00	4.45E-03	1	11
	Std Dev	2.26E-01	7.48E-04	0.00E+00	0.00E+00
F4	Best	7.36E-01	2.82E-03	1	15
	Mean	7.83E-01	1.06E-02	1	14.13
	Worst	1.18E+00	2.90E-02	1	11
	Std Dev	1.10E-01	7.52E-03	0.00E+00	1.16E+00
F5	Best	3.14E-01	1.63E-02	1	50
	Mean	6.11E-01	4.24E-02	1	37.29
	Worst	7.07E-01	2.12E-01	1	21
	Std Dev	1.12E-01	3.80E-02	0.00E+00	7.05E+00
F6	Best	6.55E-01	1.28E-02	0	81
	Mean	1.66E+00	2.31E-02	0	73.13
	Worst	3.11E+00	5.13E-02	0	66
	Std Dev	6.03E-01	9.45E-03	0.00E+00	3.81E+00
F7	Best	2.62E-01	4.46E-02	8	38
	Mean	8.41E-01	1.15E-01	1.8	16.07
	Worst	3.45E+00	2.37E-01	0	0
	Std Dev	6.59E-01	6.05E-02	1.81E+00	1.23E+01

为了使比较更为公平，在计算 CA 方法的最终种群 IGD 指标时，将种群中的个体

按照本文所给的方法进行转换。此指标能够衡量 IP 的分布和收敛程度，如果一个 NES 的解的个数有限，并且最终结果的 $IGD(IP, IP^*) = 0$ 。代表着所有的解都已经进行了良好的定位。

2) 最优解个数指标 (Number of the optimal solutions found, 简称为 NOF): 此指标是根据多模态优化问题的 NOF 指标发展而来，其数学形式如下：

$$NOF(IP, IP^*) = \sum_{i=1}^{|IP^*|} flag(\mathbf{v}_i^r) \quad (2-18)$$

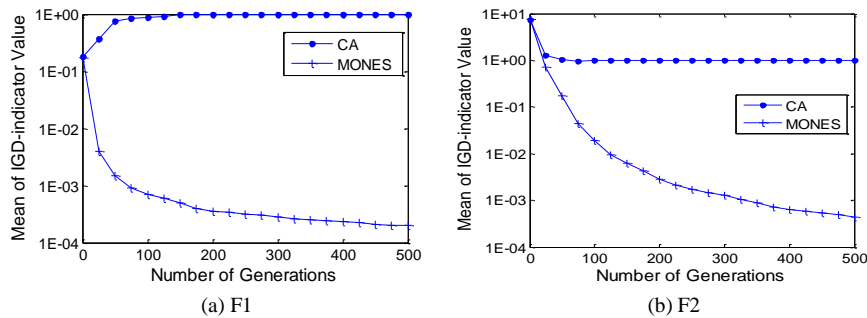
$$\begin{cases} flag(\mathbf{v}_i^r) = 1, & \text{if } d(\mathbf{v}_i^r, IP) \leq \delta, \mathbf{v}_i^r \in IP^* \\ flag(\mathbf{v}_i^r) = 0, & \text{otherwise} \end{cases}$$

δ 是一个阈值, 在本文中 F5 中的 δ 设置为 0.01, 而其他的测试函数设置为 0.02 其意义是只要一个种群中的个体在变量空间中与最优值小于阈值, 就表示此最优值已经被搜索到, 一般情况下 NOF 越大代表着定位的解越多。

3.4.3 与 CA 方法的比较

为了公平比较 CA 方法和本文所提出的方法, 种群的大小都设为 100 个个体, MONES 方法和 CA 方法在 7 个函数上执行 30 次独立的运行, 最大代数设为 500 (相应的最大适应值评价次数为 50000 评价), 采用相同的模拟二进制交叉算子和多项式变异算子。

表 3-5 展示了最好, 最差, 平均、和标准偏差的 IGD-指标值和 NOF 指标值。对于 IGD 指标值, 执行了 2 个方法的统计测试 (0.05 显著性的 Wilcoxon 秩和检验), 其默认的最优值数量为 100。对于这三个测试问题, MONES 能够获得大多数的解, 统计测试表明本文所提出的显著性的优于 CA 方法。在 F1-F3 函数上的求解上, NOF 指标清楚的显示 30 次运行中 MONES 方法 能够成功定位所有的解, 对于 F4 函数而言, MONES 方法在大部分的运行次数上, 能够发现所有的解。由于 F5-F7 函数的 $|IP^*|$ 设置为 100, 而对于 CA 方法而言获取多个解就变得很困难, 特别是在 F1-F5 函数上趋向于找到其中的一个解, 特别是对于 F6 函数, 其根本就无法收敛于最优值附近, 这是因为 F6 函数的方程个数有 6 个, 相应为 6 个目标的优化问题, 因此对于极大目标的优化问题, CA 方法难以收敛到前沿面。



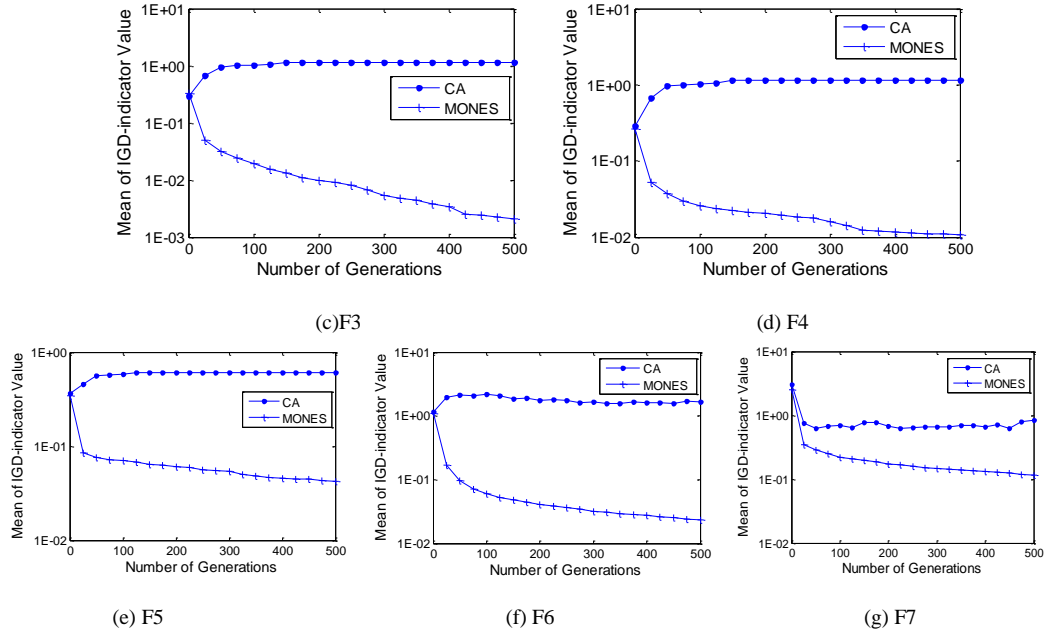


图 3-6 30 次运行平均的 IGD 值进化曲线图

图 2-10 显示了 CA 方法和 MONES 方法 30 次运行中, 进化过程中的平均 IGD 指标值, CA 的 IGD 指标值在 100 代以后陷入停滞, 另外一个有趣的现象是在 F1, F3, 初始种群的个体组成的随机均匀抽取决策空间小 (根据我们的观察 0.1 和 1.1 之间)。在目标空间中, CA 方法趋向于覆盖问题的前沿面, 而对于收敛后的种群 CA 方法趋向于找的一个解, IGD 指标值包括衡量其前沿面的覆盖情况和收敛情况, 因此会出现越来越小的可能性。相反在 F2 和 F7 函数上, IGD 指标值由于决策空间相对来说较大, CA 方法在 IGD 指标值出现越来越大的情况。而 MONES 方法在任何函数上都表现出越来越少的现象, 这说明了算法的收敛能力和寻优能力都要优于 CA 方法。给出 CA 方法和 MONES 方法决策空间的 30 次运行后种群的分布。由于 F2、F6 和 F7 函数决策变量的数量大于 3, 这三个测试函数的分布信息无法在二维决策空间显示。In Figs. 6 and 8, 我们可以观察到一些不同的最优解决能够被 CA 方法所定位, 但是由于是 30 个试验都整合到了一幅图中, 在一次实验中它可能会发现一个最优值。尽管如此, Figs. 6 and 8 即使使用不同的初始种群, CA 方法并不能够保证找到所有的最优值。例如, 相对于 F3 和 F4 函数, 有些最优值仍然无法被 CA 独立运行 30 次的实验所发现。

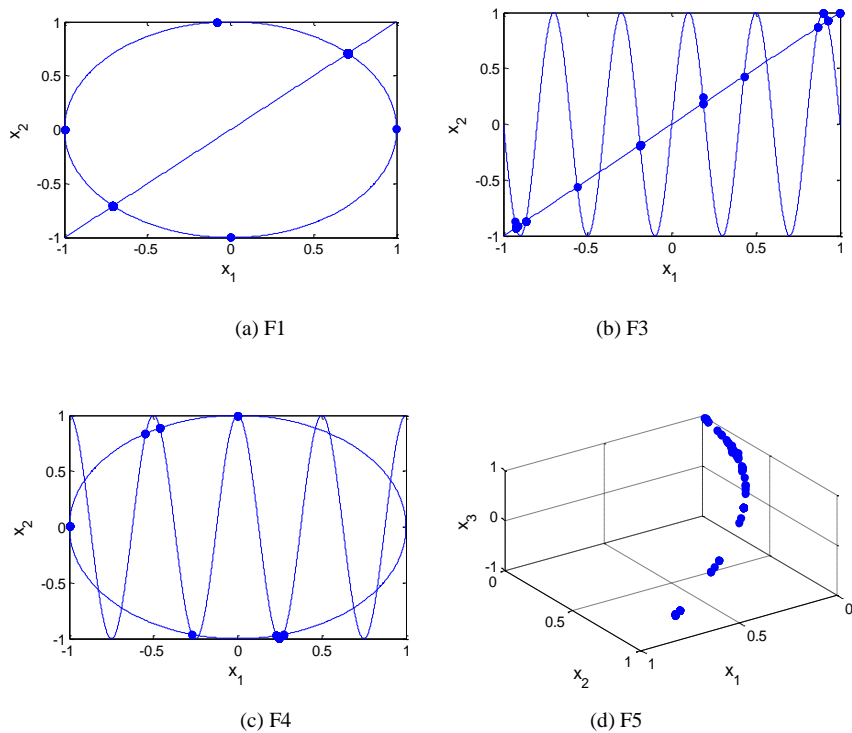


图 3-7 CA 方法 30 次运行目标空间最终结果分布

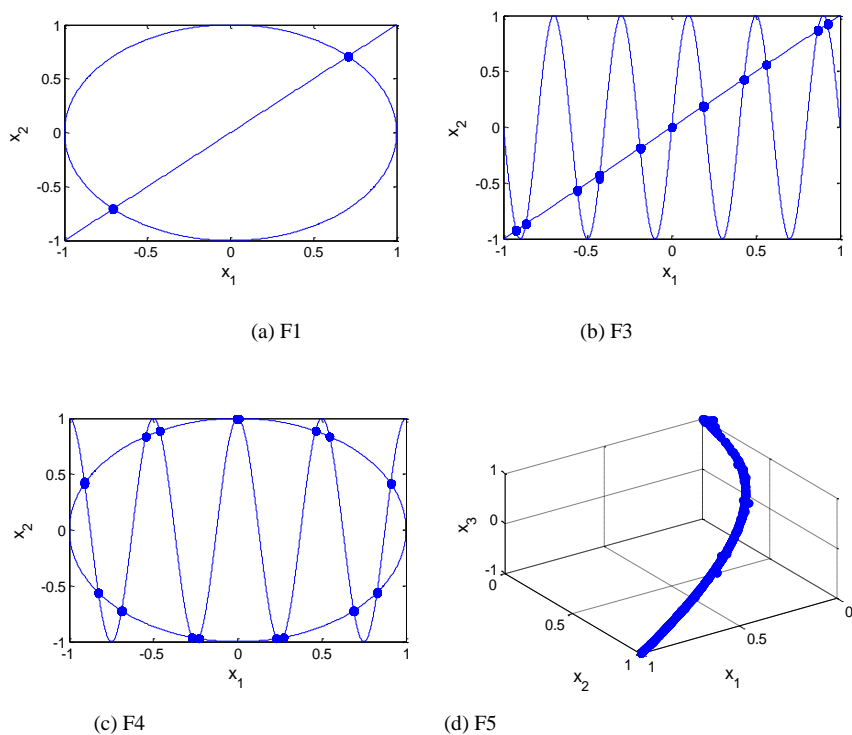


图 3-8 MONES 方法 30 次运行目标空间中最终结果分布

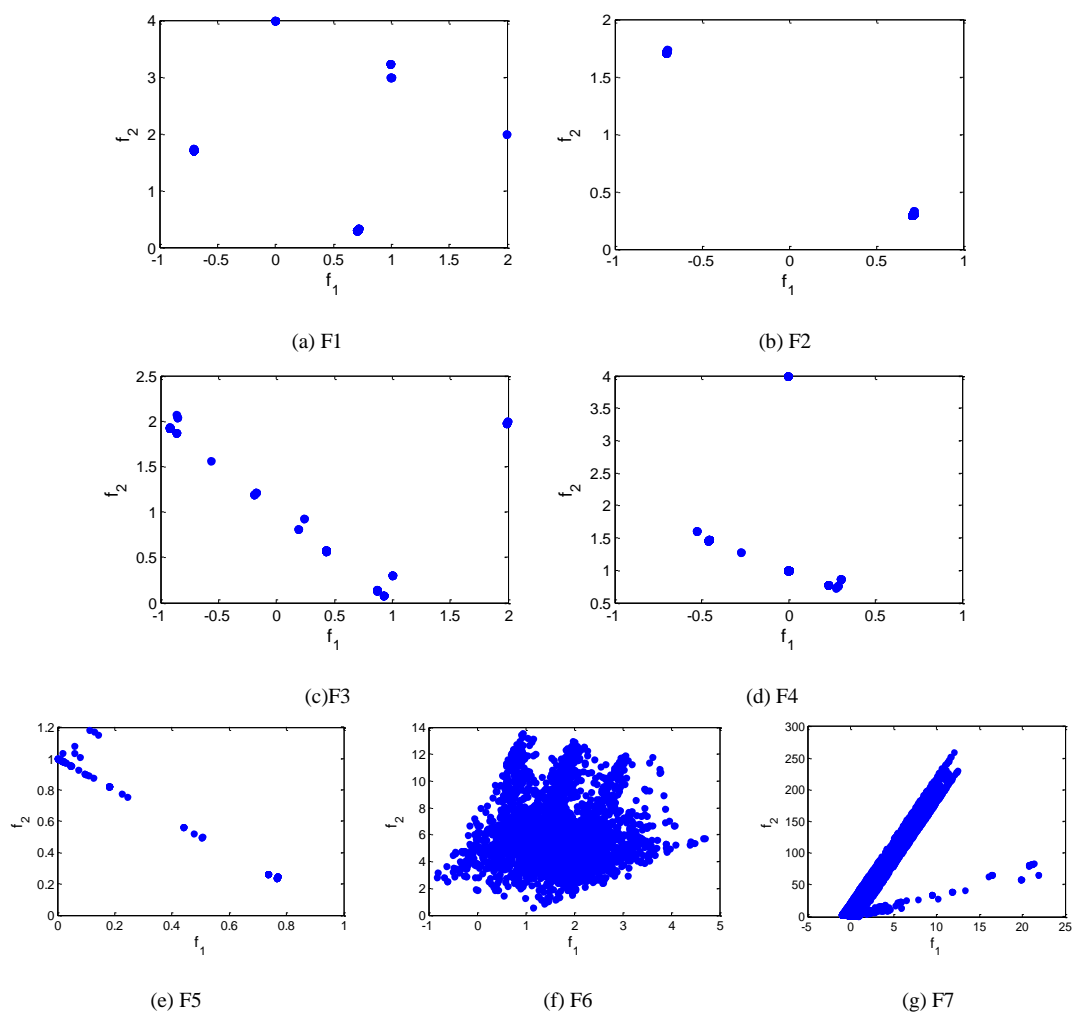
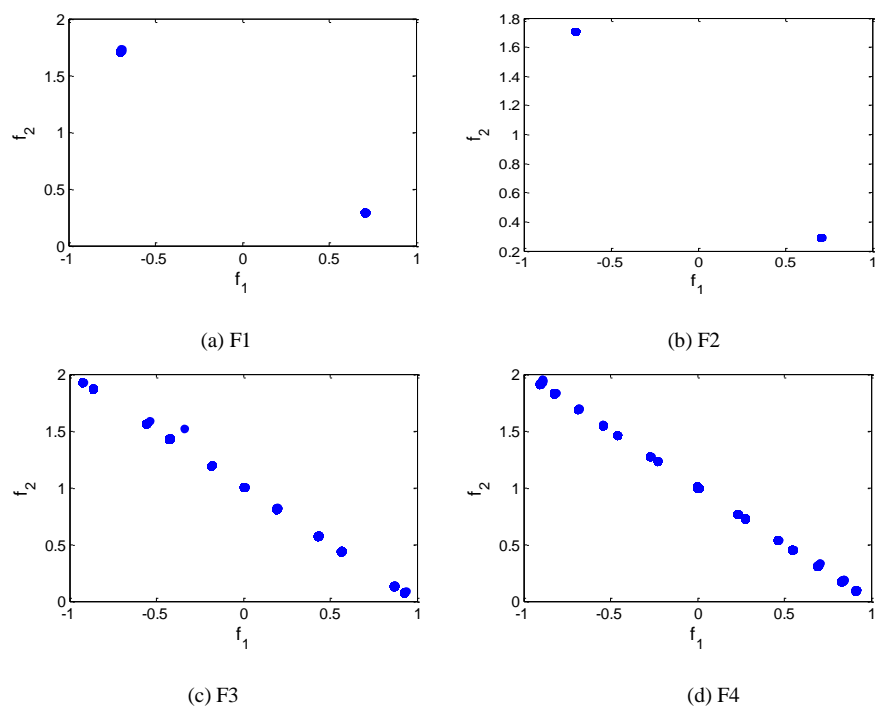


图 3-9 30 次变量空间中 MONES 最终结果分布



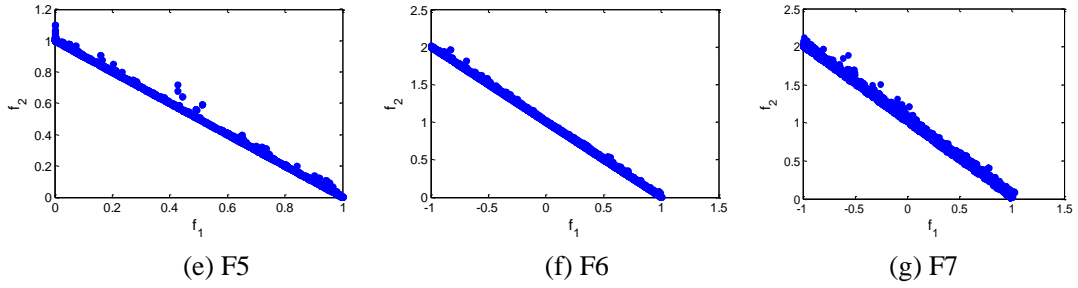


图 3-10 30 次变量空间中 MONES 最终结果分布

此外，对于 F5 函数 Pareto 的前沿面在目标空间已经被丢失了，一个值得注意的现象是，在 F1, F3 和 F4 函数中，CA 方法的种群保存了一些远离最优值的个体，这可能是 CA 方法自生所固有的缺点所带来的。这也间接的验证了上文 IGD 值随着进化的进行越来越大的现象，这一观察与我们在图 4 中的分析是一致的。最后 F6 和 F7 函数，CA 方法不能很好的近似帕累托前沿，另外在目标空间中，种群的分布是不规则。这是因为在极大目标的多目标优化问题中，个体之间不存在着支配关系因此，CA 方法变为 F6、F7 的随机游走。与 CA 方法相比较，MONES 更有效地找到所有的测试实例多个最优解。

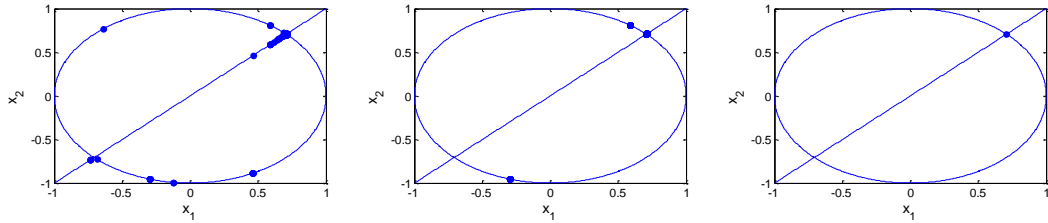


图 3-11 CA 方法 F1 函数决策空间一次运行

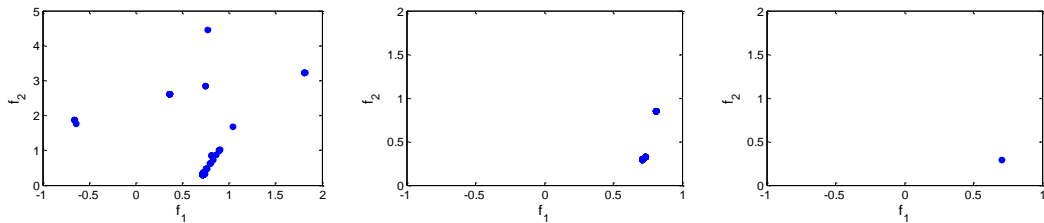


图 3-12 CA 方法 F2 函数决策空间一次运行

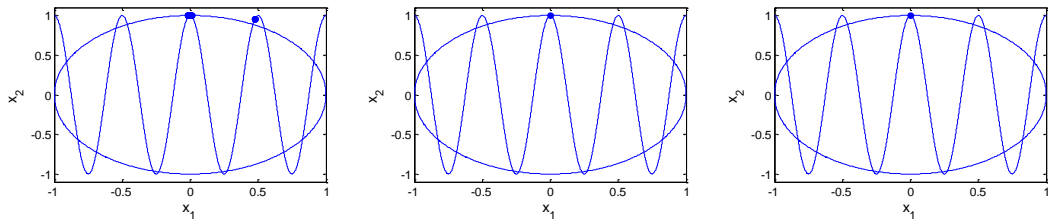


图 3-13 CA 方法 F2 函数决策空间一次运行

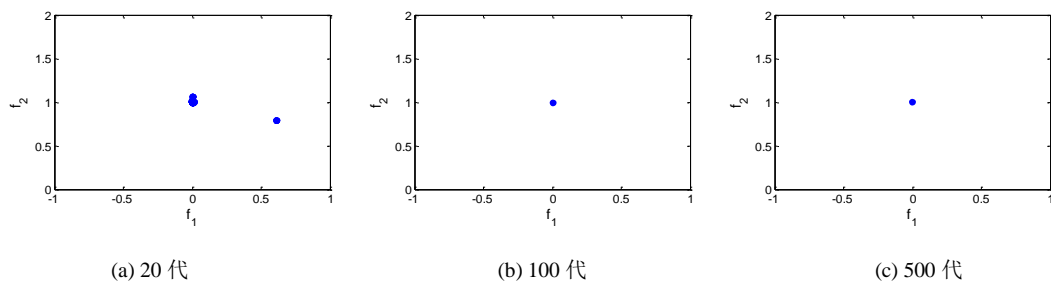


图 3-14 CA 方法 F1 函数决策空间一次运行

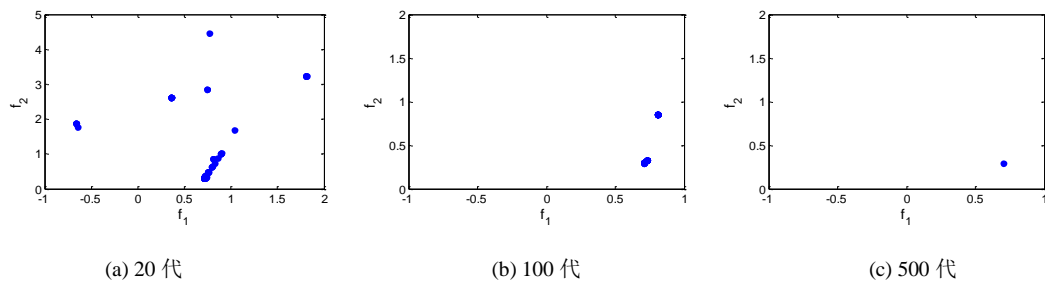


图 3-15 CA 方法 F2 函数的目标空间一次运行

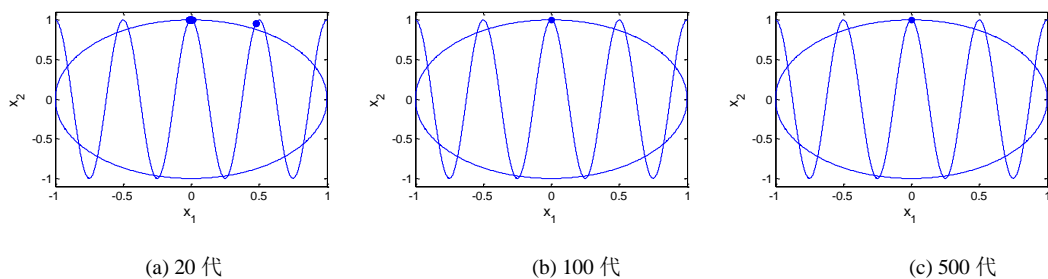


图 3-16 CA 方法 F3 函数的目标空间一次运行

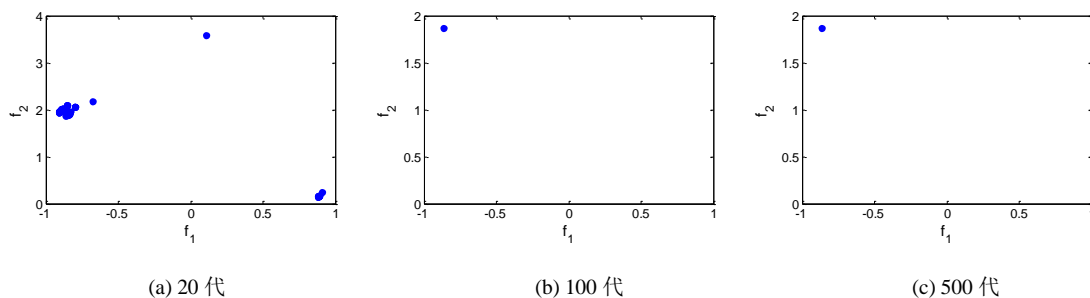


图 3-17 CA 方法 F3 函数的目标空间一次运行

3.4.4 与 CA 方法收敛行为的比较

此节为了进一步的研究 CA 和 MONES 方法的收敛行为,深入分析了在一次运行过程中的收敛行为,图 3-10 到 3-23 展示了 F1, F4, F5, 和 F6 函数, CA 和 MONES 方法在一次运行过程中的决策空间和目标空间的分布状况,由于 F6 函数的决策空间大于 2,因此无法将 F6 函数的一次典型运行的图提供出来。

从图 2-15 到图 2-20 中,早期的运行中,CA 能够发现某些解潜在的区域,但是在 100 代以后,这些个体转为搜索一个最优值的所在区域,有以下 2 个原因,首先 NES 通过 CA 方法转换为目标空间中的点后,其所有的解都对应着目标空间的一个点,在进化过程中随着选择的进行,一些较好的个体慢慢的丢弃掉了,另外一些潜在的靠近最优值的点被 Pareto 支配掉,从而排除出种群。除此之外,在 F6 函数中,CA 的收敛行为非常的随机,因为 F6 函数是转换过来的多目标优化问题具有极大目标的多目标优化问题的特征。

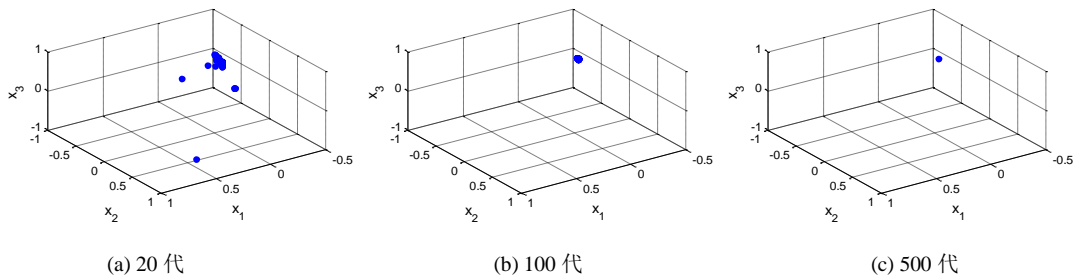


图 3-18 CA 方法在 F5 函数决策空间一次运行.

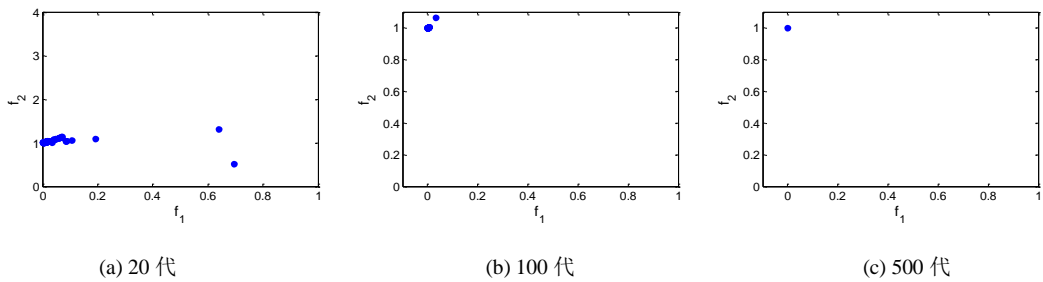


图 3-19 CA 方法在 F5 函数目标空间一次运行.

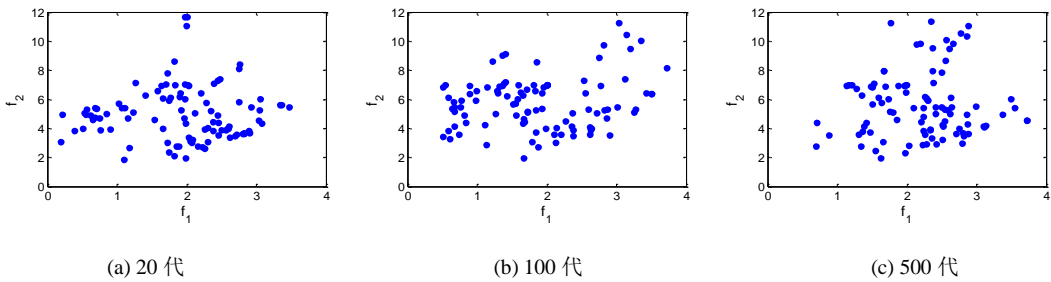


图 3-20 CA 方法在 F6 函数目标空间一次运行.

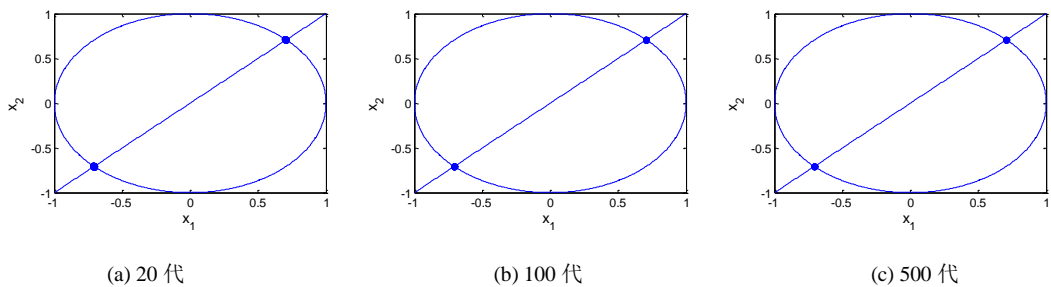


图 3-21 MONES 方法在 F1 函数目标空间一次运行.

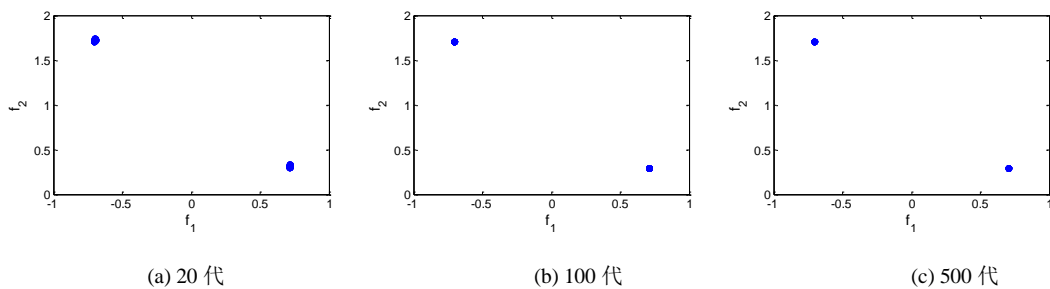


图 3-22 MONES 方法在 F1 函数目标空间一次运行.

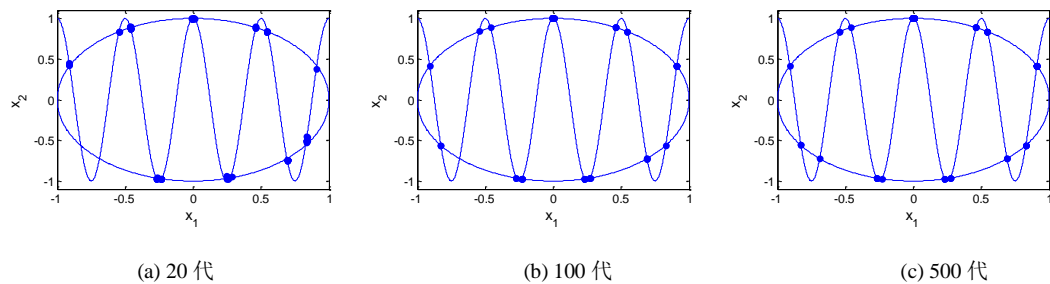


图 3-23 MONES 方法在 F3 函数目标空间一次运行.

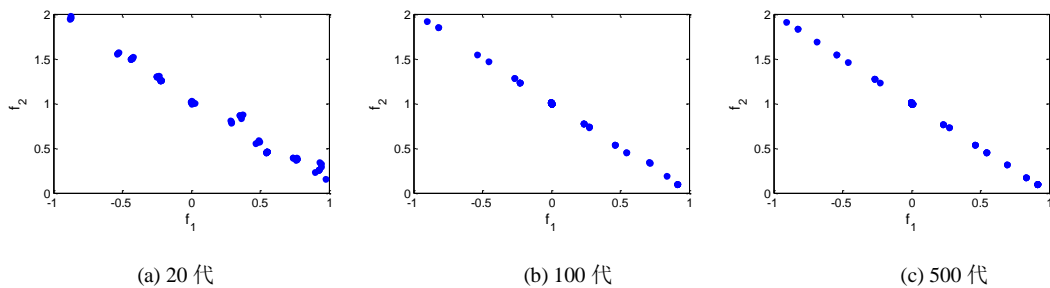


图 3-24 MONES 方法在 F4 函数目标空间一次运行.

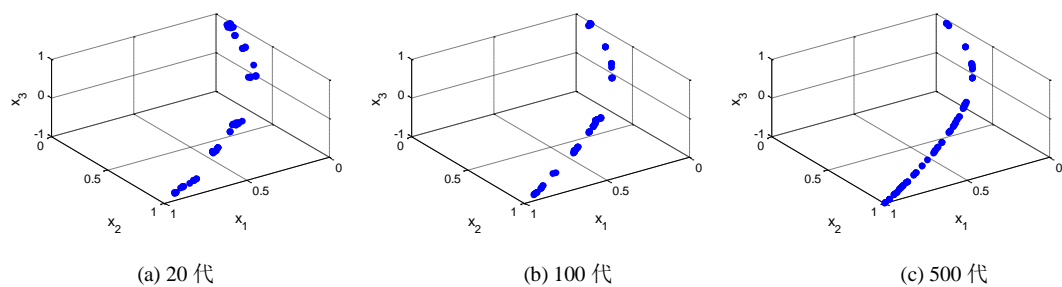


图 3-24 MONES 方法在 F5 函数决策空间一次运行

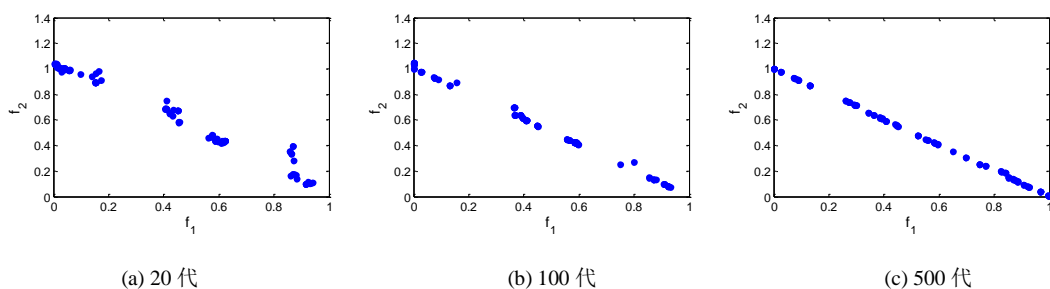


图 3-25 MONES 方法在 F5 函数目标空间一次运行

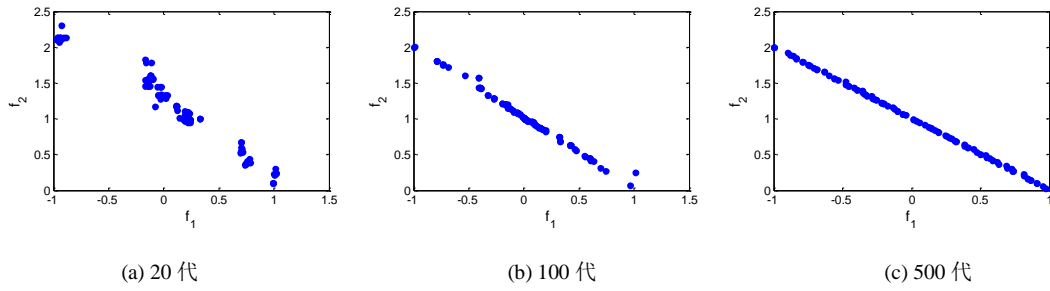


图 3-26 MONES 方法在 F6 函数目标空间一次运行

图 3-17 到图 3-23 展示了 MONES 的收敛行为。可以清楚的从图 17-20 看出，经过 100 次迭代，除了早期 F4 函数丢失了一个点，F1 和 F4 函数 MONES 是可以找到所有的最优解的，而 F4 函数丢失的最优解在进化的晚期有找到了。如图 21-23，F5 和 F6 函数，MONES 在一次 500 代运行的过程中，产生一组 Pareto 前沿面的近似集，另外一个有趣的现象是 F4 函数，在进化的早期，种群并没有覆盖所有的最优值潜在区域，可以从丢失的最优值看出，随着进化的进行，种群分布在不同的最优值附近（大约 20 次迭代）。这可以归因于一个事实，MONES 将一个 NES 的所有解映射到一个目标空间中不同的点，而多目标进化算法允许其种群中这些不具有支配关系的点同时存在。

结合上面的实验表明，MONES 作为一种多目标转换技术，比 CA 方法更有效的求解非线性方程组。

3.4.5 与单目标的进化优化方法比较

在上面通过与多目标进化优化方法-CA 方法的比较，已经获得了 MONES 的搜索特征，由于多解的非线性方程组本质上是一个单目标多模态优化问题，需要比较多目标转换方法与单目标转换方法的有效性。在这一部分，比较四个单目标优化方法。Hirsch 设计了一种改进的目标函数，以产生一个区域用以排斥周围已被发现的最优值，另外 C-GRASP 作为搜索引擎。由于 C-GRASP 本质上是单点的搜索算法，实现过程也是相当的复杂。由于普遍认为以群体为基础的元启发式算法通常比点对点的算法更强大，粒子群优化（PSO）和差分进化（DE）取代 C-GRASP。作为结果，我们得到了在本文中表示为 repulsionpso 和 repulsionde 两种方法。

对于非线性方程系统转化后的问题本质上是类似于多模态优化问题都可以包括多个最优解。因此，技术方法设计的多模态优化问题也可以适用于小生境单目标进化优化方法求解 NES 多个解。为了测试小生境技术的有效性能，选择了两个经典小生境方法。一个是使用环形拓扑的小生境粒子群算法（记为 RPSO）和拥挤距离的差分进化算法（记为 CDE）。应用 RPSO 和 CDE 解决 NES 时，采用的目标函数是方程 2-15。

表 3-4 与单目标进化优化方法的比较

Test Instance	RepulsionPSO	RepulsionDE	rpso	CDE	MONES
	Mean NOF \pm Std Dev	Mean NOF \pm Std Dev	Mean NOF \pm Std Dev	Mean NOF \pm Std Dev	Mean NOF \pm Std Dev
F1	1.73E+00 \pm 5.21E-01	-2.00E+00 \pm 0.00E+00	\approx 1.43E+00 \pm 6.26E-01	-2.00E+00 \pm 0.00E+00	\approx 2.00E+00 \pm 0.00E+00
F2	4.67E-01 \pm 5.07E-01	-0.00E+00 \pm 0.00E+00	-3.50E-01 \pm 5.57E-01	-0.00E+00 \pm 0.00E+00	-2.00E+00 \pm 0.00E+00
F3	1.63E+00 \pm 8.09E-01	-6.13E+00 \pm 1.22E+00	-1.60E+00 \pm 8.14E-01	-4.43E+00 \pm 1.72E+00	-1.10E+01 \pm 0.00E+00
F4	3.43E+00 \pm 1.33E+00	-8.07E+00 \pm 1.82E+00	-2.97E+00 \pm 1.21E+00	-8.26E+00 \pm 1.86E+00	-1.41E+01 \pm 1.16E+00
F5	2.77E+00 \pm 1.38E+00	-6.70E+00 \pm 1.62E+00	-2.63E+00 \pm 1.27E+00	-1.45E+01 \pm 3.15E+00	-3.73E+01 \pm 7.05E+00
F6	6.67E-02 \pm 2.54E-01	-1.50E+00 \pm 6.29E-01	-1.00E-01 \pm 3.05E-01	-1.95E+01 \pm 4.30E+00	-7.31E+01 \pm 3.81E+00
F7	5.57E+00 \pm 1.63E+00	-1.63E+00 \pm 6.69E-01	-9.47E+00 \pm 2.26E+00	-6.57E+00 \pm 3.09E+00	-2.02E+01 \pm 1.15E+01
-	7	6	7	6	
+	0	0	0	0	
\approx	0	1	0	1	

“—”, “+”, 和 “ \approx ” 代表差于、好于和相似于 MONES

对于此四个单目标优化方法, 即 repulsionpso, repulsionde, RPSO, CDE, 在 7 个测试函数进行了 30 次独立运行实验。为了使比较公平, 和 MONES 相同这四个方法设置种群规模为 100 和最大代数设置为 500。repulsionpso 和 CDE, 分别设置为 0.9 和 0.1 的比例因子和 DE 的交叉控制参数, 表 3-6 显示了 MONES 具有最优性能。

以上的比较表明, 结合排斥区域或小生境技术的单目标的进化优化方法有能力对一些测试实例找到多个最优解, 但是在鲁棒性和求解精度上其性能差于 MONES。与单目标形式化方法相比, MONES 的主要优势是将一个多解的非线性方程组的所有解都映射到一个转换的双目标优化问题的前沿面上, 从而能够达到良好的分布。

3.4.6 不同形状的前沿面实验

为了研究的关键参数对性能的影响, 我们更换了方程 (12) 方程 (21) 和测试的关键有四个不同的: 1, 3, 5, 和 7。有必要强调, 所有新的最优解位于非线性帕累托前用激素和激素与激素总结表五 NOF 指标值的七个测试实例。参数设置为同一节中介绍 v-d. 水平或垂直。我们选择 F3 为例来解释这个。图 2-7 显示算法具有不同参数所求得的 F3 函数中 11 最优解分布的图像, 从图 2-7 很明显看出, F3 的一些最优解, 在帕累托前沿面几乎呈水平分布, 从而使得 Pareto 前沿面形状较为复杂, 使用多目标进化算法较难以发现。

表 3-5 不同形状的前沿面的比较

Test Instance	$\gamma=3$	$\gamma=5$	$\gamma=7$	$\gamma=1$
	Mean NOF \pm Std Dev	Mean NOF \pm Std Dev	Mean NOF \pm Std Dev	Mean NOF \pm Std Dev
F1	2.00E+00 \pm 0.00E+00 \approx	2.00E+00 \pm 0.00E+00 \approx	2.00E+00 \pm 0.00E+00 \approx	2.00E+00 \pm 0.00E+00
F2	1.77E+00 \pm 4.31E-01 $-$	1.23E+00 \pm 4.30E-01 $-$	1.23E+00 \pm 4.30E-01 $-$	2.00E+00 \pm 0.00E+00
F3	8.83E+00 \pm 8.74E-01 $-$	6.80E+00 \pm 8.87E-01 $-$	6.13E+00 \pm 6.29E-01 $-$	1.10E+01 \pm 0.00E+00
F4	1.04E+01 \pm 1.06E+00 $-$	7.50E+00 \pm 1.04E+00 $-$	6.17E+00 \pm 1.14E+00 $-$	1.41E+01 \pm 1.16E+00
F5	3.51E+01 \pm 7.72E+00 \approx	2.52E+01 \pm 1.03E+01 $-$	1.77E+01 \pm 1.16E+01 $-$	3.73E+01 \pm 7.05E+00
F6	5.89E+01 \pm 7.05E+00 $-$	4.07E+01 \pm 5.37E+00 $-$	3.24E+01 \pm 5.25E+00 $-$	7.31E+01 \pm 3.81E+00
F7	1.60E+01 \pm 1.13E+01 \approx	1.36E+01 \pm 1.03E+01 $-$	1.13E+01 \pm 7.83E+00 $-$	2.02E+01 \pm 1.15E+01
$-$	4	6	6	
$+$	0	0	0	
\approx	3	1	1	

“ $-$ ”, “ $+$ ”, 和 “ \approx ” 代表差于、好于和相似于 MONES 使用 $\gamma=1$ 的前沿面

此外，可以看出从表 3-7，对关键性能逐渐劣化与图 3-25 的增加也验证了这一点。在图 3-25 中，最优解的数量不断减少而 IGD 值的增加，因为较大的值，帕累托前沿面很复杂，在前沿面的中段几乎是水平的和垂直的段。

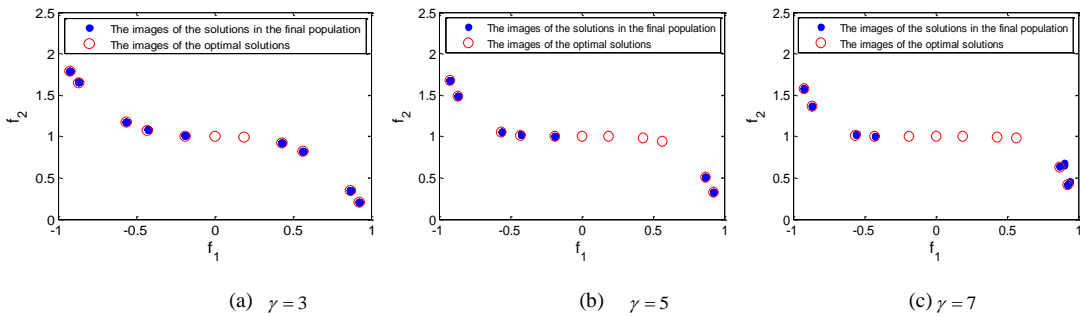


图 3-27 F3 函数不同形状的最终结果

转换函数的参数的值可以被设置为一个正奇数从而满足本文的定位函数的定义。如果是一个正奇数大于 1，帕累托前面的转化问题将成为非线性。实验结果表明，非线性帕累托方面对多目标进化算法的性能有所影响，与线性帕累托前相比较其求解性能较差。在进化多目标优化的社区，为了比较不同算法的性能，Pareto 前沿面通常表现出复杂的非线性特征。本文的主要目的是提出一个简单有效的转化技术解决非线性方程系统，而不是为了测试多目标进化算法的性能。

3.5 结论

非线性方程组具有多个最优解。在过去的十年中，进化算法（EAs）备受关注。当使用 EAs 解决其转换而来的优化问题，对于非线性方程系统的求解起着至关重要的作用。本文提出了一种简单而通用得求解非线性方程组系统的多目标转换技术，称为 **MONES**。它把一个非线性方程组转换成为一个双目标优化问题。随后，转化后的多目标优化问题可以使过多目标优化算法。由于 **CA** 和 **MONES** 是属于同类的转化技术，并且是最早的提出使用多目标进化算法求解多模态优化问题的方法，与之进行了深入的实验比较了他们之间的区别，分析了它们的性能。实验结果表明，与 **CA** 方法相比，**MONES** 表现出优越的性能，对两方面性能指标的 7 个测试实例，而 **CA** 方法是有能力同时定位在一个单一的运行 **NES** 多个最优解。采用基于单目标优化方法的比较进一步验证具有有效性。此外，研究不同的帕累托前沿面对算法性能的影响，讨论了不同前沿面的求解性质。

有必要强调的是，非线性方程系统的转换模型是多模态优化问题的一种特例，带有先验信息，而对于多模态优化问题的转换并不能直接套用过去，但是本章节提出来的多目标转换方法是求解多模态优化问题的基础。在下一章节中，将介绍求解多模态优化问题的多目标进化算法。

第四章 多模态优化问题的多目标进化求解

在上一章中,针对带有先验信息的多模态优化问题-多解的非线性系统方程组进行分析,提出了多目标转换方法,是否此种方法对于一般的多模态优化问题也有效,是本章中要解决的问题。为表述方便,一个多模态优化问题可以形式化如下:

$$\text{maximize/minimize } f(\vec{x}) \quad (4-1)$$

其中 $\vec{x} = (x_1, K, x_D) \in S$ 为决策向量, $L_i \leq x_i \leq U_i$ ($i \in \{1, K, D\}$) 为第 i 个决策向量, L_i 和 U_i 为 x_i 的上界和下界, D 为变量的个数, $f(\vec{x})$ 是目标函数, S 为决策变量空间, 其定义为:

$$S = \prod_{i=1}^D [L_i, U_i] \quad (4-2)$$

使用进化算法求解多模态优化问题时,需要知道多模态优化问题的最优值的个数,通过划分问题的变量空间的形式划分种群,而在上一章节,知道使用多目标优化方法求解非线性方程的多模态优化模型时能够较少的引入问题的相关知识,因此很自然的将上一章节的多目标优化方法求解一般的多模态优化问题,但是由于多模态优化问题和非线性方程组的多模态转换模型的最优解集的形式并不相同,多模态优化问题的最优值并没有相关的先验知识,比较于非线性方程系统的最优值分布,多模态优化问题的变量空间复杂,其形态呈现出更多的特征。在本章使用上一章节的双目标转换模型求解多模态优化问题,使用在进化过程中的知识作为其已知的最优值,在求解过程中,发现了 2 个新的问题,称之为维数遮挡问题以及峰的吸引问题,针对这些问题本章提出定位多模态优化问题的多个最优值多目标进化优化算法 (Multiobjective Optimization for Locating Multiple Optimal Solutions of Multimodal Optimization Problems, 简称为 MOMMOP), 此算法将迭代寻优过程中的最优值作为参考最优值,为了解决维数遮挡问题,将多模态优化问题转换为 D 个 2 维目标的多目标优化问题,为了解决最优值吸引问题,引入了新的支配方式,此外为了更好的求解多模态优化问题,使用了差分进化算法作为搜索引擎。本章 4.1 节将给出一种双目标的转换方法和相关证明, 4.2 节中将给出其多目标转换框架, 4.3 节给出算法流程, 4.4 节实验对比了相关的多模态优化进化算法。

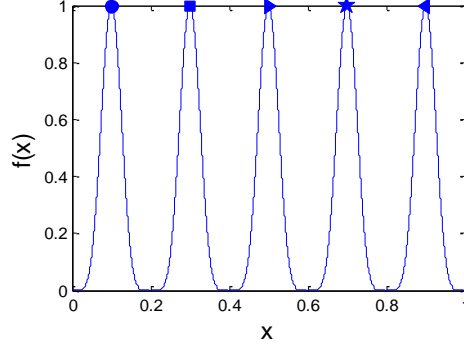
4.1 双目标转换方法

在本小节中,将给出多模态优化问题转换为双目标优化问题的转换形式,并探讨其能够有效的定位多个最优值的原理-隐含小生镜技术。

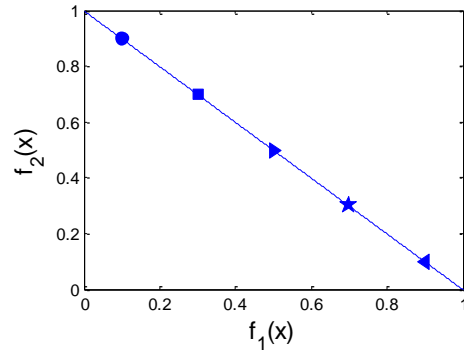
将多模态优化问题转换为双目标优化问题时,其 $\alpha(\vec{x})$ 函数不变,仍然采用以下形式:

$$\begin{cases} \text{minimize } \alpha_1(\vec{x}) = x_1 \\ \text{minimize } \alpha_2(\vec{x}) = 1 - x_1 \end{cases} \quad (4-3)$$

$\alpha(\vec{x})$ 仍然满足其在其变量空间内处处矛盾，并且其前沿面形状呈“y=1-x”的性质。仍然可以从图 4-1 中可以看出，对于函数的 5 个最优值，使用双目标转换方法，映射到目标空间中时，此 5 个点互不相同，在支配关系上不存在着比较的可能。



(a) 函数图像



(b) 目标空间图像

图 4-1 原始多模态函数图像及其所对应的多目标转换的目标空间

系统函数 $\beta(\vec{x})$ 的定义如下：

$$\text{minimize } \beta(\vec{x}) = \frac{|f(\vec{x}) - \text{BestOFV}|}{|\text{WorstOFV} - \text{BestOFV}|} \cdot (U_1 - L_1) \cdot \eta \quad (4-4)$$

其中 $\text{BestOFV} = f(\vec{x}^*)$ ，其中 \vec{x}^* 是最优值，必然有 $\beta(\vec{x}^*) = 0$ ，因为 x_1 的值在 L_1 和 U_1 之间，加入 $(U_1 - L_1)$ 的目的是使 $\beta(\vec{x})$ 和 $\alpha(\vec{x})$ 具有相同的尺度。由于 \vec{x}^* 在算法执行以前，并不知道其值的大小，因此使用进化过程中所找到的最优值作为 BestOFV 所代替。

因此多模态优化问题的双目标转换函数形式化如下：

$$\begin{cases} \text{minimize } f_1(\vec{x}) = x_1 + \frac{|f(\vec{x}) - \text{BestOFV}|}{|\text{WorstOFV} - \text{BestOFV}|} \cdot (U_1 - L_1) \cdot \eta \\ \text{minimize } f_2(\vec{x}) = 1 - x_1 + \frac{|f(\vec{x}) - \text{BestOFV}|}{|\text{WorstOFV} - \text{BestOFV}|} \cdot (U_1 - L_1) \cdot \eta \end{cases} \quad (4-5)$$

根据 4-5 定义，有如下定理：

定理 4.1：任意的向量 \vec{a} 和 \vec{b} ， $\vec{a}, \vec{b} \in S$ ，必然有 \vec{a} 和 \vec{b} 互不支配。

定理 4.2：如果一个向量 \vec{x} 为多模态优化问题的一个最优值，必然有 $\beta(\vec{x}) = 0$ 。

证明：如果 \vec{x} 为最优值，则必然有 $f(\vec{x}) - \text{BestOFV} = 0$ ，则有 $\beta(\vec{x}) = 0$ 。

定理 4.3：多模态优化问题的最优值在所构造的多目标转换函数的前沿面上。

证明：假设 \vec{x}^* 为一个多模态优化问题的最优值，而 \vec{x}' 为一个非线性方程组的任意决策变量，有 $\vec{x}' \neq \vec{x}^*$ 。由于 \vec{x}^* 为多模态函数的最优值，根据定理 4.2 必然 $\beta(\vec{x}^*) = 0$ ，那么 $f_1(\vec{x}^*) = \alpha_1(\vec{x}^*)$ 和 $f_2(\vec{x}^*) = \alpha_2(\vec{x}^*)$ ，而 \vec{x}' 有 $\beta(\vec{x}') \geq 0$ ， $f_1(\vec{x}') = \alpha_1(\vec{x}') + \beta(\vec{x}')$ 和 $f_2(\vec{x}') = \alpha_2(\vec{x}') + \beta(\vec{x}')$ ，根据 $f_1(\vec{x}') \leq f_1(\vec{x}^*)$ 重写有 $\alpha_1(\vec{x}') + \beta(\vec{x}') \leq \alpha_1(\vec{x}^*)$ ，存在 $\alpha_1(\vec{x}') - \alpha_1(\vec{x}^*) \leq \beta(\vec{x}') \leq 0$ ，推出 $\alpha_1(\vec{x}') - \alpha_1(\vec{x}^*) \leq 0$ ，同样可得到 $\alpha_2(\vec{x}') - \alpha_2(\vec{x}^*) \leq 0$ ，从而 \vec{x}^* 在 $\alpha(\vec{x})$ 上都要小于 \vec{x}' ，与定义相矛盾。

定理 4.4 所有的多模态函数的最优值在所构造的多目标转换问题的目标空间函数“ $y = 1 - x$ ”上。

由定理 4.3 都在所构造的多目标转换问题的前沿面上，并且满足 $\beta(\vec{x}) = 0$ ，其前沿面满足 $\alpha(\vec{x})$ 形状，而 $\alpha(\vec{x})$ 的目标空间为“ $y = 1 - x$ ”；

需要指出的是，虽然将一个多模态优化问题的最优值映射到了一个多目标优化问题的前沿面上，并不是说就解决了此问题，因为此多目标优化问题在变量空间的值仍然是未知的，需要使用多目标进化算法进行优化。

为了更好的理解以上多目标转换方法的工作方式，将给出一个多模态优化问题的实例，此多模态优化问题的函数图像如图所示，此问题为最大化，包含了一个决策变量和 2 个最优值，假设存在 4 个解，分别以 x_a, x_b, x_c 和 x_d 代表这些点的位置信息：

$(x_a, f(x_a)) = (0.1, 1.0)$ ， $(x_b, f(x_b)) = (0.15, 0.97)$ ， $(x_c, f(x_c)) = (0.2, 0.85)$ ，和 $(x_d, f(x_d)) = (0.8, 0.85)$ 。从图 4-2 (a) 中可以观察到 x_a 为最优值，并且 x_a, x_b 和 x_c 在同一个峰，假设此问题的 $\text{BestOFV} = 1, \text{WorstOFV} = 0, U_1 = 1$ ，和 $L_1 = 0$ 。

基于以上信息，其双目标的转换函数如下：

$$\begin{cases} \text{minimize } f_1(\vec{x}) = x_1 + |f(\vec{x}) - 1| \cdot \eta \\ \text{minimize } f_2(\vec{x}) = 1 - x_1 + |f(\vec{x}) - 1| \cdot \eta \end{cases} \quad (4-6)$$

设置 $\eta = 1$ 相关的目标值为 $(f_1(x_a), f_2(x_a)) = (0.1, 0.9)$ ， $(f_1(x_b), f_2(x_b)) = (0.18, 0.88)$ ， $(f_1(x_c), f_2(x_c)) = (0.35, 0.95)$ 以及 $(f_1(x_d), f_2(x_d)) = (0.95, 0.35)$ 。根据定义， x_a, x_b 和 x_d 是非支配解。而 $\eta = 4$ ，相关的目标函数值是 $(f_1(x_a), f_2(x_a)) = (0.1, 0.9)$ ， $(f_1(x_b), f_2(x_b)) = (0.27, 0.97)$ ， $(f_1(x_c), f_2(x_c)) = (0.8, 1.4)$ ， $(f_1(x_d), f_2(x_d)) = (1.4, 0.8)$ 。图 4-2 (c) 中可以看出 x_a 和 x_d 是非支配解集。

由此可以得到即使 x_c 和 x_d 拥有相同的多模态函数值， x_a 仍然支配在同一个相同的

峰的 x_c ，而不支配不同的峰的 x_d ，这显示了 MOMMOP 能够在同一个峰具有支配关系，而不同的峰并不具备支配关系，隐含了小生境技术，这是 MOMMOP 能够求解多模态优化问题的基本原理。

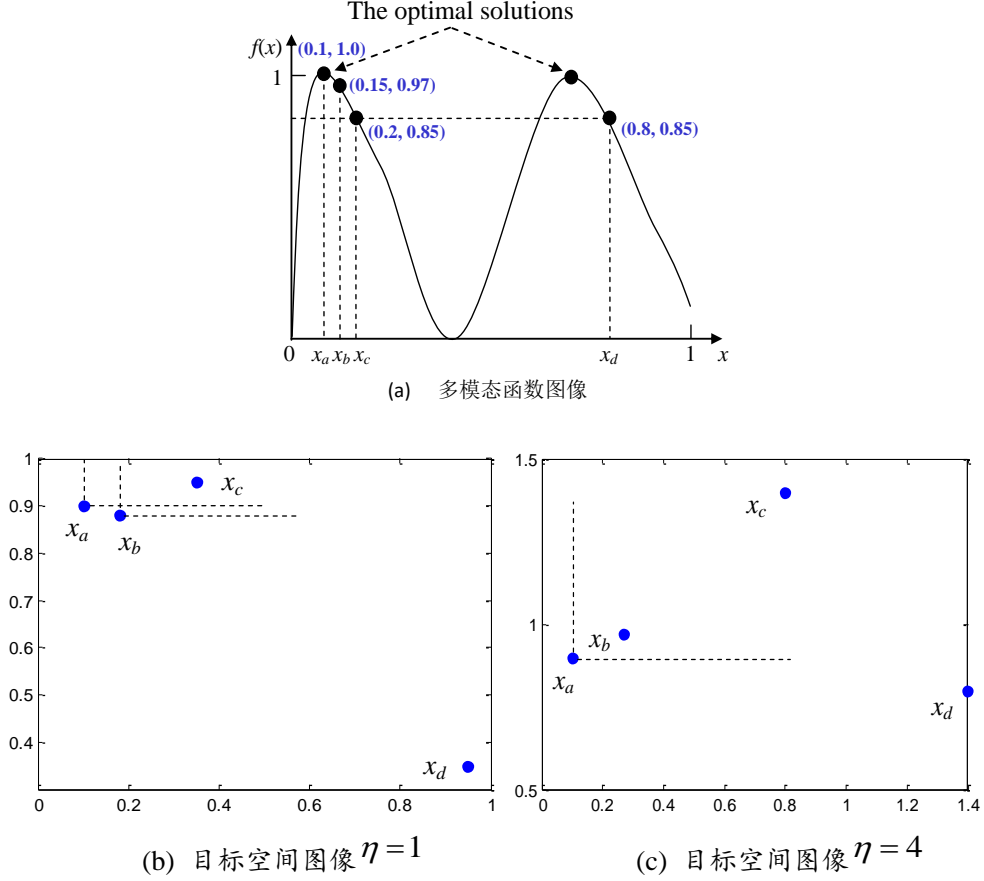


图 4-2 MOMMOP 的选择原理

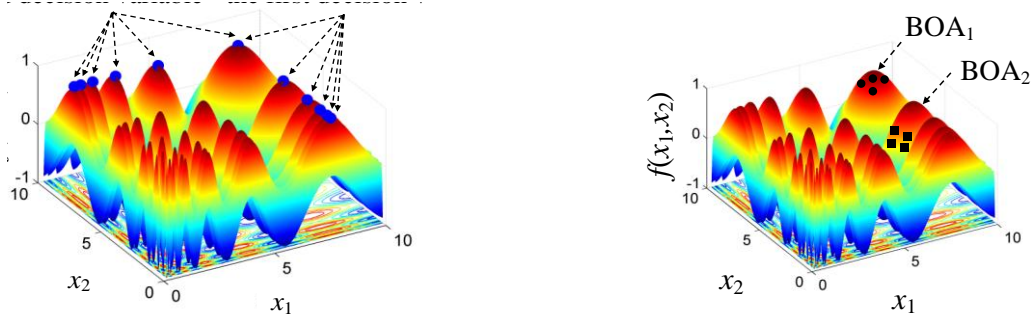
同时也能够观察到 Pareto 支配关系主要是依赖于决策变量 x_1 和目标函数值 $f(\vec{x})$ ，因为 $BestOFV$ ， $WorstOFV$ ， U_1 ， L_1 ，和 η 对于种群中所有的个体都是相同的。因此 MOMMOP 本质上平衡决策变量和目标函数，而此平衡 η 有着很大的作用，从实例可以看出当 $\eta=1$ ， x_a 能够 Pareto 支配 x_c ，当 $\eta=4$ ， x_a 能够 Pareto 支配 x_b 和 x_c 因此意味着 η 越大，其选择压力越大。

4.2 多目标目标转换框架

采用上节所提到的双目标转换方法，在求解一些特殊的多模态优化问题上性能并不好，在本文所设计的实验中所得到的数据无论在求解精度上还是求得的最优值的数目上都存在着较大的问题，下面将分析其具体原因，提出新的支配方式求解此类问题。

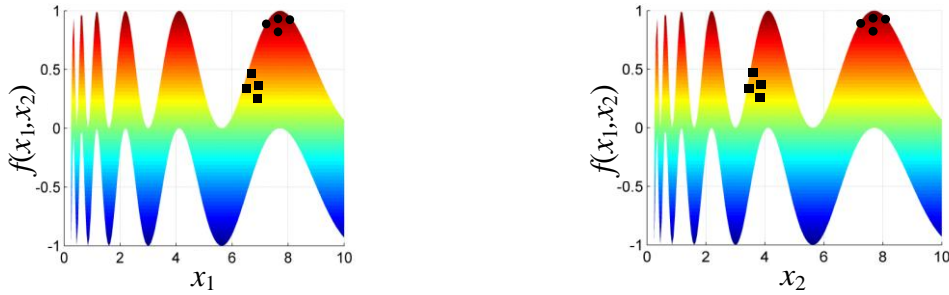
4.2.1 维数遮挡问题

定义 4.1 维数的遮挡问题：我们说一个多模态优化问题具有维数遮挡性质，其存在着 2 个最优值向量 \hat{u} 和 \hat{v} ，其中 $\exists i \in \{1, \dots, d\}$ ，使 $u_i = v_i$ 。



(a) 不同的最优值相同的决策变量值

(b) 2 个峰的吸引



(c) 个体的第一维映射

(d) 个体的第二维映射

图 4-3 Vincent 函数示意图

在一些比较特殊的多模态优化问题中，一些最优值可能在一些决策变量的位置上是相同的，以经常作为性能评价的 2 维 Vincent 函数为例，从图 4-3 (a) 看出函数具有 36 个最优值，6 个为一组，其变量空间的适应值景观如图 4-3 (a) 所示。一些最优值在第一个决策变量上具有相同的值，而在第二个决策变量上也具有相同的现象图 4-3 (b)。所造成问题是：将其进行多目标转换，一些山峰上的点由于映射到一个决策变量所形成的双目标空间，造成一些潜在的最优点被更好的点所支配掉，破坏了隐含的小生境，由于种群资源是有限的，从而造成有些最优值在进化过程中被丢弃掉，从而无法得到其所在区域的最优解。

解决此问题的一个方法是不被维数遮挡的问题的决策变量，进行构造双目标转换函数，但是存在以下两个问题，首先有些问题所有的维数都存在着维数遮挡性质，可以看到 3 维的 Vincent 函数，当映射到具有第二维决策变量的空间时，第一维仍然是被遮挡的，此现象由图 4-3 (c) 和 (d) 可以看出。此外在优化之前并没有多模态优化

问题的先验信息，也就是说，对于一个多模态优化问题并不知道哪一维存在着维数遮挡的问题。为了解决此问题，必须将每一个维数都做出转换，有如下的方式：

$$\begin{aligned}
 BOP_1 & \begin{cases} \text{minimize } x_1 + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_1 - L_1) \cdot \eta \\ \text{minimize } 1 - x_1 + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_1 - L_1) \cdot \eta \end{cases} \\
 BOP_2 & \begin{cases} \text{minimize } x_2 + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_2 - L_2) \cdot \eta \\ \text{minimize } 1 - x_2 + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_2 - L_2) \cdot \eta \end{cases} \\
 & \vdots \\
 BOP_D & \begin{cases} \text{minimize } x_D + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_D - L_D) \cdot \eta \\ \text{minimize } 1 - x_D + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_D - L_D) \cdot \eta \end{cases}
 \end{aligned} \quad (4-7)$$

上式构建了 D 个双目标优化问题 $BOP_1, BOP_2, \dots, BOP_D$ ，在个体比较时，每一个优化问题都不被支配，才能造成事实上的非支配。

定义 4-2：多模态 Pareto 支配：说一个决策向量 \vec{x}_u 支配另外一个决策向量 \vec{x}_v ，如果 $\vec{x}_u \succ \vec{x}_v$ 在所有的 D 个双目标优化问题上有如下关系：

$$\vec{x}_u \succ \vec{x}_v \text{ on } BOP_1 \wedge \vec{x}_u \succ \vec{x}_v \text{ on } BOP_2 \wedge \dots \wedge \vec{x}_u \succ \vec{x}_v \text{ on } BOP_D$$

从图 4-3 中可以看出，属于同一个山峰上的点，转换到 2 维目标时虽然被支配，但是在 BOP2 上并没有支配的可能性，从而维持了小生境技术。

4.2.2 吸引问题

使用定义 4-2 的支配方式进行个体比较，在种群的进化过程中，发现一个现象，如果多模态优化问题的峰值的峰顶越平坦，种群中的个体越趋向于此峰值进行搜索，其定位的个体就越多，而峰值越小，个体较少的搜索此峰。可以使用 CEC2013 的组合函数展示此现象，可以从图 4-4 中看出， \vec{x}_a 和 \vec{x}_b 所定位的峰只包含了一个个体，在没有改进此两个峰值的质量的情况下，由于种群并没有更多的个体集中在此位置，因此导致此区域难以得到有效的搜索，从而在进化时导致解的丢失，另一方面 $\vec{x}_c, \vec{x}_d, \vec{x}_e, \vec{x}_f, \vec{x}_g, \vec{x}_h, \vec{x}_i, \vec{x}_j$ 都在一个相同的峰上集中进行搜索，导致进化搜索的效率降低。

分析其原因，通过图 4-3 可以看出，点 A 不是最优值的点，也不会被最优值所支配。这是因为转换函数只能满足转换的充分条件而不是必要条件，从而才造成此问题的存在。在多目标的转换框架中，有些个体不是最优值，其仍然在此多目标优化问题的前沿面上。

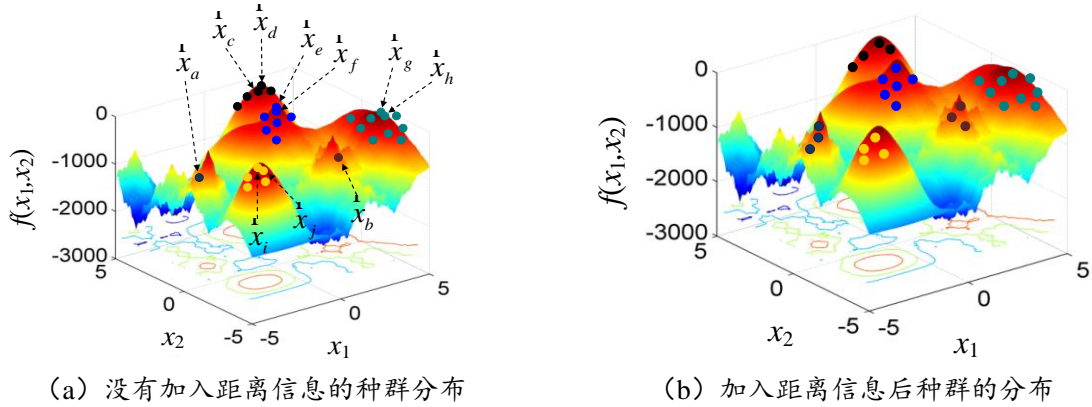


图 4-4 变量组合函数

由图 4-3 中可以看出，一个最优值附近的个体往往由于没有被其最优值所支配，其一直会存在于进化的种群中，为了减轻此方面的影响，需要新的比较标准引入到支配关系的比较中。将个体的距离信息引入到支配关系的比较中将有效的缓解此种现象。

定义：我们说 \vec{x}_u 支配向量 \vec{x}_v 如果有以下形式：

$$\begin{aligned} f(\vec{x}_u) &\leq f(\vec{x}_v) \\ \text{distance}(\text{normalization}(\vec{x}_u, \vec{x}_v)) &< 0.01 \end{aligned} \quad (4-8)$$

$f(\vec{x}_u)$ 和 $f(\vec{x}_v)$ 是多模态函数 \vec{x}_u 和多模态函数 \vec{x}_v 而所转换过来的目标函数， $\text{distance}(\text{normalization}(\vec{x}_u, \vec{x}_v))$ 表示 \vec{x}_u 和 \vec{x}_v 为决策空间归一化欧几里得距离 ($x_{u,i} = (x_{u,i} - L_i) / (U_i - L_i)$ 和 $x_{v,i} = (x_{v,i} - L_i) / (U_i - L_i)$ 对每一个 $i \in \{1, \dots, D\}$) 如果 $\text{distance}(\text{normalization}(\vec{x}_u, \vec{x}_v)) \approx 0$ 考虑 \vec{x}_u 和 \vec{x}_v 非常相近就认为其非常相似。

4.3 算法整体流程与复杂度分析

通过以上的分析，设计多目标进化算法定位多模态优化问题的多个解的整体算法流程图如下所示：

算法：MMMOP 的整体流程图

输入：所要求解问题

输出：多个所要求解问题的最优值

$G = 0$; // G is the generation number

生成初始种群 $P_0 = \{\vec{x}_1, \dots, \vec{x}_N\}$

Evaluate the multimodal function values of the initial population $f(\vec{x}_1), \dots, f(\vec{x}_N)$;

Compute *BestOFV* and *WorstOFV*;

```

FEs = N;      // FEs records the number of fitness evaluations
While FEs < MaxFEs do      // MaxFEs 代表函数最大适应值评价次数
    Implement the crossover and mutation of DE to produce an offspring population  $Q_G$ ;
    Evaluate the multimodal function values of the offspring population  $Q_G$ ;
    FEs = FEs + N;
    Update BestOFV and WorstOFV;
     $H_G = P_G \cup Q_G$ ;
    Compute the  $D$  objective vectors of each individual in  $H_G$ ;
    Partition  $H_G$  into  $k$  Pareto fronts denoted as  $F_1, \dots, F_k$  by applying the
    nondominated sorting;
     $P_{G+1} = \emptyset$  and  $i = 1$ ;
    While  $|P_{G+1}| + |F_i| \leq N$  do      //  $|P_{G+1}|$  和  $|F_i|$  代表  $P_{G+1}$  和  $F_i$  个体的数量
         $P_{G+1} = P_{G+1} \cup F_i$ ;
         $i = i + 1$ ;
    End While
     $P_{G+1} = P_{G+1} \setminus F_i$ ;
    Find  $(N - |P_{G+1}|)$  individuals with the largest crowding distances in  $F_i$  and
    incorporate them into  $P_{G+1}$ ;
     $G = G + 1$ ;
End While

```

构造支配集的算法与上一章并没有区别,其中拥挤距离计算的是目标空间的距离,而不是变量空间的距离。

4.3.1 算法的时间复杂度分析

在 MOMMOP 的执行过程中,每一个个体需要转换 D 次双目标优化问题,以上所提出的算法有 2 个过程所决定,第一个基本的非支配排序,每个个体需要与其它个体比较一次支配关系,因此每个个体需要 MN 次比较,执行 N 次,因此有 MN^2 次比较, M 为目标个数和 N 种群大小.需要指出由于有 D 个目标,个体比较需要探测 D 次的支配关系.因此其时间复杂度为:

$$O(2DN^2) \quad (4-9)$$

由于在计算时并不需要每次都比较每个目标,因此一般为 $O(2N^2)$,第二个计算每个个体之间的欧几里得距离,时间复杂度为 $O(4N^2)$. 总共的时间复杂度为:

$$O(2DN^2 + 4N^2) \quad (4-10)$$

4.4 实验

CEC2013 提供了一组基准的测试集, 在这个测试集上可以进行较为公平的比较实验, 前面的 10 个函数为较为常见的普通函数, 后面 10 个函数为组合函数, 其特征和最优值的个数可以在表 4-2 中看出。

表 4-1 测试函数性质

F1 (Five-Uneven-Peak Trap)	F6 (Shubert with 2D)	F11 (Composition Function 1 with 2D)	F16 (Composition Function 3 with 5D)
F2 (Equal Maxima)	F7 (Vincent with 2D)	F12 (Composition Function 2 with 2D)	F17 (Composition Function 4 with 5D)
F3 (Uneven Decreasing Maxima)	F8 (Shubert with 3D)	F13 (Composition Function 3 with 2D)	F18 (Composition Function 3 with 10D)
F4 (Himmelblau)	F9 (Vincent with 3D)	F14 (Composition Function 3 with 3D)	F19 (Composition Function 4 with 10D)
F5 (Six-Hump Camel Back)	F10 (Modified Rastrigin)	F15 (Composition Function 4 with 3D)	F20 (Composition Function 4 with 20D)

4.4.1 性能衡量指标

在评价算法求解多模态优化问题时, 需要知道算法一次运行所得的最优值, 将简单介绍 Li 提出的方法, 首先将种群中的个体根据目标函数进行排序, 在此之后, 选择第一个个体 \hat{x}' . 如果函数误差值 $(f(\hat{x}^*) - f(\hat{x}'))$ 小于 ε , \hat{x}^* 代表最优值, ε 代表精确水平, 将 \hat{x}' 保进档案集 S 并从种群中将其删除。此时将种群中的第一个为 \hat{x}' . 如果存在 $(f(\hat{x}^*) - f(\hat{x})) \leq \varepsilon$ 以及与档案集中所有的个体的欧几里得距离小于预定的小生境半径 r , 将其加入档案集并删除掉。以上过程反复执行直到种群中没有任何的个体。档案集的大小就为算法所搜寻到最优值个数。在本文中, 设置了 5 个不同的精确度水平: $\varepsilon = 1.0E-01, 1.0E-02, 1.0E-03, 1.0E-04$, 和 $1.0E-05$ 。

根据文献, 使用峰值比率(PR)和成功比率(SR)作为评价指标, PR 代表了算法所找到的平均数目的个体除以最优值数目, SR 表示找到所有的最优值的个数的次数除以运行的次数。

表 4-2 种群和函数评价次数设置

Test Function	MaxFEs	N
F1-F5	5.0E+04	80
F6	2.0E+05	100
F7	2.0E+05	300
F8-F9	4.0E+05	300
F10	2.0E+05	100
F11-F13	2.0E+05	200
F14-F20	4.0E+05	200

在本文中,算法执行每个测试函数 50 次独立运行。此外,函数最大评价次数(FES),和对于每一个测试函数所设置的种群规模在表 4-3 中。差分进化算子的变异因子和交叉控制参数分别为 0.5 和 0.7。方程(4-2)的缩放因子被设置为:

$$\eta = 40D(CurrentFES/MaxFES)^3 \quad (4-11)$$

4.4.2 MMMOP 算法结果

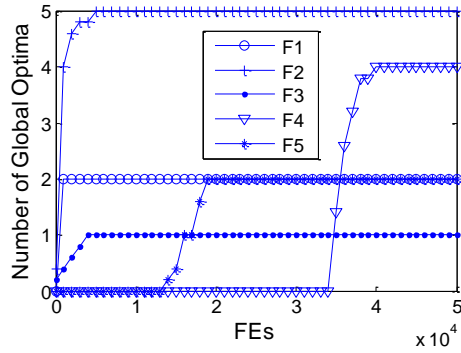
在 20 个测试函数上, MOMMOP 在五个不同的精度等级上进行了测试,并给出了 PR 和 SR 值,表。除了测试函数 F9,在前 10 个测试函数在所有水平的精确性上 MOMMOP 达到 100%的 PR 值和 100% SR 值,这意味着 MOMMOP 有能力求解这些测试函数的多个最优值。F9 涉及 216 个最优解,这些最优解之间的距离差异显著。因此,这是一个对于进化算法非常具有挑战性的函数,很难一次运行找到其所有的解。注意,在函数 F9 上的前面三个精度水平上 ($\varepsilon = 1.0E-01, 1.0E-02$, 和 $1.0E-03$), MOMMOP 达到了 100%PR 值和 100% 的 SR 值,而在其它在其他两个精度等级 PR 值超过 0.97,这意味着所有个体距离最优解非常靠近。而后面的 10 个组合函数更为复杂。对于这些组合的函数, MOMMOP 一致收敛于多个最优解,除了 F20 函数。我们也可以从成功的运行次数观察。特别是,在第一个精度等级上 MOMMOP 能够找到所有 F11 的最优解($\varepsilon = 1.0E-01$)。

为考察进化过程中最优解的数量的变化,图 4-6 给出了 50 次运行的每代所发现的最值数量的平均结果图,清楚地表明除 F20 函数 MOMMOP 可以发现所有测试函数的多个最优值,而且在进化的过程中,这些最优值并没有减少。另外从图 4-6 中也可以看出,在进化的早期 MOMMOP 在所有的测试函数上都很少找到最优解解,这是因为在早期选择压非常小, MOMMOP 均匀的搜索空间,从而造成此结果。

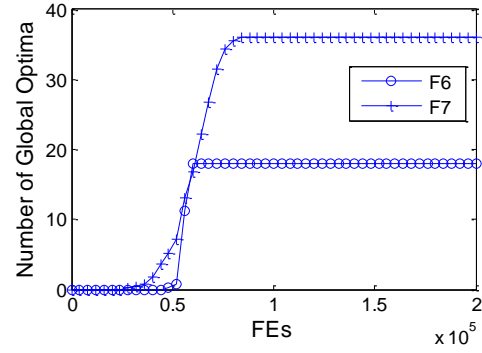
表 4-3 MOMMOP 在不同精度的 PR 值和 SR 值

Accuracy Level ε	F1		F2		F3		F4		F5	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-04	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-05	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Accuracy Level ε	F6		F7		F8		F9		F10	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

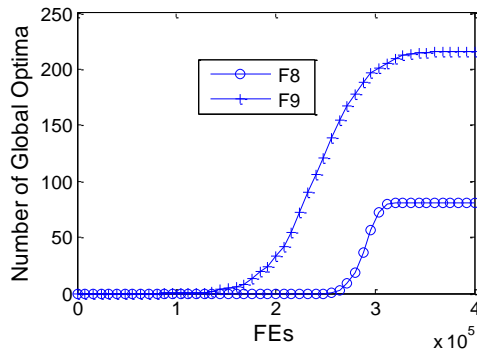
1.0E-03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-04	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.940	1.000	1.000
1.0E-05	1.000	1.000	1.000	1.000	1.000	1.000	0.977	0.100	1.000	1.000
Accuracy Level ε	F11		F12		F13		F14		F15	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	1.000	1.000	0.995	0.960	0.960	0.780	0.783	0.000	0.675	0.000
1.0E-02	0.990	0.940	0.985	0.880	0.933	0.640	0.727	0.000	0.645	0.000
1.0E-03	0.940	0.640	0.965	0.740	0.667	0.000	0.667	0.000	0.623	0.000
1.0E-04	0.717	0.020	0.960	0.700	0.667	0.000	0.667	0.000	0.605	0.000
1.0E-05	0.670	0.000	0.840	0.120	0.667	0.000	0.667	0.000	0.588	0.000
Accuracy Level ε	F16		F17		F18		F19		F20	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	0.667	0.000	0.528	0.000	0.500	0.000	0.250	0.000	0.125	0.000
1.0E-02	0.667	0.000	0.528	0.000	0.500	0.000	0.250	0.000	0.125	0.000
1.0E-03	0.667	0.000	0.528	0.000	0.500	0.000	0.250	0.000	0.125	0.000
1.0E-04	0.667	0.000	0.518	0.000	0.500	0.000	0.250	0.000	0.125	0.000
1.0E-05	0.667	0.000	0.490	0.000	0.500	0.000	0.250	0.000	0.125	0.000



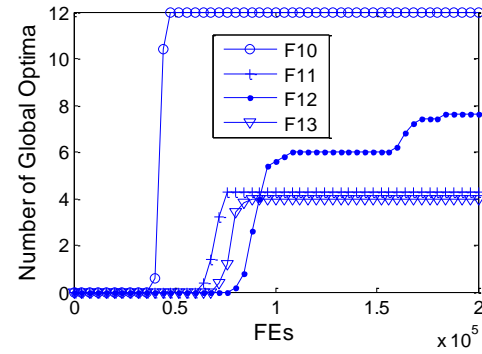
(a) F1- F5



(b) F6- F7



(c) F8- F9



(d) F10- F13

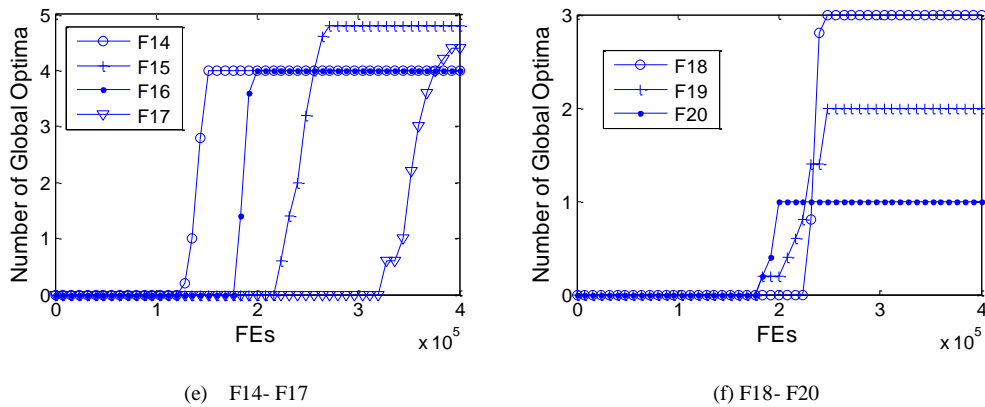


图 4-5 平均数量的最优值进化代数图

4.4.3 MOMMOP 的支配关系比较

在本节中，主要为了 2 个新的比较标准是否有效的求解多模态优化问题和参数设置对算法性能的影响，执行了 CEC2013 的 20 个测试函数的 50 次的独立运行，统计了在 $\varepsilon = 1.0E-04$ 的精确水平下的 PR 值和 SR 值。2 个不同的额外实验用于测试多目标转换函数的比较标准的有效性，在第一个实验中，多模态优化问题只是被转换成一个双目标的优化问题，而不是 D 个双目标优化问题，此双目标优化问题的生成方式是随机的选择一维变量， $x_i (i \in \{1, K, D\})$ ，在每一代中，此种方法，我们称之为 MOMMOP1，第二个实验是 MOMMOP 不用距离度量的比较，此种变化是 MOMMOP2，其中 MOMMOP1, MOMMOP2, 和 MOMMOP 的 PR 值比较在表 4-4 中进行总结。

表 4-4 比较 MOMMOP, MOMMOP1 和 MOMMOP2

Test Function	MOMMOP1	MOMMOP2	MOMMOP
F1	1.000	1.000	1.000
F2	1.000	1.000	1.000
F3	1.000	1.000	1.000
F4	1.000	0.015	1.000
F5	1.000	1.000	1.000
F6	1.000	1.000	1.000
F7	0.853	0.994	1.000
F8	0.422	0.887	1.000
F9	0.509	0.961	1.000
F10	1.000	1.000	1.000
F11	0.680	0.567	0.717
F12	0.865	0.570	0.960
F13	0.667	0.470	0.667

F14	0.667	0.060	0.667
F15	0.478	0.015	0.605
F16	0.667	0.006	0.667
F17	0.125	0.192	0.518
F18	0.180	0.000	0.500
F19	0.125	0.010	0.250
F20	0.125	0.125	0.125

从表 4-4 中可以看出 MOMMOP1 的 PR 值在 9 个函数上剧烈的下降, 可以把 9 个函数分为 3 组, 第一组为 F7-F9, 第二组为 F11, F12, F15, 和 F17-F19, 第一组函数为带有明显的维数遮挡性质, 第二组为组合函数, 对于第一组函数, 本章中讨论过, 随着进化的进行, 一些潜在的最优解会被支配掉, 而失去对峰的搜索能力。因此再一次的证明了本文的 D 个双目标优化问题的支配方案是有效的, 而 F11, F12, F15, 和 F17-F19, 由于这些函数在变量空间中最优值非常相似, 在进化过程中仍然会遇到维数的遮挡问题, 用 MOMMOP1 去求解这 6 个函数, PR 值仍然和第一组函数一样, 急剧下降。另外有趣的是, MOMMOP1 和 MOMMOP 在 F6 和 F10 函数的 PR 值并没有太大的区别, 但是他们仍然带有维数的遮挡性质。分析其原因, 这两个函数较为简单, 在进化时每个峰都保持着相同的进化速度, 维持了进化的平衡, 并且多目标进化算法本身就带有一定的分布度保持机制, 结果是 MOMMOP1 能够定位到所有的最优值

表 4-5 MOMMOP 与 MOMMOP1 和 MOMMOP2 的 Wilcoxon 检验

Algorithm	R+	R-	p-value	$\alpha=0.05$	$\alpha=0.1$
MOMMOP Vs MOMMOP1	162.5	27.5	4.95E-03	Yes	Yes
MOMMOP Vs MOMMOP2	179.5	10.5	1.87E-04	Yes	Yes

进一步确定 MOMMOP 显著性, 执行了使用 Bonferroni-Dunn 方法的 post-hoc 检验的一种 Friedman 检验, Friedman's 测试仍然是使用 KEEL 软件, 从表 4-4 中 MOMMOP 比 MOMMOP2 的在可以看出 13 个的测试函数求解质量更高, 其中有 F4, F7-F9 以及 F11-F19, 更多的是在 6 个测试函数 (F4, F14-F16, F18, 和 F19) 上, MOMMOP2 的 PR 值要小于 0.1, 而在 F18 函数上, MOMMOP2 不能够找到任何的最优值, 并不能够获得与 MOMMOP 一样多的最优值, 另外 MOMMOP2 也不能够提供比较精确的结果, 这是因为 MOMMOP2 在进化过程中丢掉了一些最优值, 此外在进

化过程中分布也是逐渐的丧失。此实验也说明了 MOMMOP 的两个比较标准能够有效的平衡种群的分布，从而搜寻到最优解。

表 4-6 MOMMOP 与 MOMMOP1 和 MOMMOP2 的 Friedman 检验

Algorithm	Ranking
MOMMOP	1.45
MOMMOP1	2.1
MOMMOP2	2.45

为了测试数据的统计有效性，执行了多问题的 Wilcoxon 测试，表 4-5 中展示了 MOMMOP 与 MOMMOP1， MOMMOP2 的 Wilcoxon 检验数据，在所有的测试实例上 MOMMOP 提供了更高的 R+ 值与更少的 R-值，由于所有测试函数的 p 值小于 0.05，因为 MOMMOP 性能明显比 MOMMOP1 和 MOMMOP2 优。从表 4-6 中可以看出，MOMMOP 有着最好的秩，其次是 MOMMOP1。MOMMOP2 最差，这也表明第二个衡量准则比第一个衡量准则在求解性能的提高上更为重要。

为了更进的一步确认 MOMMOP 所提出的 2 个新的支配关系准则的有效性，将 2 维 Vincent 函数一次典型运行的结果分别展示出来，在上文中指出过 2 维 Vincent 函数具有维数的遮挡性质，6 个为一组。3 个算法的初始种群完全一样，都随机的分布于搜索空间。

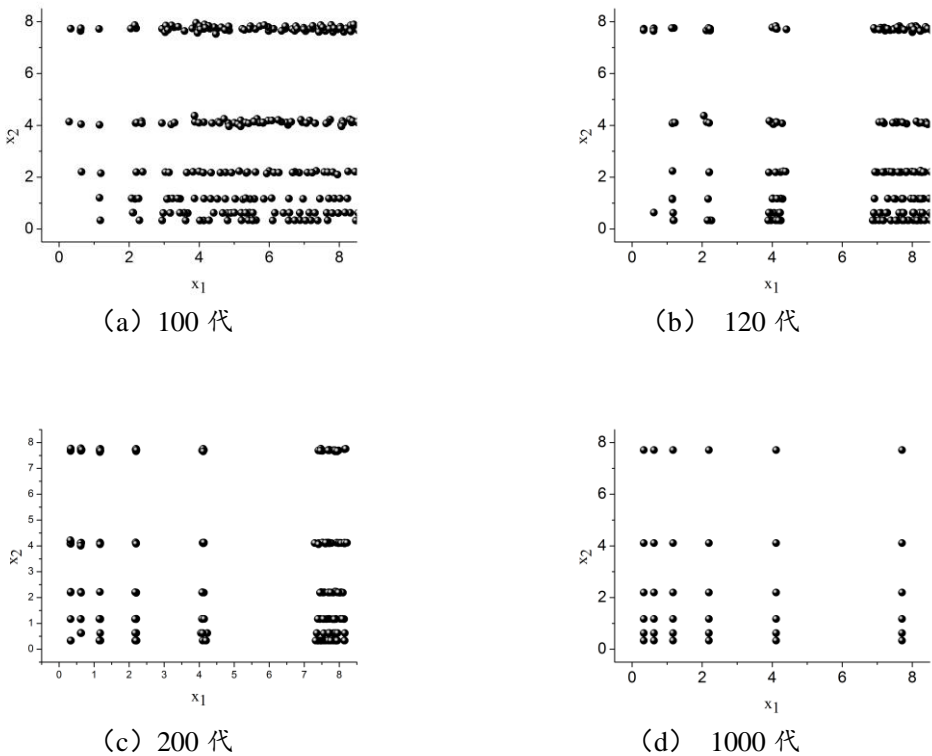


图 4-6 MOMMOP1 的一次典型运行

从图 4-7 中可以看出, MOMMOP1 能够有效地定位 VINCENT 函数的多个解, 一个有趣的现象是在 100 代时, MOMMOP1 已经定位到了 (0,4) 附近的最优值, 但是随着进化的进行, 在 120 代时在此位置, MOMMOP1 丢失了此解, 这是因为维数遮挡的性质存在, 从而导致了在此区域搜索的退化, 但是随着进化的进行, 在 200 代左右, MOMMOP1 又恢复了对此位置的搜索, 分析了原因, 这是因为, MOMMOP1 带有距离信息, 在个体的支配关系比较时, 即使适应值空间被支配, 但是其距离较远 MOMMOP1 仍然不具备支配的可能。

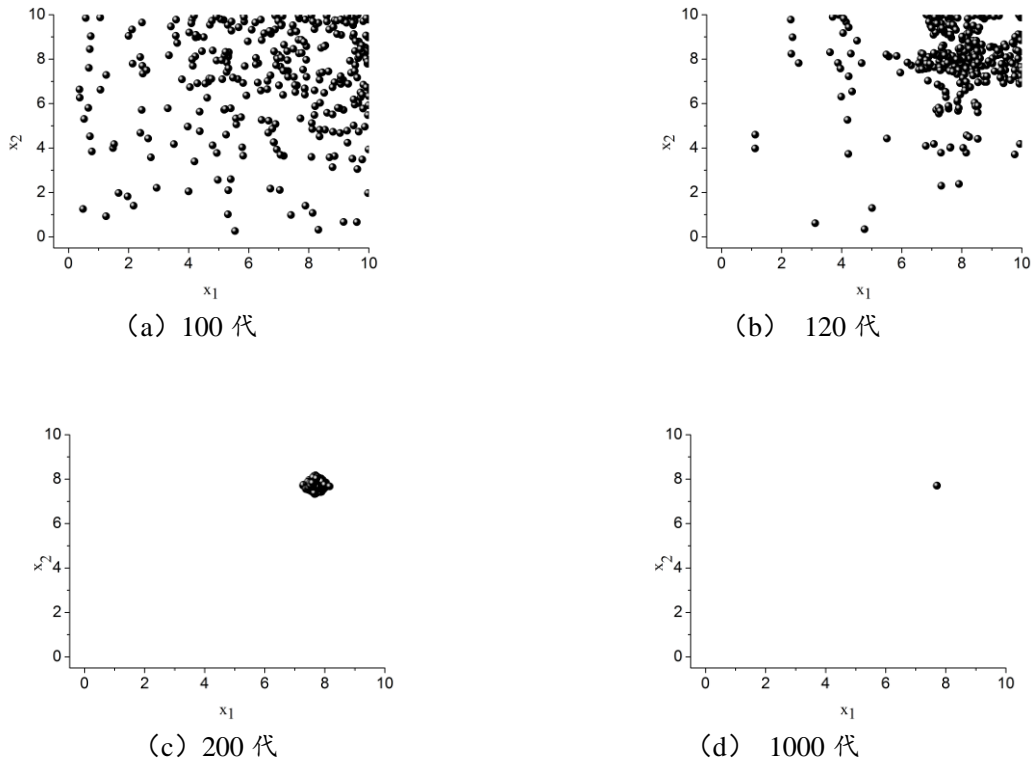


图 4-7 MOMMOP2 的一次典型运行

从图 4-8 中可以看出, MOMMOP2 对 Vincent 函数的多个解定位失败, 最终收敛于一个解。在算法早期, MOMMOP2 均匀的搜索变量空间, 由于 VICENT 函数在空间的右部峰较为平坦, 种群聚集到右上方进行搜索, 随着进化的进行, 由于种群分布度保持机制失败, 导致种群在一个峰上不停的搜索, 最终收敛于一个点。

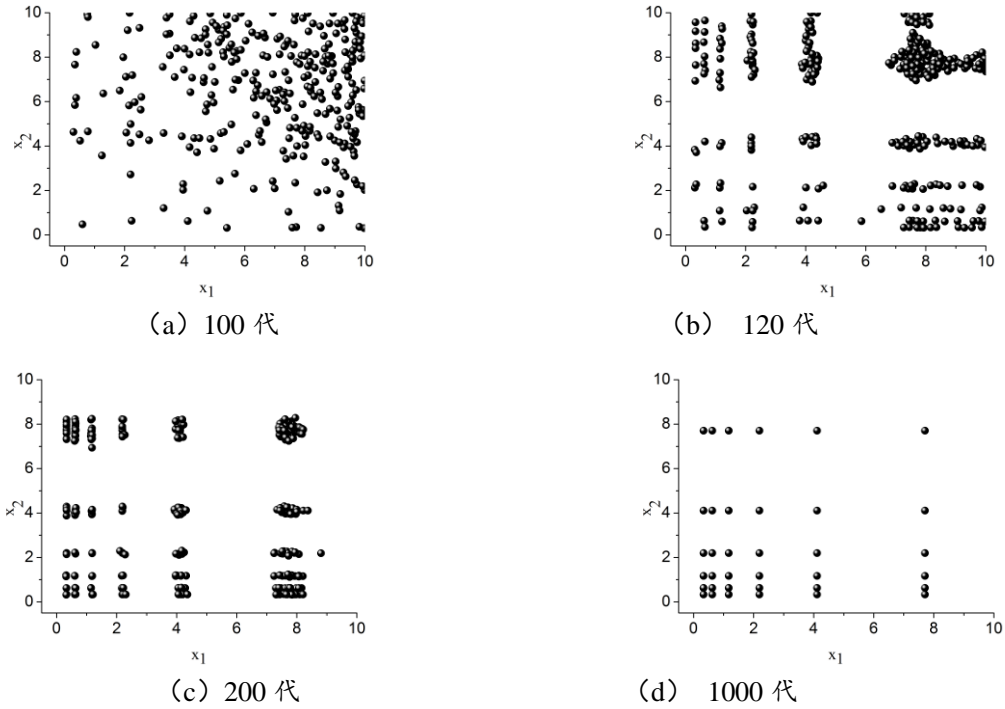


图 4-8 MOMMOP 的一次典型运行

从图 4-9 中可以看出, MOMMOP 能够有效地定位 VINCENT 函数的多个解。在算法早期, MOMMOP 与 MOMMOP2 一样均匀的搜索变量空间, 由于 VICENT 函数在空间的右部峰较为平坦, 种群聚集到右上方进行搜索, 由于采用了距离的比较信息, 阻止算法在在一个峰上往复搜索, 最终定位了函数的多个解。另外也可以看出, 放缩因子在早期, 算法在一个峰上进行搜索时, 种群的选择压较小, 群体在一个峰上保持了较好的分布。

4.4.4 MOMMOP 的参数辨识

在 MOMMOP 中存在着两个参数, 一个是个体比较时的距离参数和放缩因子。对于距离参数本文的设置都为 0.01, 在支配关系的比较实验中, 可以得到 MOMMOP2 在求解多模态优化问题时由于没有距离信息的度量性能较差。而放缩因子调整种群的选择压力, 当放缩因子较大时, 选择压较大, 放缩因子小时, 选择压较小。在此节的实验中主要对于次 2 个参数的设置进行讨论。

当设置距离参数较大时, 不是同一个峰上的个体会被支配, 导致性能下降, 而较小时, 使 MOMMOP 退化到 MOMMOP2, 从而使算法对于多个解的定位失败, 色织了 4 个不同的值: 0.001, 0.005, 0.01, 和 0.05, 分别以 MOMMOP_dp1, MOMMOP_dp2, MOMMOP_dp3, 和 MOMMOP_dp4 代表。这几个距离的设置精度的 $\varepsilon = 1.0E - 04$. PR 值在表中给出。

表 4-7 比较 MOMMOP 不同的距离参数的 PR 值

Test	0.001	0.005	0.01	0.05
Function	(MOMMOP_dp1)	(MOMMOP_dp2)	(MOMMOP_dp3)	(MOMMOP_dp4)
F1	1.000	1.000	1.000	1.000
F2	1.000	1.000	1.000	1.000
F3	1.000	1.000	1.000	0.903
F4	0.305	1.000	1.000	0.975
F5	1.000	1.000	1.000	1.000
F6	1.000	1.000	1.000	0.413
F7	1.000	1.000	1.000	0.690
F8	1.000	1.000	1.000	0.333
F9	0.998	0.999	1.000	0.713
F10	1.000	1.000	1.000	1.000
F11	0.696	0.710	0.717	0.706
F12	0.952	0.948	0.960	0.932
F13	0.667	0.670	0.667	0.667
F14	0.667	0.667	0.667	0.667
F15	0.557	0.640	0.605	0.602
F16	0.667	0.667	0.667	0.667
F17	0.412	0.490	0.518	0.395
F18	0.453	0.500	0.500	0.500
F19	0.167	0.250	0.250	0.255
F20	0.125	0.125	0.125	0.125

表 4-8 比较 MOMMOP 不同的距离参数的

Algorithm	R+	R-	<i>p</i> -value	$\alpha=0.05$	$\alpha=0.1$
MOMMOP_dp3 vs MOMMOP_dp1	171.0	39.0	1.21E-02	Yes	Yes
MOMMOP_dp3 vs MOMMOP_dp2	121.5	88.5	5.26E-01	No	No
MOMMOP_dp3 vs MOMMOP_dp4	162.0	28.0	5.33E-03	Yes	Yes

表 4-9 比较 MOMMOP 不同的距离参数的 Friedman 检验

Algorithm	Ranking
MOMMOP_dp3	2.025
MOMMOP_dp2	2.125
MOMMOP_dp1	2.825
MOMMOP_dp4	3.025

在本文的放缩因子 η 随着进化动态的设置,在 4.2 节的分析中,放缩因子有效的控制平衡搜索和选择压力。当放缩因子较小时种群的收敛精度不够,而较大时,算法将会遇到早熟收敛,设置放缩因子为 10, 30, 40, 50, 和 70.分别以 MOMMOP_sp1, MOMMOP_sp2, MOMMOP_sp3, MOMMOP_sp4, and MOMMOP_sp5 以代表。分别执行了 Wilcoxon 测试和 Friderman 测试。

表 4-10 比较 MOMMOP 不同的放缩因子的 PR 值

Test	10	30	40	50	80
Function	(MOMMOP_sp1)	(MOMMOP_sp2)	(MOMMOP_sp3)	(MOMMOP_sp4)	(MOMMOP_sp5)
F1	1.000	1.000	1.000	1.000	1.000
F2	1.000	1.000	1.000	1.000	1.000
F3	1.000	1.000	1.000	1.000	1.000
F4	0.015	1.000	1.000	1.000	0.970
F5	1.000	1.000	1.000	1.000	0.986
F6	1.000	1.000	1.000	1.000	1.000
F7	1.000	1.000	1.000	1.000	1.000
F8	1.000	1.000	1.000	1.000	1.000
F9	0.983	1.000	1.000	1.000	1.000
F10	1.000	1.000	1.000	1.000	1.000
F11	0.690	0.707	0.717	0.717	0.713
F12	0.845	0.940	0.960	0.935	0.930
F13	0.660	0.657	0.667	0.657	0.645
F14	0.667	0.667	0.667	0.667	0.667
F15	0.462	0.598	0.605	0.605	0.582
F16	0.667	0.667	0.667	0.667	0.633
F17	0.015	0.502	0.518	0.497	0.475
F18	0.500	0.500	0.500	0.500	0.483
F19	0.250	0.250	0.250	0.250	0.185
F20	0.125	0.125	0.125	0.125	0.125

从表 4-10 和表 4-11 中可以看出, MOMMOP_sp3 明显的比 MOMMOP_sp1 和 MOMMOP_sp5 优秀, 另外从表和表明显的可以看出放缩因子在 30-50 之间具有较好的性能, 本文取其中间值都设置为 40。

表 4- 11 MOMMOP 不同的放缩因子的

Algorithm	R+	R-	<i>p</i> -value	$\alpha=0.05$	$\alpha=0.1$
MOMMOP_sp3 vs MOMMOP_sp1	151.0	39.0	2.23E-02	Yes	Yes
MOMMOP_sp3 vs MOMMOP_sp2	137.5	52.5	9.15E-02	No	Yes
MOMMOP_sp3 vs MOMMOP_sp4	122.0	68.0	2.68E-01	No	No
MOMMOP_sp3 vs MOMMOP_sp5	182.5	27.5	3.59E3	Yes	Yes

表 4- 12 MOMMOP 不同的放缩因子的 Friedman 检验

Algorithm	Ranking
MOMMOP_sp3	2.375
MOMMOP_sp4	2.7
MOMMOP_sp2	2.8
MOMMOP_sp1	3.45
MOMMOP_sp5	3.675

4.4.5 MOMMOP 与 CEC2013 的四个方法比较

由 Li 等人组织的 IEEE cec2013 特别会议和竞赛专题, 有四种方法参加了求解多模态优化问题的竞赛, 其分别是: dADE/nrand/1,NVMO,PNSGA-II,MNSGA-II, 改进的 NSGA-II (本文称为 MNSGA-II)。其中在差分进化算法的基本的变异策略的基础上, Epitropakis 整合了 2 个新的机制, 第一个机制是适应性差分进化算法(类似于 JADE), 第二个机制是动态的档案集策略。Molina 提出了小生境变化的网格优化方法, 以 dADE/nrand/1 表示其方法, 此方法使用了清洗技术协同适应性小生境以及外部集保存所找到的当前最好的解。Xu.引入吸引估计遗传算法, 使用探测多模态技术去估计小生境的半径, 并使用所估计的小生境半径形成种群 (本文简称为 NVMO), 另外使用了一个外部集策略去探测不同的峰。另外还有改进版本的改进的 NSGA-II (简称为 MNSGA-II)。以确保公平的比较, 直接从原来的论文中提取着四种方法的实验结果。

值得注意的是,所有的方法实现了 50 个独立运行,并使用相同的测试指标和最大的函数评价次数。在定位阈值 $\varepsilon = 1.0E - 04$ 下 dADE/nrand/1, NVMO, PNSGA-II, MNSGA-II, 和 MOMMOP 这四个方法的 PR 值,将其列在表 4-5 中,其中最优的以黑体列出,从表中可以看出除了 F13 函数和 F18 函数差于 dADE/nrand/1 外,其它的所有函数都要优于 CEC2013 的方法。

表 4-13 MOMMOP 与 CEC2013 的方法比较

Test	dADE/nrand	NVMO	PNSGA-II	MNSGA-II	MOMMOP
F1	1.000	1.000	1.000	0.930	1.000
F2	1.000	1.000	1.000	1.000	1.000
F3	1.000	1.000	1.000	1.000	1.000
F4	1.000	1.000	0.985	0.320	1.000
F5	1.000	1.000	1.000	0.900	1.000
F6	0.984	0.670	0.473	0.001	1.000
F7	0.823	0.901	0.709	0.509	1.000
F8	0.967	0.198	0.275	0.000	1.000
F9	0.431	0.275	0.298	0.140	1.000
F10	1.000	1.000	1.000	0.953	1.000
F11	0.667	0.667	0.680	0.033	0.717
F12	0.740	0.713	0.642	0.010	0.960
F13	0.667	0.667	0.663	0.000	0.667
F14	0.667	0.667	0.663	0.000	0.667
F15	0.627	0.623	0.470	0.000	0.605
F16	0.667	0.653	0.417	0.000	0.667
F17	0.403	0.413	0.300	0.000	0.518
F18	0.633	0.470	0.110	0.000	0.500
F19	0.018	0.130	0.017	0.000	0.250
F20	0.005	0.000	0.000	0.000	0.125

为了测试此 5 个方法的 PR 值得统计有效性,执行了多问题的 Wilcoxon 测试,另外需要指出的是本文所提出的方法的 Wilcoxon 测试使用了 KEEL 软件,在所有的测试统计中可以看出, MOMMOP 提供了更高的 R+ 值与更少的 R-值。P 值都小于 0.05,这证明了 MOMMOP 是显著性有效的。

表 4-14 MOMMOP 与 CEC2013 的四个方法 Wilcoxon 检验

Algorithm	R+	R-	<i>p</i> -value	$\alpha=0.05$	$\alpha=0.1$
MOMMOP vs dADE/nrand/1	147.0	43.0	3.61E-02	Yes	Yes
MOMMOP vs NVMO	182.0	28.0	2.71E-03	Yes	Yes
MOMMOP vs PNSGA-II	185.0	5.0	3.81E-05	Yes	Yes
MOMMOP vs MNSGA-II	208.5	1.5	4.77E-06	Yes	Yes

进一步确定 MOMMOP 和 4 个方法的显著性, 执行了使用 Bonferroni-Dunn 方法的 post-hoc 检验的一种 Friedman 检验, Friedman's 测试仍然是使用 KEEL 软件, 表 4-15 展示了 5 个方法的排序。从表中可以看出 MOMMOP 是最优的, 其次是 dADE/nrand/1。

表 4-15 MOMMOP 与 CEC2013 的四个方法 Friedman 检验

Algorithm	Ranking
MOMMOP	1.75
dADE/nrand/1	2.275
NVMO	2.75
PNSGA-II	3.475
MNSGA-II	4.75

4.4.6 MOMMOP 与经典的单目标进化算法比较

为了展示多模态优化问题的多目标的形式化是有效的, MOMMOP 与四个经典的单目标优化方法: NCDE, NSDE, LIPS, r2pso。对于这四个方法, 收集了这四个作者所提供的代码, 求解 IEEE CEC2013 的测试集, 为了公平的比较, 每个方法都执行了 50 次, 其最大函数评价次数的设置都与表相同, 表 4-16 提供了在精确性水平 $\varepsilon=1.0E-04$ 的 5 个方法的 PR 值和 SR 值。

表 4-16 MOMMOP 与 NCDE, NSDE, LIPS, and r2pso 的比较

Test Function	NCDE	NSDE	LIPS	r2pso	MOMMOP
F1	1.000	1.000	0.000	0.000	1.000
F2	1.000	1.000	1.000	0.992	1.000
F3	1.000	1.000	1.000	1.000	1.000
F4	1.000	0.995	1.000	0.670	1.000
F5	1.000	1.000	1.000	1.000	1.000
F6	0.972	0.993	0.998	0.388	1.000
F7	0.917	0.685	0.527	0.509	1.000

F8	0.062	0.793	0.876	0.000	1.000
F9	0.668	0.384	0.268	0.091	1.000
F10	0.998	0.981	0.987	0.788	1.000
F11	0.667	0.990	0.993	0.667	0.717
F12	0.053	0.990	0.960	0.448	0.960
F13	0.667	0.667	0.770	0.660	0.667
F14	0.667	0.667	0.667	0.003	0.667
F15	0.373	0.497	0.590	0.003	0.605
F16	0.663	0.667	0.667	0.000	0.667
F17	0.250	0.275	0.498	0.000	0.518
F18	0.357	0.547	0.500	0.000	0.500
F19	0.020	0.243	0.250	0.000	0.250
F20	0.000	0.000	0.000	0.000	0.125

使用 KEEL 软件执行了 PR 值和 SR 值多问题的 Wilcoxon 测试和 Friedman 测试, 表 4-17 多问题的 Wilcoxon 测试和 Friedman 测试提供了实验结果在 $\varepsilon = 1.0E - 04$ 精确度上进行了测试。

表 4-17 MOMMOP 与 CEC2013 的四个方法 Wilcoxon 检验

Algorithm	R+	R-	<i>p</i> -value	$\alpha=0.05$	$\alpha=0.1$
MOMMOP vs NCDE	179.5	10.5	1.87E-04	Yes	Yes
MOMMOP vs NSDE	139.5	50.5	7.64E-02	No	Yes
MOMMOP vs LIPS	143.0	47.0	5.46E-02	No	Yes
MOMMOP vs r2pso	208.5	1.5	4.77E-06	Yes	Yes

表 4-18 显示, MOMMOP 提供了更高的 R+ 值与更少的 R- 值在所有的测试实例上, 多问题的 Wilcoxon 测试, 在 $\alpha=0.05$, MOMMOP 比其它 2 个方法显著性优 (MOMMOP vs NCDE, MOMMOP vs r2pso), 而在 $\alpha=0.1$, 多问题的 Wilcoxon 测试 MOMMOP 比所有的方法都要优。这表示在显著性 $\alpha=0.1$ 时, MOMMOP 比其它所有方法都要优。除此之外, 在 Friedman 测试上拥有最好的秩, 其次是 LIPS 方法。

基于以上的比较, 我们可以得出结论, 整体而言, MOMMOP 优于四个多模态优化的方法。

表 4-18 MOMMOP 与 CEC2013 的四个方法 Friedman 检验

Algorithm	Ranking
MOMMOP	1.875
LIPS	2.525
NSDE	2.725
NCDE	3.25
r2pso	4.625

以上的比较结果清晰的展示了 MOMMOP 要优于 4 个经典的单目标优化方法，也同时说明了多目标形式化是有效的，与单目标优化方法比较，多目标的优化进化在求解多模态优化问题时，有 2 点优势：1) 转换为多目标优化问题后，多模态优化问题定位于多目标优化问题的前沿面，从而使多目标进化算法能够一次的进行定位。2) 多目标进化算法的分布度保持技术能够有效的保持多个最优值在进化种群的分布。

4.4.7 MOMMOP 与多目标优化方法的比较

在验证了 MOMMOP 的性能优于 CEC2013 中提出的四种方法和四种先进的基于单目标优化的方法之后，我们进一步将 MOMMOP 与两个先进的基于多目标优化的方法进行比较，这两个方法分别是 MOBiDE 和 BMPGA。值得注意的是，在 CEC2013 的几个方法，中也包含了两个基于多目标优化的方法：PNSGA-II 和 MNSGA-II。

为了比较的公平，因为有 8 个测试函数 (F1-F7,F9) 已经在文献中测试过 MOBiDE 和 BMPGA 的性能。另外此 2 种算法采用了不同的进化算子并且带有参数，所以 MOMMOP 只在这 8 个测试函数上与他们进行比较。我们在其他 2 种算法的实验环境相同的情况下，将 MOMMOP 在这几个测试函数上运行，表 4-7 给出了三种比较方法的 PR 值。为了保证公平有效，MOBiDE 和 BMPGA 的 PR 值直接从文献[21]中获取。利用 KEEL 软件执行 Wilcoxon 和 Friedman 的测试来查看这三种方法的 PR 值的不同。表格 4-19 和表格 4-20 收集了测试结果。

表 4-19 MOMMOP 与多目标进化方法的比较

Test Function	The niche radius r	ε	MOBiDE	BMPGA	MOMMOP
F1	0.5	1E-06	1.000	0.750	1.000
F2	0.01	1E-06	1.000	0.928	1.000
F3	0.01	1E-06	1.000	1.000	1.000
F4	0.5	5E-04	1.000	0.835	1.000
F5	0.5	1E-05	1.000	0.620	1.000
F6	0.2	5E-02	0.967	0.762	1.000
F7	0.2	1E-03	0.983	0.882	1.000
F9	0.1	1E-03	0.814	0.563	1.000

从表格 4-17 可以看出, MOMMOP 在所有情况下都获取到了更高的 R^+ 值。而且, 在比较 MOMMOP 与 BMPGA 时, p 值低于 0.05, 这就表面 MOMMOP 拥有更加可靠的表现。另外, 依据表格 VIII 中的 Friedman 测试, MOMMOP 获取了最高排名, 其次 MOBiDE。

表 4-20 MOMMOP 与多目标方法 Wilcoxon 检验

Algorithm	R^+	R^-	p -value	$\alpha=0.05$	$\alpha=0.1$
MOMMOP vs MOMMOP1	162.5	27.5	4.95E-03	Yes	Yes
MOMMOP vs MOMMOP2	179.5	10.5	1.87E-04	Yes	Yes

表 4- 21 MOMMOP 与多目标方法 Friedman 检验

Algorithm	Ranking
MOMMOP	1.375
MOBiDE	1.75
BMPGA	2.875

通过与 MOBiDE 和 BMPGA 进行比较, MOMMOP 更好的表现可以归因于以下两点: 1) 在 MOMMOP 中, 两个目标是完全意义上的冲突。2) MOMMOP 的最优解在其转换后的双目标优化问题的 Pareto 前沿面上。

4.6 小结

多模态优化问题的求解非常相似于多目标优化问题, 都是需要求取一组而不是一个解。使用多目标进化算法求解多模态优化问题的关键是如何将一个多模态优化问题转换为多目标优化问题, 从而达到适合多目标进化算法求解的目的。在本节中, 主要工作是将多模态优化问题的最优值映射到一个多目标优化问题的前沿面上, 从而可以使用多目标进化算法进行求解。在此过程中, 发现了使用双目标转换技术求解多模态优化问题存在着 2 个新的问题, 分析了原因, 设计了新的支配方式和新的转换方式。为了公平的比较, 在 CEC2013 所提出的测试集上进行了实验, 比较了同类的多目标进化算法, 以及在单目标框架下的进化算法和在 CEC2013 专题讨论会上所提交的四个进化算法, 实验结果证明了本文所提出算法的有效性, 最后为了验证本文所提出的新的支配方式和新的转换方式的有效性, 设计了 3 个不同的实验, 并对比其结果, 充分表明本文所提出的使用多目标进化算法定位多模态优化问题的多个解(MOMMOP)是有效的, 能够有效的求解多模态优化问题。

需要指出的是本章由于将一个多模态优化问题转换为 D 个多目标优化问题, 在每次迭代中, 需要进行非支配排序, 其时间复杂度较大, 为 MN^2 , 而实际应用中, 需

要较快的运行速度，在下一章在本章的转换框架的基础上，提出了一种较快的多目标进化算法求解多模态优化问题。

第五章 基于分维树的多目标进化求解技术

在上一章中对一般的多模态优化问题,提出了一种泛化的转换框架,将其转换为如下的 D 个双目标优化问题,在比较其个体的适应值时使用 D 个多目标问题进行比较,其转换形式如下:

$$\begin{aligned}
 BOP_1 & \begin{cases} \text{minimize } x_1 + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_1 - L_1) \cdot \eta \\ \text{minimize } 1 - x_1 + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_1 - L_1) \cdot \eta \end{cases} \\
 BOP_2 & \begin{cases} \text{minimize } x_2 + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_2 - L_2) \cdot \eta \\ \text{minimize } 1 - x_2 + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_2 - L_2) \cdot \eta \end{cases} \\
 & \vdots \\
 BOP_D & \begin{cases} \text{minimize } x_D + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_D - L_D) \cdot \eta \\ \text{minimize } 1 - x_D + \frac{|f(\vec{x}) - BestOFV|}{|WorstOFV - BestOFV|} \cdot (U_D - L_D) \cdot \eta \end{cases}
 \end{aligned} \tag{5-1}$$

由于将其转换为 D 个双目标的优化问题,其算法的时间复杂度较高,而在实际的应用中需要更快的效率,是否存在一种更快的方式求解多模态优化问题是本章所需要解决的问题。所知道的是在基于 Pareto 支配关系的多目标进化算法中,每一次迭代都需要构造非支配集。其时间的复杂度主要由于非支配排序所决定,Jason 利用分治策略进行非支配排序,但是随着维数的升高,其时间效率反而有所降低,对于非支配集的排序,在目标数目大于 3 维时,在非支配排序的问题上是否存在着平均时间复杂度为 $N \lg N$ 仍然是一个正在争论的问题。本章首先提出了一种双目标的非支配排序树的概念,提出了构造性算法,然后在上一章提出的多目标转换框架的基础上,提出分维支配树的概念,其目的是限制种群中最优值附近的个体数目增长,最后设计了基于分维非支配树的多目标进化算法求解多模态优化问题。本章 3.1 节将介绍双目标的非支配排序树,3.2 节给出基于分维非支配树的多目标进化算法的框架,最后进行了实验。

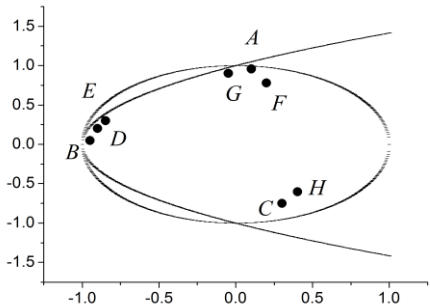
5.1 动机

由公式 5-1,将一个多模态优化问题转换为 D 个双目标优化问题,但是这 D 个双目标优化问题都要进行一次比较,才能确定其是否是进化种群的 Pareto 最优值。但是知道只要一个个体在其一个问题上不被支配,就可以确定其是 Pareto 最优的。因此可以构建更快的方法。

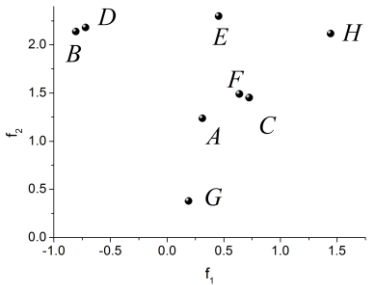
当 2 个最优值映射到其中的一个双目标优化问题的前沿面上时，都对应着同一个点，随着进化的进行，群体搜索到 2 个最优值的附近区域，当 2 个最优值附近区域的个体的搜索趋向于不平衡，其中的一个最优值区域出现了更好的适应值的个体，由于群体的资源是有限的，选择过程由于另一个最优值的附近的个体适应值较小而将其全部排除掉，从而此最优值得不到群体的充分的搜索，导致失败。

表 5-1 转换的实例的目标空间

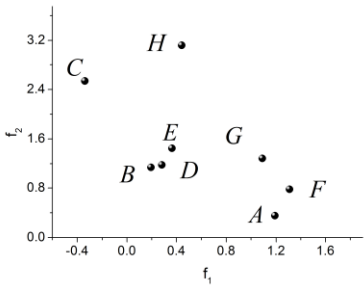
个体	Position		BOP1		BOP2	
	x_1	x_2	f_1	f_2	f_1	f_2
A	0.1	0.96	0.3468	1.2568	1.2068	0.3968
B	-0.95	0.05	-0.8075	2.14	0.1925	1.14
C	0.3	-0.76	1.3548	2.1448	0.2948	3.2048
D	-0.9	0.1	-0.69	2.2	0.41	1.1
E	-0.8	0.05	-0.6025	2.225	0.5475	1.075
F	0.2	0.88	0.0112	1.5712	1.0912	0.491
G	0	0.9	0.38	0.19	1.28	0.29
H	0.4	-0.6	2.19	2.98	2.09	3.08



(a) 实例的变量空间



(b) 第一维决策变量的目标空间



(c) 第二维决策变量的目标空间

图 5-1 实例的位置和支配关系示意图

为了深入分析此过程，并引出本章节的方法，由于非线性方程组在二维空间中更为直观，同时也为了说明非线性方程组具有维数遮挡的性质，下面将给出一个非线性方程组的实例。

$$\begin{cases} e_1(\vec{x}) = 1 - x_0^2 - x_1^2 \\ e_2(\vec{x}) = x_0 - x_1^2 + 1 \end{cases} \quad (5-2)$$

其转换过来的 2 个双目标优化问题是：

$$\begin{aligned} BOP_1 & \begin{cases} \text{minimize } x_1 + \sum_{i=1}^M \|e_1(\vec{x})\| + \|e_2(\vec{x})\| \\ \text{minimize } 1 - x_1 + M * \max(\|e_1(\vec{x})\|, \|e_2(\vec{x})\|) \end{cases} \\ BOP_2 & \begin{cases} \text{minimize } x_2 + \sum_{i=1}^M \|e_1(\vec{x})\| + \|e_2(\vec{x})\| \\ \text{minimize } 1 - x_2 + M * \max(\|e_1(\vec{x})\|, \|e_2(\vec{x})\|) \end{cases} \end{aligned} \quad (5-3)$$

以上的非线性方程组的解析图在图 2 中给出，存在三个最优值(0,-1),(0,1),(-1,0)，其最优值(0,-1),(0,1)在第一维变量上维数遮挡。为模仿一个进化过程，采样一个小的种群，分别为:A、B、C、D、E、F、G、H。可以看到 A、B、C 较靠近 3 个不同的最优点，D、E、F、G、H 离最优值较远。将其以第一维构建定位函数为式(5-3)的 BOP1，其第一层为 B,G。第二层为 D,A。第三层为 E,F，C，最后一层为 H。当选择其一半进入种群时，A,B,G D 由于在第一层和第二层被选择，而 C 被淘汰了，导致算法对于最优值(0,1)的定位失败。主要是因为(0,1)和(0,-1)在第一维的定位函数在其目标空间也是相同的。在其进化过程中，对其中的一个最优值搜索较好，其最优值所在的小生境中其它个体适应值都优于另外一个最优值所在的小生境的个体。如果使用第二维目标同时进行比较就不会有此情况发生，在第 4 章修改了转换函数的目的也就在于此，是否还有其它方式使搜索较好的最优值的个体在二维比较下不支配搜索较差的最优值附近的个体。

下面将介绍一种双目标快速非支配排序的方法，同时提出了一种数据结构：双目标非支配树的概念。

5.2 基于分治法的双目标非支配排序树

多目标进化算法是基于种群迭代的算法，在每一次迭代时，选择出非支配精英集，或者说要构造一次非支配集，因而构造非支配集的效率直接影响算法的运行效率。在上几个章节中所提出的构造非支配集的方法如下：设进化群体为 P，同时设置一个构造集 P'。算法开始时将第一个个体放入构造集 P'中，依次将进化群体 P 中的个体 p(p ∉ P')取出并放入构造集 P'中，(注意：此时放入 P'中的个体只是临时的，因为有可能在随后的比较中被删除)，同时将当前取出的 p 依次与 P'中所有个体进行比较，删除 P'中所有被 p 支配的个体，若个体 p 被 P'中任意一个个体所支配则将 p 从 P'中删除。下面将介绍一种新的基于树结构的双目标支配方法。

本文所提出的双目标非支配排序是基于以下几个定理：

设进化群体 $\text{pop}=\{p_1, p_2, \dots, p_n\}$, $\forall x \in \text{pop}$, 有 $\forall y \in \{p_1, p_2, \dots, p_n\}$, $y \bar{f} x$, x 为 pop 的任意一个非支配解。

性质4.1: 可以将 Pop 分成二部分, 其中 $ND=\{nd_1, nd_2, \dots, nd_k\}$, 其中 $k \leq n$, 和 $D=\{d_1, d_3, \dots, d_m\}$, 使 $\forall d \in \{d_1, d_3, \dots, d_m\}$, 有 $x \phi d$, $\forall nd \in \{nd_1, nd_2, \dots, nd_k\}$, 且 $\neg nd \phi x$, 必有 $d \bar{f} nd$ 。

此性质易证, 如果 $d \phi nd$, 则有 $f^1(d) < f^1(nd)$ 和 $f^2(d) < f^2(nd)$, 由 $\neg(nd \phi x)$ 和非支配定义, 必然有 $f^1(nd) < f^1(x)$ 或者 $f^2(nd) \leq f^2(x)$, 而由 $x \bar{f} d$, 有 $f^1(x) < f^1(d)$ 和 $f^2(x) < f^2(d)$, 矛盾。

此性质说明, 如果一个解是种群中的非支配集, 则可以将种群分为此解的非支配集和此解的支配集。

性质4.2: $ND=\{nd_1, nd_2, \dots, nd_k\}$, 可以将其分为 2 个子集, 其中 $ND1=\{nd_1, nd_2, \dots, nd_{k1}\}$, $ND2=\{nd_1, nd_2, \dots, nd_{k2}\}$, 其中 $k1+k2=k$, 并且 $\forall nd1 \in \{nd_1, nd_2, \dots, nd_{k1}\}$, $\forall nd2 \in \{nd_1, nd_2, \dots, nd_{k2}\}$, 有 $f^1(nd1) \leq f^1(x)$ 和 $f^1(nd2) \geq f^1(x)$ 。存在着 $nd1 \bar{f} nd2$ 。

此性质易证, 假设 $nd1 \phi nd2$, 有 $f^1(nd1) < f^1(nd2)$ 和 $f^2(nd1) < f^2(nd2)$, 由于 $nd1 \bar{f} x$, 有 $f^1(nd1) < f^1(x)$ 或者 $f^2(nd1) < f^2(x)$, 推出 $f^1(nd2) < f^1(x)$ 和 $f^2(nd2) < f^2(x)$, 与 $nd2 \bar{f} x$ 矛盾。

为了更加直观地表示此两种性质, 从图看出, 一个非支配集可以将目标空间分为三个部分, 非支配区域1 (对应着 $ND1$), 非支配区域2 (对应着 $ND2$) 和支配区域 (对应着 D), 其中任何个体在非支配区域1 不支配非支配区域2 中的个体。而支配区域 D 的个体, 不支配对于非支配区域1 ($ND1$ 区域) 和非支配区域2 ($ND2$ 区域) 的个体。因此利用此2个性质, 定义二维支配树结构如下:

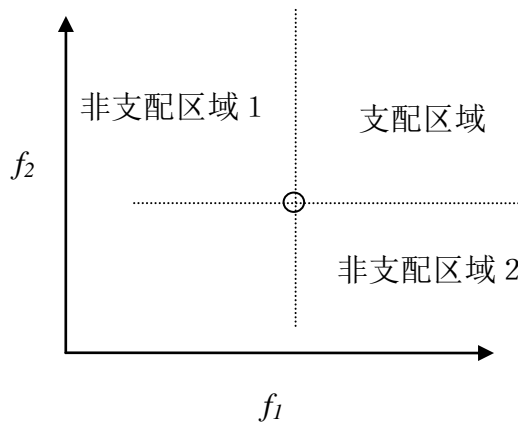


图 5-2 支配区域关系图

二维支配树 (2dimension_dominated_tree) 是包含 n ($n>0$) 个结点的有穷集, 其中:

(1) 每个元素称为结点 (node)，其中每个节点包含了进化中种群个体 (node.individual)，节点的左兄弟指针 (node.individual)，节点的左兄弟指针；

(2) 有一个特定的结点被称为根结点或树根 (root)。

(3) 除根结点之外的其余数据元素被分为 m ($m \geq 0$) 个根节点互不支配的集合 T_1, T_2, \dots, T_{m-1} ，其中每一个集合 T_i ($1 \leq i \leq m$) 本身也是一棵树，被称作原树的子树 (subtree)。并且满足如下2个性质：

1 ind_i 为树 T_i 的根节点， ind_j 为 T_j 的根节点中的个体，并且有 $i < j$ ，那么 $individual_i.f_1 < individual_j.f_1$ 。

2 进化个体 ind 为树 T_i 的根节点， $individual_i$ 为树 T_i 的根节点中的个体，满足 $ind.f_1 < individual_i.f_1$ 。

双目标非支配排序树的子孙节点都被父亲节点所支配，而对应于父亲节点的同一层次上没有支配关系的存在。另外还存在一性质是其左兄弟一定在目标空间中非常靠近其个体的非支配个体。

下面将给出一个面向对象方式的树节点的结构，然后给出算法的流程。首先定义一个节点具有如下形式：

```
TreeNode
{
    Type Date;
    Type Dateset[];
    Point *parent;
    Point *Child;
    TreeNode *leftsibling;
    TreeNode *rightsibling;
}
```

其节点结构如图所示：

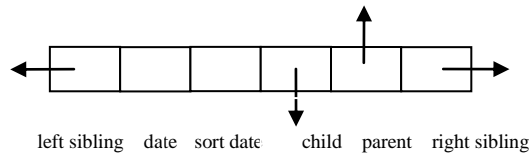


图 5-3 节点结构

在构建每个节点时，需要找到其对应的种群的个体（也就是数据集一个非支配解），在本文中称之为节点的非支配标定，其过程是将节点的数据集中找到一个非支配个体，其算法形式如下：

算法节点的非支配标定

输入：节点TreeNode、节点的数据集TreeNode. dataset

输出：节点的左兄弟节点、右兄弟节点、以及孩子节点

```

if (dataset==空集)
end;
record=first individual of dataset //设置指针指向数据集的第一个节点
For each i Dateset;
    If (record<i)
        Record=i;
    TreeNode.Date= record;
End
Endfor

```

当一个节点的非支配集找到后，利用一个非支配集将目标空间分为3个互不相关的部分，分别对其节点所含的数据集进行分治策略，将其分为做兄弟节点和右兄弟节点以及孩子节点。由于此方法很像菌丝的分裂，在本文称为节点的分裂算法，其算法流程如下所示：

算法双目标树的节点的分裂算法

输入：节点、节点的数据集

输出：节点的左兄弟节点、右兄弟节点、以及孩子节点

```

For each i in Dateset;
if(i<date)//插入孩子节点
    Insert_child.dataset;
end
if(i.fl<date.fl)//插入左兄弟节点
    Insert_leftsibling.dataset;
end
if(i.fl>date.fl)
    Insert_rightsibling.dataset;
end
Delete i from dataset;//从保存的数据集中删除掉个体i;
endfor
end

```

通过节点的非支配标定和分裂，可以构建一颗双目标的非支配排序树，其实现算法通过栈模拟其树的递归过程，算法如下所示：

双目标非支配树的构建算法

输入：种群pop

输出：一颗双目标非支配树

建立根节点root;

其孩子节点root.child;

种群中的所有个体进入rootchild. dataset;

建立一个空栈L;

L. pushback (rootchild);

while (L.size()!=0)

node=L.pop();

node. Find_domi();

node. Divide();

L.pushback (node.leftsibling);

L.pushback (node.rightsibling);

L.pushback (node.child);

End while;

双目标的二维支配树是一种从上至下的构造方法，每一次构造中首先找到一个非支配解，然后将此非支配解所支配的个体加入到孩子节点的数据集中，而非支配个体根据在第一个目标的大小将其放入到兄弟节点中。为了更好的理解双目标非支配排序树的构建算法，将构造一个实例，将构建过程进行说明。下面将给出一个实例介绍其过程：

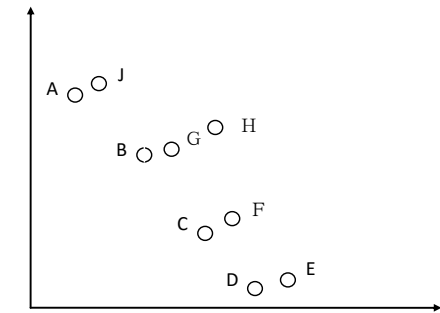


图 5-4 双目标非支配排序的目标空间

如图表示一个双目标的对象空间，在其空间中具有 9 个点，其中的非支配集为 A、

B、C和D，而E、F、G、H、I和J为支配个体。首先根据第一步生成 root 节点，所有个体依次进入孩子节点的数据集中，然后根据双目标非支配树将 root 节点的孩子节点压入到栈 L 中，调用节点的非支配解发现算法，个体 B 为其中的一个非支配个体，将此节点的数据域赋给个体 B，然后调用节点的分裂算法，生成孩子节点，左兄弟节点和右兄弟节点。此时形成的树的形状如图 5-4 所示，可以看出此时只标定了一个节点 B，但是调用节点分裂的方法，其孩子和左右兄弟节点同时生成。依次按照此方法构建一棵二维非支配排序树的结构如图 5-5。在此过程中，根据其树结构所示，第一层非支配集为 ABCD，第二层为 JGFE，第三层为 H。

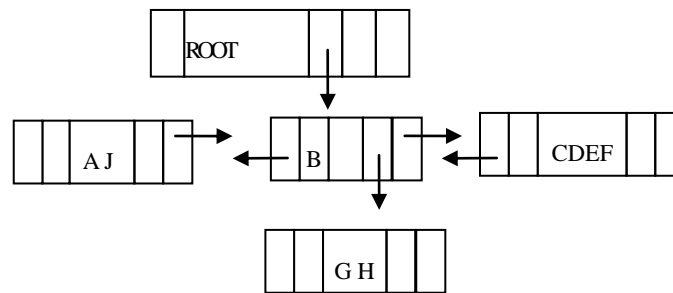


图 5-5 第一次调用节点分裂算法树的结构

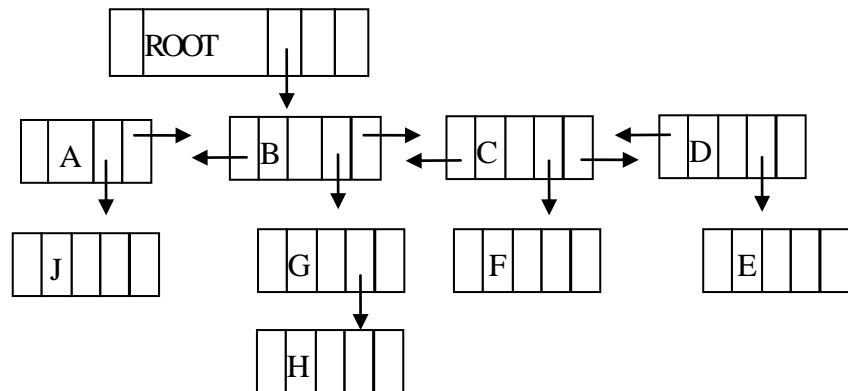


图 5-6 最终所形成的双目标非支配排序树

5.3 分维的非支配排序树

在上一节中，提出了双目标的非支配排序树，在此节中将给出如何将此种数据结构使用到求解多模态优化问题，本文所提出的基于分维非支配排序树的动机，随着进化的进行，通过树的层数限制一个峰的个体在种群中得到较多的资源。其做法是在构建树的过程只比较其一个双目标优化问题，将一个最优值其维数遮挡性质所存在的最

优值附近的区域个体和其周围的个体都加入其子孙结点中。在子孙节点的支配集构造中，不使用此双目标优化问题进行比较，而是用下一个双目标优化问题进行支配树的构建，从而使由于遮挡性质而支配淘汰掉的最优值附近的区域不被父亲节点附近的区域的个体所支配。

在介绍分维的非支配排序树之前，先给出几个定义。

定义 4.1 我们说 \vec{x} 向量在维数 i 上支配向量 \vec{y} ，多模态优化问题在决策变量 x_i 所转换的多目标优化问题 BOP_i ，有 $f_{BOP_i}^1(\vec{x}) < f_{BOP_i}^1(\vec{y})$ 和 $f_{BOP_i}^2(\vec{x}) < f_{BOP_i}^2(\vec{y})$ ，记为 $\vec{x} \overset{i}{f} \vec{y}$ ，如果不存在维数 i 支配，有 $\vec{x} \overset{i}{\bar{f}} \vec{y}$ 。

定义 4.2 我们说向量在维数 i 上所形成的多目标优化问题 BOP_i 上 ξ 支配向量 $\vec{x} \overset{i}{f}_\xi \vec{y}$ ，有 $\vec{x} \overset{i}{f} \vec{y}$ 和 $|x_i - y_i| \leq \xi$ ，其中 ξ 为一预先定义的常数，如果不存在多模态 ξ 维数 i 支配的关系记为多模态 ξ 维数 i 支配，记为 $\vec{x} \overset{i}{\bar{f}}_\xi \vec{y}$ 。

与 MOMMOP 不同，本节个体之间的比较仍然使用某一维转换所形成的双目标优化问题。利用树的层次关系，提升较差适应值附近的个体的适应度，降低搜索较为充分的最优值附近的个体的适应度。如图所示：使用 BOP1 进行非支配排序时一个支配序列是：A、G、C、F 和 H。此时对于最优值 $(0, 1)$ ， $(0, -1)$ 最靠近的个体是 G 和 C。当构建 G 的子孙节点时，使用 BOP2 进行个体的比较。此时个体 A 并不支配个体 C，从而个体 A 和 C 为 G 的孩子节点。此时提升了 C 的适应度，由于 C 的提升，其存在着选择的可能性，从而使个体 $(0, -1)$ 得到较为有效的搜索。

通过以上的例子，可以看出仍然使用双目标的排序方法，但是变换其维数是可以解决所搜索到的最优值，得不到较为充分的搜索，从而淘汰掉的可能。因此本文提出了分维的非支配排序树的定义。

多模态支配：我们说 \vec{x} 向量支配 \vec{y} ，我们有 $\forall i \vec{x} \overset{i}{f} \vec{y}$ ，记为 $\vec{x} \overset{i}{f} \vec{y}$ 。

根据以上定义给出分维支配树的定义

分维支配树 (divide_dimension_tree) 是包含 n ($n > 0$) 个结点的有穷集，其中：

- (1) 每个元素称为结点 (node)；
- (2) 有一个特定的结点被称为根结点或树根 (root)。
- (3) 除根结点之外的其余数据元素被分为 m ($m \geq 0$) 个互不相交的集合 T_1, T_2, \dots, T_{m-1} ，其中每一个集合 T_i ($1 \leq i \leq m$) 本身也是一棵树，被称作原树的子树 (subtree)。

- (4) 其第 i 层节点满足以下性质：

a 如果此节点记为 ind ，那么其左兄弟节点 ind_left ，有 $ind_left \overset{i}{f}_\xi ind$ ，并且有 $ind_left.f_{BOP_i}^1 < ind.f_{BOP_i}^1$ ，其中 BOP_i 为第 i 个优化问题。

b 其右兄弟节点 ind_right ，有 $ind_right.f_{BOP_i}^1 > ind.f_{BOP_i}^1$ 和 $ind_right \overset{i}{\bar{f}}_\xi ind$

c 其任意孩子节点的个体记为 ind_child ，有 $ind f_{\varepsilon} ind_child$ 。

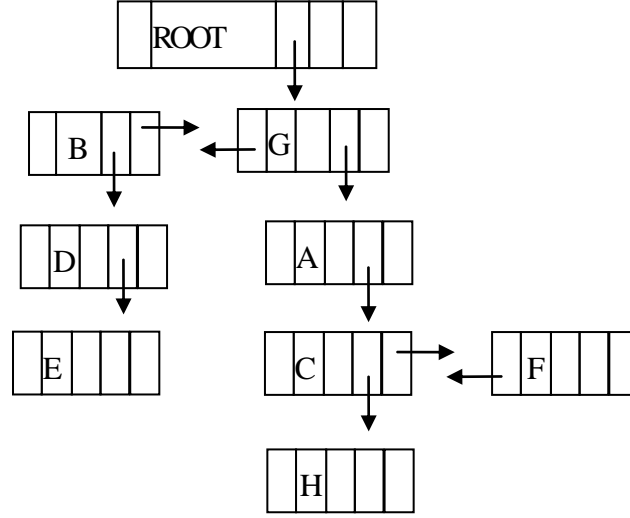


图 5-7 实例的双目标排序树

由以上定义可知：一个节点的孩子节点都被父亲节点所在层次所对应的双目标优化问题所支配，而兄弟节点所在层次所对应的双目标优化问题互不支配，而孩子节点之间变换问题进行比较从而避免的位数的遮挡问题，另外在使用支配关系时，为了避免峰的吸引问题，定义了 ε 支配的关系，将一个较为平坦的峰进入到孩子节点中。

分维非支配排序树的构建算法(construct_dividetree)

输入：种群pop

输出：一颗分维非支配树

```

建立根节点root;
其进化种群进入孩子节点root.child;
种群中的所有个体进入rootchild. dataset;
建立一个空栈 L;
L. pushback (rootchild);
while (L.size() != 0)
    node = L.pop();
    node. Find_domi();
if (dataset == 空集)
    Return;
record = first individual of node. dataset //设置指针指向数据集的第一个节点
For each i in node.dataset Dateset;
if (if  $f_{\varepsilon}^i$  record)
Record = i;
    node. date = record;
end

```

```

endfor
node. Divide();//节点的分裂过程
    For each individual i in Datasets;
        if(i f $\varepsilon$  node.date)//插入孩子节点
            Insert_child.dateset;
        end
        if(i.f1<date.f1)//插入左兄弟节点
            Insert_leftsibling.dateset;
        end
        if(i.f1>date.f1)
            Insert_rightsibling.dateset;
        end
        Delete i from dateset;//从保存的数据集中删除掉个体i;
    endfor

    L.pushback (node.leftsibling) ;
    L.pushback (node.rightsibling) ;
    L.pushback (node.child) ;
End while;
End

```

在算法中为了避免参数 ε 的设置，统一设置为 $\varepsilon = (\text{lowbound} - \text{upbound}) / \text{popsize}$ ，其中 lowbound 为此决策变量的下界， upbound 为上界。由以上算法可以得到图4-1所提到的实例的分维排序树如图5-7所示：

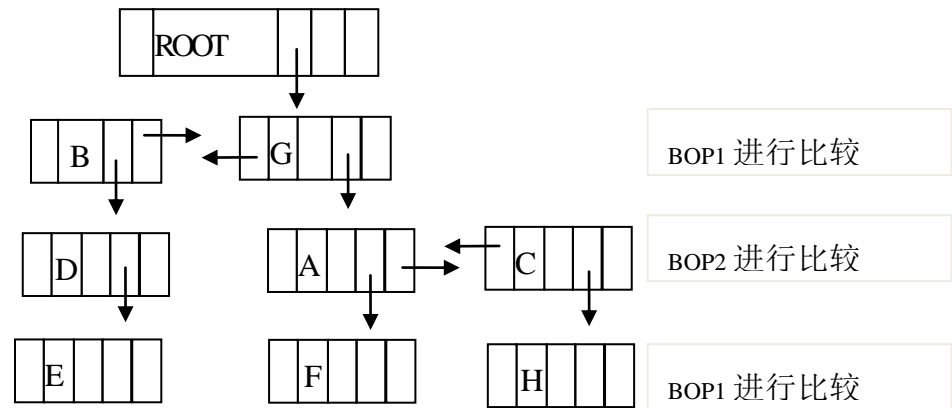


图 5-8 实例的分维多目标排序树

通过 BOP1 所构造的二维支配树的层数数目：第一层的非支配个体为 B 和 G。然后由 BOP2 构建第二层为 D，C 和 A，第三层为 E，F 和 H。

由分维的非支配排序树的层数数目可以得出：第一层的非支配个体为 B 和 G。第

二层为 D, A 和 C, 第三层为 E, F 和 H。

C 的层次数目得到了提升。也就是说使点 G 的子孙或者周围个体不支配定位其它最优值的个体。

基于分维非支配树求解多模态优化问题的主要流程

输入：所要求解问题

输出：多个所要求解问题的最优值

```

G = 0;    // G is the generation number
生成初始种群  $P_0 = \{x_1^I, K, x_N^I\}$ 
Evaluate the multimodal function values of the initial population  $f(x_1^I), K, f(x_N^I)$ ;
Compute BestOFV and WorstOFV;
FES = N;    // FES records the number of fitness evaluations
While FES < MaxFES do    // MaxFES 代表函数最大适应值评价次数
    Implement the crossover and mutation of DE to produce an offspring population  $Q_G$ ;
    Evaluate the multimodal function values of the offspring population  $Q_G$ ;
    FES = FES + N;
    Update BestOFV and WorstOFV;
     $H_G = P_G \cup Q_G$ ;
    Compute the D objective vectors of each individual in  $H_G$ ;
    Construct_dimensiontree ();
    Find  $(N - |P_{G+1}|)$  individuals with least level;
    G = G + 1;
EndWhile

```

5.4 时间复杂度分析

每一个节点的非支配标定, 找到一个非支配集的时间 N, 而节点的分裂过程中, 每个个体都要与节点的非支配标定的个体进行比较一次, 其时间的复杂度为 N。也就是说每一个节点的操作时间为 2N。

当每次节点分裂过程中, 每个个体都是其子孙个体时, 其运算时间为最坏的时间复杂度, 其节点共有 N 个, 每个个体操作 2N 次, 其最坏的时间复杂度为 $O(2DN^2)$ 。

下面来说明其平均时间复杂度, 由于每个个体可以分裂为 3 个互不相关的个体。当此三个互不相关的个体的个数相差不大时, 有以下递归式:

$$T(N) = 3T(N/3) + O(2N) \quad (5-4)$$

因此根据递归式的定义可以得到，算法的平均复杂度为：

$$N \lg N \quad (5-5)$$

5.5 实验

章节所提出的求解多模态的多目标进化算法的目的是保持一定求解精度的前提下提高进化算法的求解速度，因此设置了 2 个实验，第 1 个实验是为了比较算法的运行速度，另一个是在 CEC2013 的测试集上，测试其算法的求解精度。

5.5.1 运行速度比较

文献^[117]指出，Deb 的方法在非支配排序方面所用的时间复杂度与种群中的非支配集比例相关，而相对于本文中所对应的多模态优化问题，最终种群中的非支配集对应着多模态优化问题的最优值个数。因此分别设计了 3 个测试函数如表 5-6 所示，在表 5-6 中的三个测试函数中，最优值分别为 2，36 和 216，当使用种群数目为 300 时分别占种群的 0.7%,10%和 60%，代表着种群中的非支配集个数极少，一般和较多的水平，另外此三个测试函数的维数为可变，执行 3 维到 9 维的多目标优化问题测试。

表 5-2 用于比较运行速度测试函数集

	测试函数	决策变量个数	决策空间	解的个数
FC1	$ x_1^2 + x_2^2 - 1 + x_1 - x_2 + \sum_{i=3}^{nvar} x_i^2$	nvar	$[-1,1]^{nvar}$	2
FC2	$1/2 \sum_{i=1}^2 \sin(10 * \lg(x_i)) + \sum_{i=3}^{nvar} x_i^2$	nvar	$[-0.25,10]$ nvar	36
FC3	$1/3 \sum_{i=1}^3 \sin(10 * \lg(x_i)) + \sum_{i=4}^{nvar} x_i^2$	nvar	$[-0.25,10]$ nvar	216

为了比较多目标方法和单目标方法的时间运行速度，实现了基于拥挤距离的 DE 方法（CDE），另外与 MOMMOP 进行比较。算法独立运行 30 次，取 30 次的平均值。为公平比较这三个算法的运行速度，机器的配置为 win10 的操作系统，都采用 Microsoft Visio 2013 实现，机器的 CPU 为 Intel Core i5-4590 3.3GHZ，内存 8.00GB。其运行结果如图 5-8 到图 5-10。

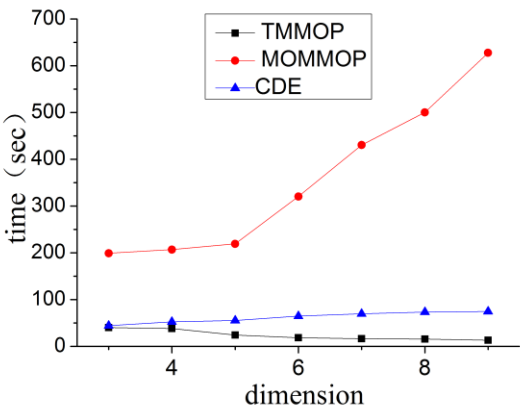


图 5-9 CF1 运行平均时间比较

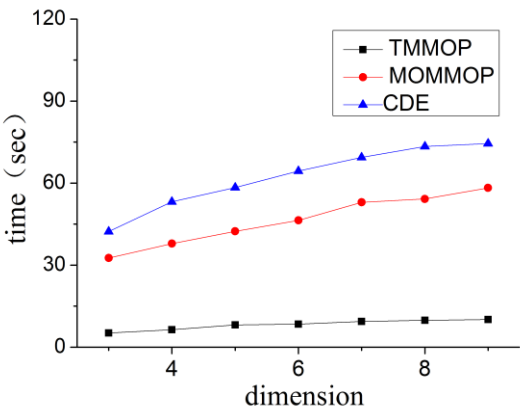


图 5-10 CF1 运行平均时间比较

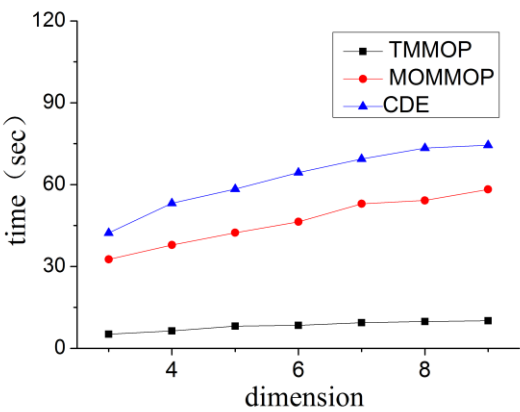


图 5-11 CF1 运行平均时间比较

在 CF1 的运行上，最快的为 TMMOP，而 MOMMOP 最慢，CDE 在两者之间。而在 CF2 函数的运行上，MOMMOP 要快于 CDE，而 TMMOP 的运行时间最快。同样在 CF3 函数的运行上，MOMMOP 的运行要快于 CDE。CDE 在三个函数的运行上，时间变动并无变化，而 MOMMOP 的变化较大，一个有意思的现象是，优化问题的最优值

越多其运行的时间越快，而最优值数目越少，运行越慢，而 TMMOP 也有此现象，但是表现并不敏感，此结果与文献的结果相反。分析其原因如下：当最优值较少时，算法分一层相当于寻找一个最优值，类似于排序方法的插入排序，此时算法无法快速的排除掉一层的个体，从而造成时间的消耗。需要指出的是 TMMOP 在所有的测试函数和维数上都是最快的，并且对于维数并不敏感。

5.5.2 TMMOP 的实验结果

本章节使用相同的评价指标值评价本文所提出的算法，其参数设置包括种群大小，放缩因子与上一章节相同，另外为公平比较使用了 CEC2013 的测试集，运行了 50 次结果。由上表可以看到，对于前面的 10 个组合函数，本章所提出的算法都能够找到全部的最优解，而对于组合函数，除了 F20 函数只能找到一个最优值以外，其他的函数都能够找到最优值，此外与 2 个不同的多目标进化算法求解多模态优化问题相比，本文的算法占有绝对的优势。

表 5-3 TMMOP 在不同精度的 PR 值和 SR 值

Accuracy Level ε	F1		F2		F3		F4		F5	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-04	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-05	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Accuracy Level ε	F6		F7		F8		F9		F10	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-04	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.940	1.000	1.000
1.0E-05	1.000	1.000	1.000	1.000	0.000	0.000	0.477	0.000	1.000	1.000
Accuracy Level ε	F11		F12		F13		F14		F15	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	0.500	0.000	0.800	0.060	0.40	0.080	0.800	0.060	0.40	0.080
1.0E-02	0.500	0.000	0.800	0.060	0.40	0.080	0.800	0.060	0.40	0.080
1.0E-03	0.500	0.000	0.800	0.060	0.40	0.080	0.800	0.060	0.40	0.080
1.0E-04	0.500	0.000	0.8	0.060	0.40	0.080	0.8	0.060	0.40	0.080
1.0E-05	0.500	0.000	0.800	0.060	0.40	0.080	0.800	0.060	0.40	0.080

Accuracy Level ε	F16		F17		F18		F19		F20	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	0.800	0.060	0.40	0.080	0.800	0.000	0.250	0.000	0.125	0.000
1.0E-02	0.800	0.060	0.40	0.080	0.800	0.000	0.250	0.000	0.125	0.000
1.0E-03	0.800	0.060	0.40	0.080	0.800	0.000	0.250	0.000	0.125	0.000
1.0E-04	0.8	0.060	0.40	0.080	0.8	0.000	0.250	0.000	0.125	0.000
1.0E-05	0.800	0.060	0.40	0.080	0.800	0.000	0.250	0.000	0.125	0.000

由上表可以看到, 对于前面的 10 个函数, 本章所提出的算法都能够找到全部的最优解, 而对于组合函数, 除了 F20 函数只能找到一个最优值以外, 其他的函数都能够找到多个最优值。但是需要指出的是本章所提出的算法, 收敛精度与 MOMMOP 相比较差, 这是因为算法在运行过程中, 执行效率较快, 从而损失了算法在局部过程中的开发。为了验证与其它多目标进化算法的有效性能, 为了公平地比较, 采用文献所提供的结果, 此外与 2 个不同的多目标进化算法求解多模态优化问题相比, 本文的算法占有优势。在实验中的 TMMOP 找到了全部的最优值, 而 BMPGA 稍差。

表 5-4 TMMOP 与多目标进化算法的比较实验

Test Function	The niche radius r	ε	MOBiDE	BMPGA	TMMOP
F1	0.5	1E-06	1.000	0.750	1.000
F2	0.01	1E-06	1.000	0.928	1.000
F3	0.01	1E-06	1.000	1.000	1.000
F4	0.5	5E-04	1.000	0.835	1.000
F5	0.5	1E-05	1.000	0.620	1.000
F6	0.2	5E-02	0.967	0.762	1.000
F7	0.2	1E-03	0.983	0.882	1.000
F9	0.1	1E-03	0.814	0.563	1.000

5.6 本章小结

本章为了提高多目标进化算法求解多模态优化问题的计算效率, 使用了一种新的数据结构用于双目标非支配排序的非支配排序树, 此种排序树的对种群进行非支配排序时, 采用分治法使用已经找到的非支配集对目标空间划分成三个互不相交的子集, 其中的 2 个子集不具有支配关系, 而具有支配关系的个体将其划分为孩子节点, 针对每个子集采用此方法进行递归划分, 从而完成了非支配排序的过程。将此方法用于上一章提出的多目标转换方式的非支配排序, 另外定义了一种新的支配方式, 此种支配方式邻域附近的个体加入其孩子节点当中。在深入分析了上一个章节所提出的多目标

转换框架之后,提出了一种新的构建树的方案,在树的每一层采取任意的一个双目标优化问题进行排序。其算法的平均时间复杂度为 $N \lg N$,实验结果充分证明了本文所提出方法求解多模态优化问题时间运行的速度非常快,而对于多个解的定位是有效的。

第六章 总结与展望

6.1 论文工作总结

随着进化计算在生产实践中的应用不断深入,对于问题求解技术的要求越来越高。无使用者在优化过程中往往是求取一个全局最优解或者是满意解,但是随着实践活动的深入,有些全局最优解在实际中由于约束或者鲁棒等条件的限制往往并不适用,在另外一方面来讲,一些全局最优解有助于使用者更好的发现问题的特征,因此需要求出更多的全局最优解和局部极值解备选,从而形成了多模态优化领域。但是传统的数学方法往往只是从一个解进行寻优,无法一次运行定位多个最优解。从生物行为启发而形成的进化算法,由于在算法运行时维持种群的概念,其潜在的天然优势能够利用种群而不是一个个体定位多个最优解。因此进化算法作为对多模态优化问题的求解成为了主流。

本文围绕使用多目标进化算法求解多模态优化问题这一课题,研究了带有先验信息的多模态优化问题-多解的非线性方程系统、一般的多模态优化问题,以及如何设计更高效的多目标优化算法求解多模态优化问题上开展了研究工作。其主要研究工作和创新性成果如下:

本文提出了一种简单而通用的求解非线性方程组系统的多目标转换技术,称为 **MONES**。它把一个非线性方程组转换成为一个双目标优化问题。随后,转化后的问题可以直接使用多目标进化优化算法。讨论了不同的多目标转换函数的前沿面性质的基础上得出线性的最为简单和高效。设计了非线性方程系统的比较平台,由于 **CA** 是第一个且是唯一一个使用多目标进化算法求解多解的非线性方程系统的方法,在所设计的比较平台的基础上,进行了深入的实验比较了它们之间的主要区别,分析了它们的性能。实验结果表明,与 **CA** 方法相比, **MONES** 表现出优越的性能, **MONES** 方法在一次运行中有能力同时定位 **NES** 多个最优解。与基于单目标优化方法的实验比较进一步验证具有有效性。此外,研究帕累托前沿面对算法性能的影响,讨论了不同前沿面的求解性质。

多模态优化问题的求解非常相似于多目标优化问题,都是需要求取一组而不是一个解。在本文中,将多模态优化问题的最优值映射到一个多目标优化问题的前沿面上,从而可以使用多目标进化算法进行求解。在此过程中,发现了使用双目标转换技术求解多模态优化问题存在着 2 个新的问题,分析了原因,设计了新的支配方式和新的转换方式。为了公平的比较,在 **CEC2013** 所提出的测试集上进行了实验,比较了同类的多目标进化算法,以及在单目标框架下的进化算法和在 **CEC2013** 专题讨论会上所提交的四个进化算法,对于实验结果的统计测试证明了本文所提出算法的有效性,最后为了验证本文所提出的新的支配方式和新的转换方式的有效性,设计了 3 个不同的

实验, 并对比其结果, 充分表明本文所提出的使用多目标进化算法定位多模态优化问题的多个解(MOMMOP)是有效的, 能够有效的求解多模态优化问题。

由于将一个多模态优化问题转换为 D 个多目标优化问题, 在每次迭代中, 需要进行非支配排序, 其时间复杂度较大, 而实际应用中, 需要较快的运行速度, 提出了一种分维支配树的数据结构用于多模态优化问题所转换过来的多目标优化框架的非支配排序, 并将其嵌入到多目标进化算法中求解多模态优化问题, 此种方法具有较快的运行速度和运行效率。

6.2 进一步的研究方向

虽然本论文围绕多模态优化问题的多目标求解技术的相关问题进行了一系列研究, 内容涉及非线性方程系统的双目标转换; 多模态优化问题的多目标转换框架, 以及多目标进化算法求解多模态优化问题的计算效率, 提出的算法、模型、机制等取得了一定的效果, 但由于时间和精力有限, 论文还存在很多不足之处需要进一步研究和改进:

1) 非线性方程系统的维数遮挡性质, 作为特殊的多模态优化问题的一种, 在文中并没有涉及到非线性方程系统的维数遮挡性质。本文主要是为了使用多目标进化算法求解多模态优化问题, 因此对于一些特殊性质的非线性方程系统还没有深入研究。此外对于当非线性方程系统无实数解时, 其虚数解如何求取。是否多目标进化算法是否有效求解多个虚数解仍然有效, 是一个值得研究的问题。

(2) 适合求解多模态优化问题的进化范式, 在本文所设计的算法中, 并没有太多的涉及到适合在多模态优化问题的适应值景观的进化范式, 包括粒子群算法和和声搜索技术等。在下一步的研究过程中, 主要考察粒子群算法在多目标优化问题的目标空间中如何搜索不破坏其潜在的小生境, 对于多目标粒子群算法求解多目标优化问题其关键是全局导引粒子和局部导引粒子的选取问题。另外多模态优化问题其本质还是一个单目标优化问题, 使用多目标粒子群算法是否可以利用其单目标的特征以达到进一步减少粒子在飞行过程中的路径以及选取的时间。

(3) 动态的多模态优化问题, 因为实际世界中, 动态是永恒的, 优化问题的拓扑结构并不是静态的, 而是随着时间变化而变化的, 其最优值也不是静态的, 其位置也是动态变化的。而现今大部分的动态问题使用多种群等算法去追踪最优值的位置动态变化情况。而实际问题最优值的个数也是动态变化的, 这类问题对使用进化算法求解是一个挑战, 而使用多目标进化算法求解多模态优化问题时, 并不关心于峰的个数的多少, 其选择压由转换的多目标优化问题的放缩算子动态调节, 非常适合求解最优值个数动态变化的问题, 在下一步的研究中, 将设计动态优化问题的测试平台, 并使用多目标进化算法对其进行求解。

(4) 多模态优化问题的多目标进化算法的学习机制, 大量的文献表明, 进化过程中加入学习机制能够加快算法所获得的全局最优解的精度以及搜索效率, 而在本文所提出的算法中, 并没有使用学习机制, 对于以后使用何种学习机制能够有效地提高多目标进化算法的搜索效率, 这是值得研究的一个方向。而在本文所提出的多目标进化算法求解多模态优化问题使用局部搜索机制将比单目标优化问题上更占优势, 首先由于较优的解在前沿面上, 进行局部搜索可以避免对解空间的重复搜索, 其次可以利用非支配集的概念, 选择较优的个体进行并行的而不是串行的局部搜索, 其在下一步的研究过程中, 将考察局部搜索技术与多目标转换技术协同对多模态优化问题进行求解。

(5) 离散的多模态优化问题, 传统的多模态进化优化框架只是针对连续的多模态优化问题, 而很少涉及到离散的多模态优化问题, 实际问题中, 离散的多模态优化问题更为常见。使用本文所提出多目标进化算法求解离散的多模态优化问题的关键是使用转换函数将其转换为多目标优化问题, 其具体做法是将离散的多模态优化问题转换到连续的空间中, 在连续的空间进行多目标转换, 在函数的评价过程中将离散空间转换到离散空间, 从而完成对离散的多模态优化问题的求解。

参考文献

- [1] 郑金华 多目标进化算法及其应用[M]. 科学出版社. 2007 年 2 月.
- [2] 焦李成等:免疫优化计算、学习与识别[M].科学出版社,2007,7
- [3] 陈国良, 王煦法等. 1999. 遗传算法及其应用[M].人民邮电出版社.
- [4] 王宇平 进化计算的理论和方法[M]. 科学出版社 2011 年 3 月.
- [5] Zbigniew Michalewicz, David B.fogel: 如何求解问题--现代启发式方法[M], 中国水利水电出版社, 2002,9.
- [6] J. Kennedy and R. C. Eberhart, Particle swarm optimization[C] Proc. IEEE Int. Conf. Neural Netw., pp. 1942-1948, 1995
- [7] M. Dorigo & L.M. Gambardella, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computation, 1(1):53-66,1997
- [8] S. Das and P. N. Suganthan, Differential Evolution: A Survey of the State-of-the-art [J], IEEE Trans. on Evolutionary Computation, Vol. 15, No. 1, pp. 4-31, Feb. 2011
- [9] de Castro, L. N.; Von Zuben, F. J. Learning and Optimization Using the Clonal Selection Principle [J]. IEEE Transactions on Evolutionary Computation, 2002: 239 – 251
- [10]李航 遗传算法求解多模态优化问题的研究[D], 天津大学, 管理科学与工程, 2007
- [11]李宝磊; 多元优化过程记忆算法及动静条件下多模态寻优研究[D]. 云南大学, 通信与信息系统, 2015.
- [12]D. Zaharie, Density based clustering with crowding differential evolution[C], Proc. 7th Int. Symp. Symbolic Numeric Algorithms Sci. Comput. pp. 343-352, 2005
- [13]余谦, 王先甲. 基于粒子群优化求解纳什均衡的演化算法[J] 武汉大学学报 2006, 52(1) :332-338
- [14]Q. Ling , G. Wu , Z. Yang and Q. Wang, Robust optimal design under standard crowding differential evolution framework[C], Proc. 6th World Congr. Intell. ControlAutom., pp. 3173-3177, 2006
- [15]P.S. Georgilakis, Market-based transmission expansion planning by improved differential evolution [J], Int. J. Electr. Power EnergySyst., vol. 32, no. 5, pp. 450-456, 2010
- [16]R. I. Lung , T. D. Mihoc and D. Dumitrescu, Nash equilibria detection for

- multi-player games[C], Proc. IEEE CEC, pp. 1-5, 2010
- [17] W. Sheng, S. Swift, L. Zhang and X. Liu, A weighted sum validity function for clustering with a hybrid niching genetic algorithm[J], IEEE Trans. Syst. Man Cybern. B Cybern., vol. 35, no. 6, pp. 1156-1167, 2005
- [18] E. Dilettoso and N. Salerno, A self-adaptive niching genetic algorithm for multimodal optimization of electromagnetic devices [J], IEEE Trans. Magn., vol. 42, pp. 1203-1206, 2006
- [19] E. Perez, F. Herrera and C. Hernandez, Finding multiple solutions in job shop scheduling by niching genetic algorithms [J], J. Intell. Manuf., vol. 14, no. 3-4, pp. 323-339, 2003
- [20] W. Sheng, A. Tucker and X. Liu, Clustering with niching genetic K-means algorithm[C], Proc. Genet. Evol. Comput., pp. 162-173, 2004
- [21] W. Sheng, X. Liu and M. Fairhurst, A niching memetic algorithm for simultaneous clustering and feature selection [J], IEEE Trans. Knowl. Data Eng., vol. 20, no. 7, pp. 868-879, 2008
- [22] J. L. Redondo, J. Fernandez, I. Garcia and P. M. Ortigosa, Solving the multiple competitive facilities location and design problem on the plane [J], Evol. Comput., vol. 17, no. 1, pp. 21-53, 2009
- [23] X. Li, A. Engelbrecht, and M. G. Epitropakis, Benchmark functions for CEC' 2013 special session and competition on niching methods for multimodal function optimization[C], Evolutionary Computation and Machine Learning Group, RMIT University, Melbourne, Australia, Tech. Rep. 2013.
- [24] W. Song, Y. Wang, H.-X. Li, Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization [J], IEEE Transactions on Evolutionary Computation, vol. 19, no. 3, pp. 414-431, 2015.
- [25] J. Yao, N. Kharma, and P. Grogono, Bi-objective multipopulation genetic algorithm for multimodal function optimization [J], IEEE Trans. Evol. Comput., vol. 14, no. 1, pp. 80-102, 2010.
- [26] K. Deb and A. Saha, Multimodal optimization using a bi-objective evolutionary algorithm [J], Evol. Comput. vol. 20, no. 1, pp. 27-62, 2012.
- [27] A. Basak, S. Das, and K. C. Tan, Multimodal optimization using a bi-objective differential evolution algorithm enhanced with mean distance based selection [J], IEEE Trans. Evol. Comput., vol. 17, no. 5, pp. 666-685, 2013.
- [28] H. A. Abbass and R. Sarker, The Pareto differential evolution algorithm [J], Int. J. Artif. Intell. Tools, vol. 11, no. 4, pp. 531-552, 2002

- [29] Y. Wang, H.-X. Li, T. Huang, and L. Li. Differential evolution based on covariance matrix learning and bimodal distribution parameter setting [J]. *Applied Soft Computing*, vol. 18, pp. 232-247, 2014.
- [30] Y. Wang and Z. Cai. Combining multiobjective optimization with differential evolution to solve constrained optimization problems [J]. *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 117-134, 2012.
- [31] B-Y Qu, P N Suganthan, J J Liang, Differential Evolution with Neighborhood Mutation for Multimodal Optimization [J], *IEEE Trans on Evolutionary Computation* vol. 16, no. 5, pp. 601-614, 2012.
- [32] Cavincchio. D. J. Reproductive adaptive plans[C]. *Proc of the ACM 1970 Annual Conf*, 1970. 1-11
- [33] Jong.K.A.De. An analysis of the behavior of a class of genetic adaptive systems [D]. MI: University of Michigan, 1975
- [34] Mahfoud .S.W. Crowding and pre-selection revisited [C]. *Parallel Problem Solving from Nature*, 1992:27-36
- [35] Goldberg. DE, Richardson.J. Genetic algorithms with sharing for multimodal function optimization [C]. *Proc of the 2nd Int'l Conf on Genetic Algorithms*, 1987: 41-49
- [36] Pétrowski A. A clearing procedure as a niching method for genetic algorithms [C]. *Proc of the 1996 IEEE Int'l Conf on Evolutionary Computation*. 1996:798-803
- [37] C. Hocaoglu and A. Sanderson, Planning multiple paths with evolutionary speciation [J], *IEEE Trans. Evol. Comput.*, vol. 5, no. 3, pp. 169-191, 2001
- [38] L. Li and K. Tang, History-Based Topological Speciation for Multimodal Optimization [J], *IEEE Transactions on Evolutionary Computation* 2015 19 (1) pp: 136 -150,
- [39] Spears W M. Simple subpopulation scheme [C]. *Proc of the 1994 Conf on Evolutionary Programming*, 1994: 296-307
- [40] Spears WM. *Evolutionary algorithms: The role of mutation and re-combination* [M]. Berlin: Springer Verlag, 2000: 28-34
- [41] Beasley D, Bull D R, Martin R R. A sequential niche technique for multimodal function optimization[J].*Evolutionary Computation*,1993, 1(2): 101-125
- [42] Mikami S, Wada M, Fogarty T C. Learning to achieve co-operation by temporal-spatial fitness sharing [C]. *Proc of the IEEE Int'l Conf on Evolutionary Computation*, 1995: 803 -807
- [43] X. Li, Niching without niching parameters: Particle swarm optimization using a ring

- topology [J], IEEE Transactions on Evolutionary Computation. 2010,2(3): 150 -169
- [44]O. J. Mengshoel and D. E. Goldberg, The crowding approach to niching in genetic algorithms [J], *Evol. Comput.*, vol. 16, no. 3, pp. 315-354, 2008.
- [45]L. N. de Castro and J. Timmis, An artificial immune network for multimodal function optimization, in *Proc. CEC*, 2002, pp. 699-704.
- [46]J. Barrera and C. A. Coello Coello, A review of particle swarm optimization methods used for multimodal optimization[C], in *Swarm Intelligence for Knowledge-Based Systems*, Springer-Verlag, 2010.
- [47]R. Thomsen, Multimodal optimization using crowding-based differential evolution, in *Proc. CEC*, 2004, pp. 1382-1389.
- [48]O. M. Shir, M. Emmerich, and T. Bäck, Adaptive niche radii and niche shapes approaches for niching with the CMA-ES [J], *Evol. Comput.*, vol. 18, no. 1, pp. 97-126, 2010.
- [49]X. S. Yang, Firefly algorithms for multimodal optimization, in *Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Science*, vol. 5792, Springer-Verlag, Berlin, 2009, pp. 169-178.
- [50]Preuss, Niching the CMA-ES via nearest-better clustering, in *Proc. GECCO*, 2010, pp. 1711-1718.
- [51]N. Hansen and A. Ostermeier, Completely derandomized self-adaptation in evolution strategies [J], *Evolut. Comput.*, vol. 9, no. 2, pp. 159-195, 2001.
- [52]X. Li, Niching without niching parameters: particle swarm optimization using a ring topology [J], *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150-169, 2010.
- [53]R. C. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, in *Proc. 6th Int. Symp. Micromachine Human Sci.*, Nagoya, Japan, 1995, pp. 39-43.
- [54]M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, Finding multiple global optima exploiting differential evolution' s niching capability, in *2011 IEEE Symposium on Differential Evolution (SDE)*, 2011, pp. 1-8.
- [55]B. Y. Qu, P. N. Suganthan, and J. J. Liang, Differential evolution with neighborhood mutation for multimodal optimization [J], *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601-614, 2012.
- [56]B. Y. Qu, J. J. Liang, and P. N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization [J], *Inf. Sci.*, vol. 197, pp. 131-143, 2012.
- [57]B. Y. Qu, P. N. Suganthan, and S. Das, A distance-based locally informed particle swarm model for multimodal optimization [J], *IEEE Trans. Evol. Comput.*, vol. 17, no. 13, pp. 387-402, 2013.

- [58]S. Biswas, S. Kundu, and S. Das, An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution [J], IEEE Trans. Cybern., 2014, in press.
- [59]M. G. Epitropakis, X. Li, and E. K. Burke, A dynamic archive niching differential evolution algorithm for multimodal optimization, in Proc. CEC, 2013, pp. 79-86.
- [60]J. Zhang and A. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput., vol. 13, no. 5, pp. 945-958, 2009.
- [61]D. Molina, A. Puris, R. Bello, and F. Herrera, Variable mesh optimization for the 2013 CEC Special Session Niching Methods for Multimodal Optimization, in Proc. CEC, 2013, pp. 87-94.
- [62]Z. Xu, M. Polojarvi, M. Yamamoto, and M. Furukawa, Attraction basin estimating GA: An adaptive and efficient technique for multimodal optimization, in Proc. CEC, 2013, pp. 333-340.
- [63]Deb Kalyanmoy, Amrit Pratap, Sameer Agrawal and T. Meyrivan. 2002. A Fast and Elitist Multi-objective Genetic Algorithm : NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 6(2):182-197.
- [64]chaffer J. D. 1984. Some experiments in machine learning using vector evaluated genetic algorithms. PhD thesis, Vanderbilt University.
- [65]Schaffer, J. D. 1985. Multi objective optimization with vector evaluated genetic algorithms. In J.Grefenstette, ed., Proceedings of an International Conference on Genetic Algorithms and their Applications, 93-100.
- [66]Corne David W., Joshua D. Knowles and Martin J. Oates. 2000. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization, In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo and Hans-Paul Schwefel (editors), Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 839--848. Springer.
- [67]Corne, D.W., Jerram, N.R., Knowles, J.D., and Oates, M.J. 2001. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), pp. 283-290, Morgan Kaufmann Publishers.
- [68]Zitzler, E. and L. Thiele. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation, 3(4): 257-271.
- [69]Zitzler E., M. Laumanns, and L. Thiele. 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. EUROGEN 2001 -

- Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems.
- [70] Coello Carlos A., et al. 2002. Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic/Plenum Publishers.
- [71] Chen H. W. and Ni-Bin Chang. 1998. Water pollution control in the river basin by fuzzy genetic algorithm-based multiobjective programming modeling [J]. Water Science and Technology, 37(8):55-63.
- [72] Dorn Jason L. and S. Ranji Ranjithan. 2003. Evolutionary Multiobjective Optimization in Watershed Water Quality Management[C], in Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb and Lothar Thiele (editors), Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, pp. 692--706, Springer. Lecture Notes in Computer Science. Volume 2632, Faro, Portugal.
- [73] Erickson Mark, Alex Mayer, and Jeffrey Horn. 2001. The Niche Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems[C]. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, pages 681-695. Springer-Verlag. Lecture Notes in Computer Science No. 1993.
- [74] Reed Patrick M., Barbara S. Minsker and David E. Goldberg. 2001a. A multiobjective approach to cost effective long-term groundwater monitoring using an elitist nondominated sorted genetic algorithm with historical data[J], Journal of Hydroinformatics, Vol. 3, No. 2, pp. 71--89.
- [75] Erickson Mark, Alex Mayer, and Jeffrey Horn. 2001. The Niche Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems[C]. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, pages 681-695. Springer-Verlag. Lecture Notes in Computer Science No. 1993.
- [76] Matthews K.B., K. Buchan, A.R. Sibbald and Susan Craw. 2002. Using soft-systems methods to evaluate the outputs from multi-objective land use planning tools[C], in Proceedings of the 2002 iEMSs International Meeting, Vol. 3, pp. 247--252, International Environmental Modelling and Software Society, Lugano, Switzerland.
- [77] Ascia Giuseppe, Vincenzo Catania and Maurizio Palesi. 2002. A Framework for Design Space Exploration of Parameterized VLSI Systems[C], in Proceedings of the

- 7th Asia and South Pacific Design Automation Conference and the 15th International Conference on VLSI Design, pp. 245--250, IEEE.
- [78]Ramírez-Rosado Ignacio L. and José L. Bernal-Agustín. 2001. Reliability and Cost Optimization for Distribution Networks Expansion Using an Evolutionary Algorithm [J], IEEE Transactions on Power Systems, Vol. 16, No. 1, pp. 111-118.
- [79]Sait Sadiq M., Mohammed Faheemuddin, Mahmood R. Minhas and Syed Sanaullah. 2005. Multiobjective VLSI Cell Placement using Distributed Genetic Algorithm, in Hans-Georg Beyer et al. (editors), 2005 Genetic and Evolutionary Computation Conference (GECCO'2005), pp. 1585--1586, Vol. 2, ACM Press, New York, USA.
- [80]Barba P. Di. 2005. Multiobjective design optimisation: A microeconomics-inspired strategy applied to electromagnetics [J], International Journal of Applied Electromagnetics and Mechanics, Vol. 21, No. 2, pp. 101--117.
- [81]Eklund Neil H. and Mark J. Embrechts. 2001. Determining the Color-Efficiency Pareto Optimal Surface for Filtered Light Sources. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, pages 603-611. Springer-Verlag. Lecture Notes in Computer Science No. 1993.
- [82]Schnier Thorsten, Xin Yao and Pin Liu. 2004. Digital Filter Design Using Multiple Pareto Fronts [J], Soft Computing, Vol. 8, No. 5, pp. 332--343.
- [83]Zhao Shuguang, Licheng Jiao, Jianxun Zhao and Yuping Wang. 2005. Evolutionary Design of Analog Circuits with a Uniform-Design Based Multi-Objective Adaptive Genetic Algorithm, in Jason Lohn, David Gwaltney, Gregory Hornby, Ricardo Zebulum, Didier Keymeulen and Adrian Stoica (editors), 2005 NASA/DoD Conference on Evolvable Hardware, pp. 26--29, IEEE Computer Society Press, Los Alamitos, California.
- [84]Kumar Rajeev, V. Prasanna Krishnan and Kartik S. Santhanakrishnan. 2000. Design of an Optimal Communication Network Using Multiobjective Genetic Optimization, in Proceedings of the IEEE International Conference on Industrial Technology, Vol. 1, pp. 515--520.
- [85]Knowles Joshua and David Corne. 2003. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors [J], IEEE Transactions on
- [86]Jiménez Fernando, Gracia Sánchez, Antonio F. Gómez-Skarmeta, H. Roubos and R. Babuska. 2002. Fuzzy Modeling with Multi-Objective Neuro-Evolutionary Algorithms, in Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics (SMC'02), IEEE.

- [87]ixiang Cheng; Yen, G.G.; Gexiang Zhang. A Many-Objective Evolutionary Algorithm With Enhanced Mating and Environmental Selections[J], Evolutionary Computation, IEEE Transactions on, On page(s): 592 - 605 Volume: 19, Issue: 4, Aug. 2015
- [88]K. Ikeda, H. Kita and S. Kobayashi, Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal? [C], Proc. IEEE Congr. Evol. Comput., pp. 957-962
- [89]H. K. Singh, A. Isaacs and T. Ray, A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems[J], IEEE Trans. Evol. Comput., vol. 15, no. 4, pp. 539-556, 2011
- [90]Z. Cai and Y. Wang A multiobjective optimization-based evolutionary algorithm for constrained optimization[J]. IEEE Transactions on Evolutionary Computation, vol. 10, no. 6, pp. 658-675, 2006.
- [91]Y. Wang, Z. Cai, G. Guo, and Y. Zhou. Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 37, no. 3, pp. 560-575, 2007
- [92]Y. Wang and Z. Cai. Combining multiobjective optimization with differential evolution to solve constrained optimization problems[J]. IEEE Transactions on Evolutionary Computation, vol. 16, no. 1, pp. 117-134, 2012.
- [93]C. G. Broyden, A class of methods for solving nonlinear simultaneous equations[J], Math. Comput., vol. 19, no. 92, pp. 577-593, 1965.
- [94]A. Friedlander, M. A. Gomes-Ruggiero, D. N. Kozakevich, J. M. Martínez, and S. A. Santos, Solving nonlinear systems of equations by means of Quasi-Newton methods with a nonmonotone strategy[J], Optimization Methods and Software, vol. 8, no. 1, pp. 25-51, 1997.
- [95]M. D. González-Lima and F. M. de Oca, A Newton-like method for nonlinear system of equations[J], Numer. Algor., vol. 52, no. 3, pp. 479-506, 2009.
- [96]G. C. V. Ramadas and E. M. G. P. Fernandes, Solving nonlinear equations by a Tabu search strategy[C], in Proceedings of the 11th International Conference on Computational and Mathematical Methods in Science and Engineering, 2011, pp. 1578-1589.
- [97]A. Rovira, M. Valdés, and J. Casanova, A new methodology to solve non-linear equation systems using genetic algorithms. Application to combined cycle gas turbine simulation[J], International Journal for Numerical Methods in Engineering,

- vol. 63, no. 10, pp. 1424-1435, 2005.
- [98] Z. Wu and L. Kang, A fast and elitist parallel evolutionary algorithm for solving systems of non-linear equations[C], in Proc. CEC, 2003, pp. 1026-1028.
- [99] P. S. Mhetre, Genetic algorithm for linear and nonlinear equation[J], International Journal of Advanced Engineering Technology, vol. 3, no. 2, pp. 114-118, 2012.
- [100] A. Noriega, J. L. Cortizo, E. Rodriguez, R. Vijande, and J. M. Sierra, A new method to approximate the field of movements of 1-DOF linkages with lower-pairs [J], Meccanica, vol. 45, no. 5, pp. 681-692, 2010.
- [101] J. Wang, Immune genetic algorithm for solving nonlinear equations[C], in Proceedings of 2011 International Conference on Mechatronic Science, Electric Engineering and Computer, 2011, pp. 2094-2097.
- [102] J. Xie, Y. Zhou, and H. Chen, A novel bat algorithm based on differential operator and Lévy flights trajectory, Computational Intelligence and Neuroscience, vol. 2013,
- [103] J. Wu, Z. Cui, and J. Liu Using hybrid social emotional optimization algorithm with metropolis rule to solve nonlinear equations[C], in 10th IEEE International Conference on Cognitive Informatics & Cognitive Computing, 2011, pp. 405-411.
- [104] M. Abdollahi, A. Isazadeh, and D. Abdollahi, Imperialist competitive algorithm for solving systems of nonlinear equations[J], Computers & Mathematics with Applications, vol. 65, no. 12, pp. 1894-1908, 2013.
- [105] Y. Zhou, J. Liu, and G. Zhao, Leader glowworm swarm optimization algorithm for solving nonlinear equations systems[J], Electrical Review, vol. 88, no. 1, pp. 101-106, 2012.
- [106] R. C. Eberhart and J. Kennedy, A new optimizer using particle swarm theory[C], in Proc. 6th Int. Symp. Micromachine Human Sci., Nagoya, Japan, 1995, pp. 39-43.
- [107] Y. Mo, H. Liu, and Q. Wang, Conjugate direction particle swarm optimization solving systems of nonlinear equations[J], Computers and Mathematics with Applications, vol. 57, no. 11-12, pp. 1877-1882, 2009.
- [108] M. Jaberipour, E. Khorram, and B. Karimi. Particle swarm algorithm for solving systems of nonlinear equations[J], Computers and Mathematics with Applications, vol. 62, no. 2, pp. 566-576, 2011.
- [109] C. A. Voglis, K. E. Parsopoulos, and I. E. Lagaris, Particle swarm optimization with deliberate loss of information[J], Soft Computing, vol. 16, no. 8, pp. 1373-1392, 2012.

- [110] R. Brits, A. P. Engelbrecht, and F. van den Bergh, Solving systems of unconstrained equations using particle swarm optimization, in Proceedings of International Conference on Systems, Man and Cybernetics, 2002, pp. 6-9.
- [111] M. J. Hirsch, P. M. Pardalos, and M. G. C. Resende, Solving systems of nonlinear equations with continuous GRASP, *Nonlinear Analysis: Real World Applications*[J], vol. 10, no. 4, pp. 2000-2006, 2009.
- [112] E. Pourjafari and H. Mojallali, Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering[J], *Swarm and Evolutionary Computation*, vol. 4, pp. 33-43, 2012.
- [113] A. F. Kuri-Morales, Solution of simultaneous non-linear equations using genetic algorithms[J], *WSEAS Transactions on Systems*, vol. 2, no. 1, pp. 44-51, 2003.
- [114] A. Pourrajabian, R. Ebrahimi, M. Mirzaei, and M. Shams, Applying genetic algorithms for solving nonlinear algebraic equations[J], *Appl. Math. Comput.*, vol. 219, no. 24, pp. 11483-11494, 2013.
- [115] C. Grosan and A. Abraham, A new approach for solving nonlinear equation systems[J], *IEEE Transactions on Systems Man and Cybernetics - Part A*, vol. 38, no. 3, pp. 698-714, 2008.
- [116] S. Effati and A. R. Nazemi, A new method for solving a system of the nonlinear equations[J], *Appl. Math. Comput.*, vol. 168, no. 2, pp. 877-894, 2005.
- [117] 郑金华, 蒋浩, 邝达, 史忠植. 擂台赛法则构造多目标 Pareto 最优解集的方法研究[J], *软件学报*, 2007, 18(6): 1287-1297.
- [118] B.-Y. Qu, P. N. Suganthan and S. Das. A distance-based locally informed particle swarm model for multimodal optimization [J], *IEEE Transactions on Evolutionary Computation*. 2013, 17(3): 114-132
- [119] R. M.A. Silva, M.G.C. Resende and P.M.Pardalos. Finding multiple roots box-constrained system of nonlinear equations with a biased random-key genetic algorithm [J]. *J. of Global Optimization*. 2014, 60: 289-306
- [120] Pourjafari, Ebrahim, Mojallali, Hamed. Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering [J], *Swarm and Evolutionary Computation*, 2011, 15(3): 54-62
- [121] Zhigang Ren; Aimin Zhang; Changyun Wen; Zuren Feng. A Scatter Learning Particle Swarm Optimization Algorithm for Multimodal Problems [J], *Cybernetics, IEEE Transactions on*, On page(s): 1127 - 1140 Volume: 44, Issue: 7, July 2014
- [122] Mike Preuss *Multimodal Optimization by Means of Evolutionary Algorithms*

- ISBN: 978-3-319-07406-1.
- [123] L. Qing, W. Gang, Y. Zaiyue, W. Qiuping Crowding clustering genetic algorithm for multimodal function optimization [J]. *Applied Soft Computing*, 8 (2008), pp. 88 - 95
- [124] Ka-Chun Wong, Chun-Ho Wu, Ricky K.P. Mok, Chengbin Peng, Zhaolei Zhang: Evolutionary multimodal optimization using the principle of locality [J]. *Information Sciences*, Volume 194, 1 July 2012, Pages 138-170
- [125] K. A. De Jong, An analysis of the behavior of a class of genetic adaptive systems, 1975
- [126] S. Mahfoud, Niching methods for genetic algorithms[M], 1995, Springer
- [127] G.R. arik, Finding multimodal solutions using restricted tournament selection[C], *Proc. 6th Int. Conf. Genet. Algorithms*, pp. 24-31, 1995
- [128] D. E. Goldberg and J. Richardson, Genetic algorithms with sharing for multimodal function optimization[C], *Proc. 2nd Int. Conf. Genet. Algorithms*, pp. 41-49, 1987
- [129] J.-P. Li , M. E. Balazs , G. T. Parks and P. J. Clarkson, A species conserving genetic algorithm for multimodal function optimization [J], *Evol. Comput.*, vol. 10, no. 3, pp. 207-234, 2002
- [130] D. Beasley, D. R. Bull and R. R. Martin, A sequential niche technique for multimodal function optimization [J], *Evol. Comput.*, vol. 1, no. 2, pp. 101-125, 1993
- [131] W. Sheng, A. Tucker and X. Liu, Clustering with niching genetic K-means algorithm[C], *Proc. Genet. Evol. Comput.* pp. 162-173, 2004
- [132] W. Sheng , X. Liu and M. Fairhurst, A niching memetic algorithm for simultaneous clustering and feature selection [J], *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 7, pp. 868-879, 2008
- [133] J. L. Redondo , J. Fernandez , I. Garcia and P. M. Ortigosa, Solving the multiple competitive facilities location and design problem on the plane [J], *Evol. Comput.*, vol. 17, no. 1, pp. 21-53, 2009
- [134] M. Boughanem and L. Tamine, A study on using genetic niching for query optimization in document retrieval[C], *Proc. Adv. Inform. Retrieval*, pp. 93-100, 2002
- [135] Y. Li, Using niche genetic algorithm to find fuzzy rules[C], *Proc. Int. Symp. Web Inform. Syst. Applicat.*, pp. 64-67, 2009
- [136] 喻寿益, 郭观七. 选择的遗传漂移分析 [J]. *计算机研究与发展*, 2003, 46 (12):

346-351

- [137] 郭观七, 喻寿益. 重组的遗传漂移分析 [J]. 软件学报, 2003, 14(11): 1875-1881
- [138] 郭观七, 喻寿益. 自适应小生态技术的理论分析与加速技术 [J]. 计算机学报, 2003, 5 (6): 753-758
- [139] 李航, 李敏强, 寇纪淞. 遗传算法求解多模态优化问题的动力性 [J]. 自动化学报, 2008, 34 (3): 116-118
- [140] 陆青, 梁昌勇, 杨善林, 张俊岭. 面向多模态函数优化的自适应小生境遗传算法 [J], 模式识别与人工智能, 2009, 22(1): 89-94
- [141] 张贵军等. 基于广义凸下界估计的多模态差分进化算法 [J]. 软件学报, 2013, 24(6): 1177-1195
- [142] 曹先彬, 罗文尖. 基于生态种群竞争模型的协同进化 [J]. 软件学报, 2000, 12(4): 556-562
- [143] 林焰, 郝聚民. 隔离小生态遗传算法研究[J]. 系统工程学报, 2000, 15(1): 86-91

攻读博士学位期间所发表的论文

1 W. Song, Y. Wang, H.-X. Li, Z Cai. Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization, *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 414-431, 2015. (第一作者, SCI 影响因子 3.665, 中科院人工智能类 1 区)

2 Y. Wang, H.-X. Li, G. G. Yen, and W. Song. MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems, *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 830-843, 2015. (通讯作者, SCI 影响因子 3.469, 中科院控制类 1 区)

1 国家自然科学基金重大专项重点项目: 高速公路车辆智能驾驶中的关键科学问题研究, 2009-2012。

2 国家自然科学基金面上项目: 异质进化算法集成研究, 2011.01-2015.12

3 国家自然科学基金面上项目: 面向动态约束优化问题的进化算法: 设计、分析与应用, 2013.01-至今.

致 谢

博士论文完成之际，近六年的研究生生涯行将结束。依稀记得从海南三亚赴中南大学参加博士生入学考试，在飞机上由于气候原因在空中飞行了一天的情景，那时态度是坚决而执着的，进一步深造自己，体验学术精髓，但是心情是充满忐忑的，因为旅程上充满了波折是否也对应着我的学术生涯也是充满波折。在攻博期间，经历了痛苦与彷徨，经过了汗水与欢乐。在未知的世界中探索事物的真理的激情燃烧的岁月里，充实，我无悔于五年攻读博士的决心。回首求学路上的点点滴滴，我的每一步成长离不开老师和亲友们的鞭策与帮助，衷心感谢你们！

感谢我的导师蔡自兴教授。是您为我创造了进一步深造的机会，教育我树立起报国的理想；是您一次次审思与斧正我的论文并督促我反复修改，培养我勤奋、自强和严谨的治学态度，没有您论文是不可能出来；是您不厌其烦地悉心指导与谆谆教诲，让我深刻思考怎样做人、做一个什么样的人。永远记得，当我由于经济困难和身体原因想放弃学业时，您对我说人要保持向上的心情和永不放弃的精神，永远记得工作得不到进展，您找到我说不要放弃，继续坚持，永远记得，您对我说一个专家和学者是对一件事情反复的坐上十年以上，永远记得，您在备课时把我叫到一边，严谨到了讲解每一内容的时间都精确到秒的态度。我所取得的每一点成绩都凝聚着您的心血，您虚怀若谷、厚德载物的人格魅力不断激励我奋勇向前。

感谢我的师兄王勇老师，王勇老师在我攻博期间指导着我，一丝不苟的治学态度以及对科学的热爱、对事业的执着是我终生学习的榜样。忘不了和您一起在探索未知世界中我的彷徨，以及您的执着。我应该说，勇哥，得到您的指导，是我一生的荣幸，希望人生路上以后继续做我的老师，继续指导我。感谢我的硕士生导师郑金华教授，您是我走上科研工作的引路人，您对我说作为一个学者应该报以什么态度。

感谢智能所教导过我、关心过我的各位老师。特别是智能所的唐璘教授、刘丽珏副教授、肖晓明副教授、谢斌副教授、余伶俐副教授、陈白帆老师、谭平老师、李仪老师、邹逸群老师和郭璠老师。你们为人师表的风范令我敬仰，严谨治学的态度令我敬佩，同时你们为我提供了优厚的科研环境，在此对你们表示深深的敬意。

在那些激情燃烧岁月，永远忘不了那些一起奋斗的兄弟们，那些毕业的师兄，师弟在我的人生将不会忘怀，任孝平、彭梦、文莎、向键、李龙、贾光波、牛勇永、黄晶、廖彩明、查建林、龙成辉、刘华南、刘志中、王冰川、徐彪、刘善源、黄春阳，黄佳伟、尹大庆、李佳朋、薛喜辉、刘昊、魏文燕、李沁、雷晓亮、罗舒宁、张财旺和梁照栋，会永远怀念那些风雨兼程的岁月。

最后感谢我的父母，是您养育了我，培养了我。你们用深切的爱给了我战胜一切困难的勇气，你们是我温暖的港湾，无论海上的风浪再大，这里永远张开双臂，期

待着我的归来，儿子无以报答。感谢我的弟弟，在我读博士这几年，风雨无阻的过来看我给我鼓励，我感觉不到我是兄长。