



Y1885992

**Research on Multi-Objective Artificial Physics Optimization Algorithm
and Its Application**

by

WANG Yan

B.E.(Taiyuan University of Technology) 1998

M.S.(Taiyuan University of Science and Technology) 2007

A dissertation submitted in partial satisfaction of the

Requirements for the degree of

Doctor of Science

in

Control Theory and Control Engineering

in the

Graduate School

of

Lanzhou University of Technology

Supervisor

Professor ZENG Jian-chao

April, 2011

兰州理工大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的
研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或
集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均
已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名：王艳

日期：2011年 4 月 20 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、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论文的复印件和电子版，允许论文被查阅和
借阅。本人授权兰州理工大学可以将本学位论文的全部内容编入有关数据
库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同
时授权中国科学技术信息研究所将本学位论文收录到《中国学位论文全文数据库》，
并通过网络向社会公众提供信息服务。

作者签名：王艳

日期：2011年 4 月 20 日

导师签名：

日期：2011年 4 月 20 日

目 录

摘 要.....	I
Abstract.....	III
插图索引.....	VI
附表索引.....	VII
第 1 章 绪论.....	1
1.1 多目标优化的基本概念.....	1
1.1.1 多目标优化问题描述.....	1
1.1.2 基于 Pareto 支配概念的多目标最优解集.....	2
1.2 传统的多目标优化方法.....	4
1.2.1 加权法.....	4
1.2.2 约束法.....	4
1.2.3 目标规划法.....	5
1.3 多目标进化算法.....	6
1.3.1 经典多目标进化算法.....	6
1.3.2 多目标进化算法的关键技术.....	9
1.3.3 多目标进化算法的性能评价.....	11
1.3.4 目前多目标进化算法存在的问题.....	12
1.4 多目标微粒群算法.....	12
1.4.1 多目标微粒群算法分类.....	12
1.4.2 多目标微粒群算法的关键技术.....	16
1.4.3 目前多目标微粒群算法存在的问题.....	17
1.5 本论文的研究意义及研究内容.....	18
1.5.1 论文的研究意义.....	18
1.5.2 论文的研究内容.....	19
1.6 论文的组织结构.....	20
第 2 章 拟态物理学优化算法及多目标拟态物理学优化算法的基本框架.....	23
2.1 拟态物理学优化算法.....	23
2.1.1 APO 算法提出.....	23
2.1.2 标准 APO 算法流程.....	25
2.1.3 影响 APO 算法性能的关键因素.....	26
2.2 应用 APO 算法解决多目标优化问题可行性分析.....	26
2.3 多目标拟态物理学优化算法基本框架.....	28

2.3.1 算法初始化.....	29
2.3.2 非支配解的选取.....	30
2.3.3 个体所受合力的计算.....	30
2.3.4 个体运动更新及 <i>archive</i> 集更新.....	30
2.3.5 MOAPO 算法流程.....	31
2.4 本章小结.....	31
第3章 基于聚集函数法的多目标拟态物理学优化算法.....	33
3.1 引言.....	33
3.2 算法思想与算法流程.....	33
3.2.1 适应值函数设计及算法思想.....	33
3.2.2 算法流程.....	35
3.3 数值实验及结果分析.....	35
3.4 本章小结.....	41
第4章 基于虚拟力排序的多目标拟态物理学优化算法.....	43
4.1 引言.....	43
4.2 算法思想及算法流程.....	43
4.2.1 算法思想.....	44
4.2.2 算法流程.....	45
4.3 算法收敛性证明.....	46
4.3.1 相关收敛性理论.....	46
4.3.2 算法收敛性.....	47
4.4 数值实验及结果分析.....	49
4.5 本章小结.....	53
第5章 无约束多目标拟态物理学优化算法适应值函数研究.....	55
5.1 引言.....	55
5.2 基于序值的质量函数的计算.....	56
5.2.1 个体序值的定义.....	56
5.2.2 邻域拥挤度.....	56
5.2.3 质量函数的计算.....	57
5.3 基于序值的无约束 MOAPO 算法.....	58
5.3.1 算法思想.....	58
5.3.2 算法流程.....	58
5.4 数值实验及结果分析.....	59
5.5 本章小结.....	65

第 6 章 约束多目标拟态物理学优化算法研究.....	67
6.1 引言	67
6.2 基于可行规则法的约束多目标拟态物理学优化算法研究	69
6.2.1 虚拟力减小的约束多目标拟态物理学优化算法.....	69
6.2.2 算法收敛性证明.....	72
6.2.3 实验及结果分析.....	76
6.3 基于约束保持法的 CMOAPO 算法研究	78
6.3.1 基于序值的约束多目标拟态物理学优化算法.....	78
6.3.2 实验及结果分析.....	79
6.4 本章小结	80
第 7 章 基于多目标拟态物理学优化算法的移动机器人路径规划.....	81
7.1 引言	81
7.2 环境描述	83
7.3 移动机器人路径规划建模	85
7.4 基于 CRMOAPO 的移动机器人路径规划算法	86
7.4.1 算法思想及流程.....	86
7.4.2 仿真试验及结果分析.....	87
7.5 基于 VDCMOAPO 的移动机器人路径规划算法	89
7.5.1 算法思想及流程.....	89
7.5.2 仿真实验与结果分析.....	90
7.6 本章小结	92
结论与展望.....	93
参考文献.....	95
致 谢.....	107
附录 A 攻读博士期间发表的学术论文	108

摘 要

无论科学研究还是工程应用,许多领域都涉及到多目标优化问题,寻求高效稳健的多目标优化算法是研究者们长期以来关注的一个重要课题。拟态物理学优化算法是刚刚出现的一种随机搜索算法,该方法从拟态物理角度来模拟生物觅食机制,对单目标全局优化问题进行求解,在算法收敛性、解集多样性及算法鲁棒性等方面具有较好性能。本文将拟态物理学优化算法应用于多目标优化领域,主要围绕多目标拟态物理学优化算法框架、算法的有效性、种群多样性、适应值的选取、多目标拟态物理学优化算法的约束处理等方面进行理论研究,并将其应用于移动机器人的路径规划中,主要研究成果如下:

1. 通过比较拟态物理学优化算法与典型的基于种群的优化算法之间特征的异同,阐述了基于拟态物理学优化算法解决多目标优化问题的可行性,分析了将拟态物理学优化算法应用于多目标优化领域需要解决的关键问题,建立了多目标拟态物理学优化算法的基本框架。在此基础上,融合聚集函数法,提出了一种多目标拟态物理学优化算法,通过仿真测试说明了多目标拟态物理学优化算法的有效性。
2. 从拟态物理学优化算法自身特点出发,分析了个体作用于种群中其他个体的虚拟力之和与种群多样性的关系,提出了一种基于虚拟力排序的多目标拟态物理学优化算法,并基于概率论的基本知识,对所提算法的收敛性进行了理论分析与证明,通过仿真实验说明了算法具有较好性能,特别在解集多样性保持方面,要优于经典多目标进化算法。
3. 分析了多目标拟态物理学优化算法中个体质量与多目标优化问题中个体适应值的关系,从多目标优化问题的特点出发,利用基于 Pareto 的序值概念及基于邻域半径的共享技术,来确定多目标拟态物理学优化算法中的质量函数,提出了一种基于序值的多目标拟态物理学优化算法。通过与经典多目标进化算法及多目标微粒群优化算法在一系列标准的、复杂的多目标优化测试函数上的数值实验,分析了算法的性能特征。
4. 对约束多目标拟态物理学优化算法进行了研究。首先将可行规则法作为约束处理机制,设计了可行个体及不可行个体的质量函数及虚拟力作用规则,在此基础上提出了一种基于虚拟力减小的约束多目标拟态物理学优化算法,有效处理了可行个体进入不可行域的情况,从而较好解决了 Pareto 最优解集位于可行域

及不可行域边界的约束多目标优化问题,同时,利用概率论的基本理论对该算法进行了收敛性分析证明,通过仿真测试说明了该算法的性能。然后利用约束保持法对约束多目标优化问题中的约束进行处理,结合基于序值的拟态物理学优化算法的基本思想,提出了一种基于序值的约束多目标拟态物理学优化算法,通过数值实验,分析了算法性能。

5. 针对具有多个优化目标的移动机器人全局路径规划问题,首先根据障碍物信息建立机器人工作空间模型,将移动机器人路径规划转化为一个连续空间的约束多目标优化问题。其次,分析移动机器人路径规划特点,将路径长度及路径平滑度作为优化目标,构造移动机器人路径规划的适应值函数,建立了基于多目标拟态物理学优化的移动机器人路径规划模型。然后利用基于序值的约束多目标拟态物理学优化算法对移动机器人的路径进行规划。同时,采用 NSGA-II 在同样的工作环境与模型下对移动机器人路径进行了规划,通过仿真实验的对比结果说明了将约束多目标拟态物理学优化算法应用于移动机器人路径规划的可行性与有效性。最后利用基于虚拟力减小的约束多目标拟态物理学优化算法对移动机器人路径进行规划,通过仿真实验分析了基于两种不同约束多目标拟态物理学优化算法的移动机器人路径规划算法的工程应用范围。

关键词: 多目标拟态物理学优化; 多样性; 适应值函数; 质量函数; 虚拟力规则; 约束处理; 移动机器人路径规划

Abstract

There are a lot of multi-objective optimization problems no matter in natural science area or in engineering applications. It is an important subject for researchers to find a multi-objective optimization algorithm with characteristics of efficient and robust. Artificial Physics Optimization (APO) algorithm is a stochastic optimization algorithm proposed recently. It simulates social animals' foraging from the viewpoint of artificial physics to solve global optimization problems with single objective, and it has a good performance in the aspects of convergence, diversity and robust. In this thesis, APO algorithm is applied into multi-objective optimization area and several aspects of the algorithm, such as its framework establishment, its effectiveness, diversity of population, selection of fitness function, constraint handling, path planning of mobile robot, are studied to make it more effective. It mainly contains:

1. Due to the similarities between APO algorithm and the classic population-based optimization algorithms, the feasibility of applying APO algorithm to solve multi-objective optimization problems is discussed detailedly. Synchronously, some key problems need to be solved when applying APO algorithm to multi-objective optimization area are analyzed. Then the framework of multi-objective artificial physics optimization algorithm is constructed. Based on it, a multi-objective artificial physics optimization algorithm is proposed by combining with the idea of aggregating functions method. It is tested on a well-known benchmark suite and the experimental results show that the proposed approach is effective.
2. The relationship between the diversity of population and virtual force exerted on an individual by the other individuals in population is analyzed based on the characteristics of APO algorithm. Then a multi-objective artificial physics optimization algorithm based on virtual force sorting is presented. At the same time, its convergence is analyzed theoretically. Simulation test shows that the proposed approach has a good performance especially with a perfect diversity comparing with those of the classic multi-objective evolutionary algorithms.
3. The relationship between the mass of an individual in multi-objective artificial physics algorithm and the fitness value of the individual in multi-objective optimization problem is analyzed firstly. Then Pareto-based concept of rank and sharing techniques

based on neighbourhood radius are used to construct mass function. Based on it, a multi-objective artificial physics based on rank is brought forward. The performance of the approach is tested on some well-known benchmarks and the results show that the proposed algorithm is competitive, effective and efficient comparing with some popular multi-objective optimization algorithms, such as the classic multi-objective evolutionary algorithm, multi-objective particle swarm optimization algorithm and standard multi-objective artificial physics optimization algorithm.

4. Constraint multi-objective artificial physics optimization algorithms are studied in this thesis. Firstly, feasibility-based method is adopted as constraint handling mechanism. Meanwhile, different mass functions and virtual force rules for feasible individuals and infeasible individuals are constructed, respectively. According to these rules, a constraint multi-objective artificial physics optimization algorithm based on virtual force decreasing is proposed, which deals the case of feasible individuals moving into infeasible area efficiently. As a result of that, the proposed approach can solve the problems with the solutions on the boundary of feasible area and infeasible area well. Synchronously, the convergence of the presented algorithm is analyzed with the basic technique of probability. Simulation test results prove it has a good performance. Then the constraint-preserving method is used as constraint handling mechanism. Combining with the idea of multi-objective artificial physics optimization algorithm based on rank, a constraint multi-objective artificial physics optimization algorithm based on rank is proposed and its performance is analyzed by simulation test.
5. Using multi-objective artificial physics optimization algorithms to deal with mobile robot path planning problems with multiple optimization objectives are studied in this thesis. Firstly, the mobile robot's workspace is built up according to the information of obstacles. Therefor, the problem of mobile robot path planning is tuned into a constraint multi-objective optimization problem in continuous space. Then, the characteristics of mobile robot path planning problems are analyzed and the fitness function of mobile robot path planning is constructed, in which path length and path smoothness are optimization objectives. Thirdly, constraint multi-objective artificial physics optimization based on rank is used to plan mobile robot path. At the same time, the classic multi-objective evolutionary algorithm, NSGA-II, is adopted to plan mobile robot path in the same environment. The comparing results of simulation tests show that it is feasible and effective to use constraint multi-objective artificial physics optimization algorithm to solve the problems of mobile robot path planning. Finally,

constraint multi-objective artificial physics optimization algorithm based on virtual force decreasing is used to plan the path of mobile robot. The feasible engineering domains for each constraint multi-objective artificial physics optimization algorithm are presented by analyzing the results of simulation tests.

Key Words: multi-objective artificial physics optimization algorithm; diversity; fitness function; mass function; virtual force rules; constraint handling; mobile robot path planning

插图索引

图 1.1 MOP 目标空间中个体间的支配关系图	3
图 2.1 MOAPO 算法基本框架	29
图 3.1 Schaffer 1 的 Pareto 前沿	37
图 3.2 Schaffer 2 的 Pareto 前沿	38
图 3.3 ZTD1 的 Pareto 前沿	38
图 3.4 ZTD2 的 Pareto 前沿	38
图 3.5 ZTD3 的 Pareto 前沿	39
图 4.1 VFMOAPO 与比较算法产生的 ZDT1 的 Pareto 前沿	52
图 4.2 VFMOAPO 与比较算法产生的 ZDT2 的 Pareto 前沿	52
图 4.3 VFMOAPO 与比较算法产生的 ZDT3 的 Pareto 前沿	52
图 5.1 各算法对测试函数 ZDT1 的 Pareto 前沿	61
图 5.2 各算法对测试函数 ZDT2 的 Pareto 前沿	62
图 5.3 各算法对测试函数 ZDT3 的 Pareto 前沿	62
图 5.4 各算法对测试函数 ZDT4 的 Pareto 前沿	62
图 5.5 各算法对测试函数 ZDT6 的 Pareto 前沿	63
图 6.1 VDCMOAPO 算法框架	70
图 6.2 VDCMOAPO 测试函数 Binh(2)的约束 Pareto 前沿	77
图 6.3 VDCMOAPO 测试函数 SRN 的约束 Pareto 前沿	77
图 6.4 VDCMOAPO 测试函数 DEB 的约束 Pareto 前沿	77
图 6.5 CRMOAPO 测试函数 Binh(2)的约束 Pareto 前沿	79
图 6.6 CRMOAPO 测试函数 SRN 的约束 Pareto 前沿	80
图 7.1 移动机器人工作环境模型	84
图 7.2 较少障碍物的 CRMOAPO 路径规划结果	88
图 7.3 较多障碍物的 CRMOAPO 路径规划结果	88
图 7.4 较少障碍物的 VDCMOAPO 路径规划结果	91
图 7.5 较多障碍物的 VDCMOAPO 路径规划结果	91

附表索引

表 1.1 经典 MOPSO 算法及其特征	15
表 3.1 基于聚集函数法 MOAPO 算法的测试函数	36
表 3.2 基于聚集函数法 MOAPO 测试函数的 <i>GD</i> 值比较	39
表 3.3 基于聚集函数法 MOAPO 测试函数的 <i>SP</i> 值比较	40
表 4.1 VFMOAPO 及比较算法对测试函数 ZDT1 的 <i>GD</i> 结果	50
表 4.2 VFMOAPO 及比较算法对测试函数 ZDT1 的 <i>SP</i> 结果	50
表 4.3 VFMOAPO 及比较算法对测试函数 ZDT2 的 <i>GD</i> 结果	51
表 4.4 VFMOAPO 及比较算法对测试函数 ZDT2 的 <i>SP</i> 结果	51
表 4.5 VFMOAPO 及比较算法对测试函数 ZDT3 的 <i>GD</i> 结果	51
表 4.6 VFMOAPO 及比较算法对测试函数 ZDT3 的 <i>SP</i> 结果	51
表 5.1 RMOAPO 算法测试函数	60
表 5.2 RMOAPO 测试函数的 <i>GD</i> 值比较	63
表 5.3 RMOAPO 测试函数的 <i>SP</i> 值比较	64
表 6.1 VDCMOAPO 算法测试函数	76
表 7.1 不同环境下 CRMOPAO 路径规划算法指标	87
表 7.2 不同环境下 NSGA-II 路径规划算法指标	89
表 7.3 不同环境下 VDCMOAPO 路径规划算法指标	91

第1章 绪论

现实世界中的很多优化问题是由多个目标组成的,而这些目标之间往往是相互冲突的,这类问题就是所谓的“多目标优化问题”。人们在求解这类问题时,需要同时对多个相互冲突的目标进行优化和权衡,对这类问题的优化结果一般不会是单个的绝对最优解,而是得到一组均衡解,即所谓的 Pareto 最优解。传统的多目标优化算法通常是通过加权方法将多个目标转化成单个目标,利用数学规划的方法进行优化,故而传统的多目标优化算法往往效率较低,并且对于权重或目标给定次序较为敏感^[1,2]。基于种群的优化算法具有隐并行性以及可以在单轮模拟过程中得到多个解等特点,非常适合求解多目标优化问题^[3-5]。自从 20 世纪 80 年代中期,进化算法开始应用于解决多目标优化问题,表现出了强大的求解能力,在许多实际问题求解中获得了成功,这极大地提高了人们在多目标优化领域的认知水平及解决实际问题的能力^[6-8]。但进化算法模拟生物进化机制,收敛速度较慢,通过单一的多目标进化算法得到一组分布较为均匀的 Pareto 最优解,往往需要较大的迭代次数,而且许多多目标进化算法对一些参数依赖性较强,而这些参数通常需要根据先验知识预先得到,但许多现实世界中的多目标优化问题的先验知识无法预知^[9,10]。随着基于种群的启发式随机算法的发展,以多目标微粒群优化算法为代表的多目标优化算法受到了许多专家学者的关注,然而微粒群优化算法自身的早熟收敛等特点不可避免地被引入到了多目标微粒群优化算法中,为克服这些不足,其他的一些算法被应用于多目标微粒群优化算法中,从而增加了算法的复杂性。故而需要开发基于其他启发机制的、简单高效的多目标优化算法,为多目标优化问题的解决提供一条实用的新途径。

本章首先介绍多目标优化的基本概念,并回顾了解决多目标优化问题的传统算法及其局限性。然后概述了多目标进化算法的发展状况、关键技术及目前存在的一些问题。接下来对多目标微粒群算法的研究动态及其关键技术进行了介绍,分析了目前该领域内需要解决的问题。最后说明本论文的研究意义、内容及论文的组织结构。

1.1 多目标优化的基本概念

1.1.1 多目标优化问题描述

在实际应用中,人们常常会遇到在给定区域上使多个目标同时尽可能最佳的问

题,例如投资问题,人们一般都希望投入的资金量少、风险最小化、获得的投资收益最大。在大多数情况下,这些目标之间是相互冲突的,在不降低一种目标值的情况下,不能任意提高其他目标值,这类问题一般被称为多目标优化问题(Multi-Objective Optimization Problems, MOPs)^[12,10,11],其中目标分为总目标和子目标,这里所谓的多目标优化是对多个子目标同时实施的最优化。通常在多目标优化领域被广泛采用及普遍接受的多目标优化问题的定义如下^[12]:

定义 1.1 (多目标优化问题) 一般的,一个最小化 MOP 可以被描述为:

$$\begin{aligned} \min \quad & y = f(x) = [f_1(x), f_2(x), \dots, f_k(x)] \\ \text{s.t.} \quad & g(x) = (g_1(x), g_2(x), \dots, g_m(x)) < 0 \\ & h(x) = (h_1(x), h_2(x), \dots, h_p(x)) = 0 \\ \text{where} \quad & x = (x_1, x_2, \dots, x_n) \in X \\ & y = (y_1, y_2, \dots, y_k) \in Y \end{aligned} \quad (1.1)$$

其中, x 是决策向量, 决策空间 $X \subseteq R^n$, y 是目标向量, 目标空间 $Y \subseteq R^k$, $f(x)$ 是目标函数, $g(x) < 0$ 和 $h(x) = 0$ 是约束条件, 约束条件限定了可行解集合。

定义 1.2 (可行解集合) 可行解集合 X_f 被定义为满足约束条件 $g(x)$ 和 $h(x)$ 的决策向量 x 的集合^[13]

$$X_f = \{x \in X \mid g(x) < 0 \cap h(x) = 0\} \quad (1.2)$$

可行解集合 X_f 在目标空间的映射被定义为

$$Y_f = f(X_f) = \bigcup_{x \in X_f} f(x) \quad (1.3)$$

1.1.2 基于 Pareto 支配概念的多目标最优解集

多目标优化不同于单目标优化,其本质是对多个目标的同时优化,而这些被同时优化的子目标之间往往又是相互冲突的,所以对于多目标优化问题一般不存在唯一的全局最优解,而是一组均衡解,通常被称为 Pareto 最优解。在定义 Pareto 最优解之前,首先对个体间的支配关系进行定义。

定义 1.3 (个体间的支配关系) 令 $p, q \in R^n$, 称 p 支配 q ^[9], 如果

$$\begin{aligned} (1) \quad & \forall i \in \{1, \dots, k\}, f_i(p) \leq f_i(q) \\ (2) \quad & \exists l \in \{1, 2, \dots, k\}, f_l(p) < f_l(q) \end{aligned} \quad (1.4)$$

其中 k 为子目标的数量, 此时 p 为非支配的, q 为被支配的, 定义为 $p \succ q$, “ $\succ$ ” 表示支配关系。即对于一个最小化问题而言, 向量 p 的所有子目标函数值都不大于相应的向量 q 的目标函数值, 且至少存在一个子目标, 向量 p 对应的目标函数值严格小于向量 q 相应的目标函数值。这里所定义的支配关系是针对决策空间的, 目标空间中的支配关系与其是一致的。

如图 1.1 所示, 在最小化子目标函数 f_1 和 f_2 的 MOP 中, 点 1、2、3、4 代表四个可行解。其中, 点 4 代表的解支配点 1、2、3 所代表的解, 点 2、3 所代表的解均支配点 1 所代表的解, 而点 2 与点 3 所代表的解, 彼此间不存在优劣关系, 它们之间是非支配的。

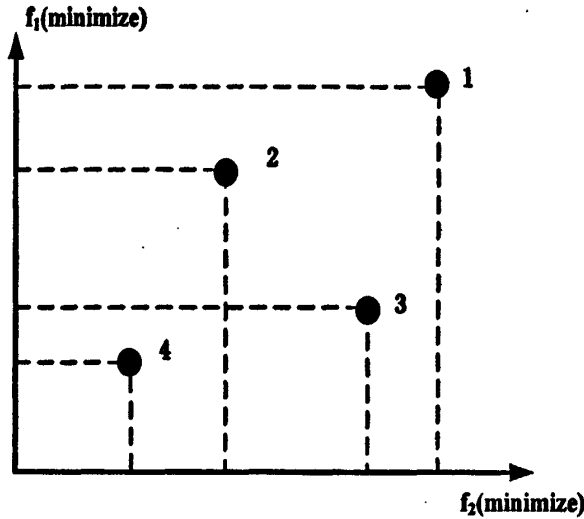


图 1.1 MOP 目标空间中个体间的支配关系图

定义 1.4 (Pareto 最优解) 给定多目标优化问题 (1.1), 决策变量 $\mathbf{x} \in X_f$ 称为 X_f 上的 Pareto 最优解^[1], 当且仅当

$$\neg \mathbf{x}' \in X_f, \mathbf{x}' > \mathbf{x} \quad (1.5)$$

即 $\mathbf{x}$ 是 Pareto 最优解当且仅当 $\mathbf{x}$ 在 X_f 中是非支配的, 也就意味着 $\mathbf{x}$ 的指标性能不可能被提高, 否则会引起其他至少一项指标性能的下降。Pareto 最优解之间是没有优劣差别的, 所有的 Pareto 最优解的集合就构成了 Pareto 最优解集, 其相应的目标向量集合为 Pareto 前沿 (Pareto front)。

定义 1.5 (Pareto 最优解集) 给定多目标优化问题, 其可行决策空间 X_f 中的所有 Pareto 最优解构成了 Pareto 最优解集 P ^[1]

$$P = \{\mathbf{x} \in X_f \mid \neg \exists \mathbf{x}' \in X_f, \mathbf{x}' > \mathbf{x}\} \quad (1.6)$$

该集合相应的目标向量集合 PF 被称为 Pareto 前沿或 Pareto 边界

$$PF = \{f(\mathbf{x}) \mid \mathbf{x} \in P\} \quad (1.7)$$

通常, 多目标优化算法的目标就是找到一个 Pareto 最优解集, 该解集在目标空间中与真实 Pareto 前沿的距离尽可能的小, 并且在目标空间的分布性尽可能的好。因此, 解决多目标优化问题的关键就是找出该问题所有的 Pareto 最优解。

由于多目标优化问题在现实世界中广泛存在, 因此在很久以前研究人员就对此类问题投入了很多精力, 研究出了多种求解方法, 例如加权法, 约束法, 目标规划法等

等, 每类方法都有各自的优点, 但也有各自的应用限制。

1.2 传统的多目标优化方法

传统的多目标优化方法大多是通过聚集的方式将多个子目标转换成单个总目标, 具有代表性的传统多目标优化算法有加权法^[14,15], 约束法^[16], 目标规划法等^[17,18]。

1.2.1 加权法

将多目标优化问题转换成单目标优化问题的最原始的方法就是线性聚合子目标函数, 也就是我们通常所说的加权法。这类方法的基本思想是为每个子目标函数分配一个权重, 并将各子目标函数通过求和的形式组合成一个总目标函数, 从而将多目标优化问题转化为单目标优化问题来进行求解。即^[14]

$$\begin{aligned} \min \quad & y = f(x) = \omega_1 * f_1(x) + \omega_2 * f_2(x) + \dots + \omega_k * f_k(x) \\ \text{sub.} \quad & x \in X_f \end{aligned} \quad (1.8)$$

其中, ω_i 称为权重, 并且不失一般性, 可以将其正则化为 $\sum_{i=1}^k \omega_i = 1$, 通过设置不同权重求解上述优化问题, 如果所有权重取正值, 则得到一些 Pareto 最优解。权重可以理解决策者对目标偏好的标量化的表示, 也可以认为是目标与目标之间的相对价值或重要性的表示。这类方法的优点是直接、简单, 并且对于 Pareto 最优前沿特性为凸的多目标优化问题, 该类方法可保证找到 Pareto 最优解集中的所有解^[19]。缺点是当决策空间通过非线性函数映射到目标空间时, 均匀分布的加权系数并不能保证同样得到均匀分布的解集, 另外, 使用不同的加权系数也不能保证找到不同的 Pareto 最优解。特别的, 这类方法最大的缺点是无论如何调节权重系数, 都无法搜索到非凸均衡曲面上的所有 Pareto 最优解^[20]。

1.2.2 约束法

针对加权法无法求解目标空间中凹区域的问题, Haimes 于 1971 年提出了 ϵ 约束法 (ϵ -Constraint Method)^[16]。该方法的基本思想就是将需要优化的 k 个子目标中的 $k-1$ 个子目标看作是约束条件, 将剩下的那个子目标 (一般是最重要的或者是决策者最偏好的目标, 也可以任意选择) 作为单目标优化问题中的目标函数, 如公式 (1.9) 所示^[16]:

$$\begin{aligned} \min \quad & y = f(x) = f_h(x) \\ \text{sub.} \quad & g_i(x) \geq \epsilon_i \quad (1 \leq i \leq k, i \neq h) \\ & x \in X_f \end{aligned} \quad (1.9)$$

其中, ε_i 为给定的原目标函数的边界值, 在优化过程中, ε_i 往往取不同的值, 这样就可以得到较多数量的 Pareto 最优解。研究表明^[1,2], 约束法在均衡曲线的非凸部分一般情况都可以找到最优解。但该方法在很大程度上依赖于 ε_i 的取值, 若 ε_i 选择不适宜, 则可能会导致无法得到 Pareto 最优解, 这就需要决策者对目标函数有比较清楚的认识, 事先能够确定 ε_i 的取值范围, 但是当目标函数比较复杂时, 这样的要求就很难达到了。另外, 这类方法中, 无论 ε_i 取值如何, 目标空间的可行域一般都会被缩小, 这样往往使得一些本来就是最优解的个体成为不可行个体^[20]。

1.2.3 目标规划法

目标规划法(Goal Programming)是由 Charnes 等人于 1955 年提出的, 直到 1961 年 Charnes 和 Cooper 才将这一概念用来解决多目标优化问题^[8]。该方法的基本思想如下:

首先决策者为每一个目标设置一个理想目标值, 即所谓的目标函数期望值。然后把这期望值当作原问题的附加约束条件, 将原始的多目标优化问题进行修改, 再对修改后的问题进行优化, 找到能达到或最接近期望值的解, 也就是最小化各目标函数值与其理想值之间的绝对偏差。实现各目标绝对偏差的最小化方法有很多, 包括加权法、极大极小法、字典法等等。这种方法的最简单模型如下:

$$\min \sum_{i=1}^k (f_i(\mathbf{x}) - T_i), \quad \mathbf{x} \in X_f \quad (1.10)$$

其中 T_i 为决策者给第 i 个目标函数 $f_i(\mathbf{x})$ 设置的预定理想目标值。这类方法的不足主要体现在各目标函数的理想值在优化之前就必须决定, 所以决策者事先清楚各目标的理想值, 该类方法是非常有效的, 否则在优化开始的阶段, 由于缺乏对搜索空间的形状以及其他特征的充分认识, 使得各目标函数理想值的确定具有主观性和随意性, 从而导致非常低的求解效率。

通过以上对几种典型的传统多目标优化方法的描述可知, 其特点就是使用了一些单目标优化方法的成熟运行机理, 然而 MOPs 与单目标优化问题的本质不同, 在解决单目标优化问题经常使用的一些启发式优化算法, 在解决 MOPs 时, 很难真正被有效利用。总体而言, 传统多目标优化算法存在对 Pareto 前沿形状敏感以及需要具备求解问题应用背景相关的启发性知识的缺点, 并且每次优化过程只能产生一个 Pareto 最优解, 为得到相应的 Pareto 最优解集必须运行多次优化过程, 而每次运行的优化过程都是相互独立的, 故而得到的结果往往不一致, 令决策者难以选择, 只能依靠求解问题相关的应用背景知识, 并且需要很大的时间花费。

进化算法由于其在单轮模拟过程中能够产生多个均衡解, 引起了多目标优化领域专家学者的重视^[21-24]。20 世纪 80 年代中期, 进化算法被引入到多目标优化领域, 出

现了很多研究成果, 现今多目标进化算法 (Multi-Objective Evolutionary Algorithms, MOEAs) 已成为解决多目标优化的一个有效工具。

1.3 多目标进化算法

早在 1967 年, Roesnberg 就在其博士学位论文中提到, 可用遗传搜索算法来求解多目标优化问题, 但没有给出具体算法。第一个多目标进化算法 VEGA(vector Evaluated Genetic Algorithm)是由 Shaffer 于 1985 年提出^[25], 其本质仍然是加权和算法。在随后的几年里, 多目标进化算法一直未引起进化演化计算领域研究者的重视, 直到 1989 年, Goldberg 在其著作中分析了 VEGA 的局限性, 提出了基于 Pareto 支配关系的序值概念^[26]。其后, 基于此基本思想, 三个经典的 MOEAs (MOGA 1993, NSGA 1994, NPGA 1994) 被开发出来, 引起了演化计算领域学者的兴趣, MOEAs 的研究才逐渐受到关注。

1.3.1 经典多目标进化算法

早期的 MOEAs 主要是将遗传算法引入到多目标优化领域, 具有代表性的算法有 1993 年 Fonseca 等提出的多目标遗传算法 (Multi-objective Genetic Algorithm, MOGA)^[27], 1994 年 Srinivas 等提出的非支配排序遗传算法 (Non-Dominated Sorting Genetic Algorithm, NSGA)^[28], 1994 年 Horn 等提出的小生镜遗传算法 (Niche Pareto Genetic Algorithm, NPGA)^[29], 通常将这些算法称为第一代 MOEAs^[10], 其特点是利用 Pareto 排序概念选择个体, 通过适应度共享机制来保持种群的多样性, 下面对上述三种 MOEAs 的基本思想进行简单介绍。

MOGA 是一个出现较早且具有较好性能的 MOEA。在该算法中, 种群中的所有个体基于 Pareto 最优概念进行排序: 对于当前群体中的每一个个体, 计算支配该个体的其他个体的数量, 然后再加 1 作为该个体的排序数, 采用基于排序数的方法对适应度赋值来实施进化过程中的选择运算。该算法利用自适应的共享适应度技术确定小生镜半径, 并将其用于具有相同序号的个体的比较中, 以此来提高种群多样性。由于共享是在目标空间中进行的, 所以能够得到在 Pareto 边界上均匀分布的 Pareto 解集。但该算法易受共享半径大小 σ 的影响, 并且通过共享适应度机制使用随机采样的方法进行选择, 这将会大大降低算法的收敛速度。

NSGA 是根据 Goldberg 基于 Pareto 关系的序值概念, 对多目标解群体进行逐层分类: 首先, 确定种群中的非支配个体, 为其分配一个很大的虚拟适应度, 为了维持种群多样性, 这些非支配个体用它们的虚拟适应度进行适应度共享, 再将这些非支配个体暂时从种群中“移出”, 即暂时不予考虑。然后, 再从余下的种群个体中确定第

二批非支配个体,并给它们分配一个比前一批的非支配个体共享后的最小适应度还要小一些的虚拟适应度。这一过程一直持续到整个种群都被分配了适应度。该算法的优点是没有对优化的目标个数进行限制,能够产生分布均匀的最优解和多个不同的等效解。该算法的不足主要表现在以下3个方面:计算效率较低,计算复杂度为 $O(MN^3)$ (其中 M 为目标数, N 为种群大小);没有采用精英保留策略来保留进化过程中生成的非劣最优解;涉及到的共享参数 σ 需要预先确定。

NPGA 使用了基于 Pareto 支配关系的锦标赛机制来进行选择运算:从种群中随机选择两个个体,然后从种群中再随机选取一定数量的其他个体作为比较集。如果只有一个个体被比较集中的所有个体支配,而另一个个体至少支配比较集中的一个个体,则后一个个体被选中进入下一代;如果两个个体都支配或被支配于比较集中的所有个体,则通过小生境技术实现共享来决定哪个个体进入下一代,即在每个个体的目标空间邻域 σ 内,共享个体数较少的个体获得进入下一代的机会。该算法不是整个种群进行 Pareto 非劣解的选择,相当于减小了种群规模,所以运算起来速度较快,并且能维持一个较长的种群更新期,在很多代都能得到很好的 Pareto 前沿。然而,该算法中不但选择和调整小生境的半径比较困难,而且还要选择一个较为合适的比较集规模,这样使得该算法在实际应用中变得较为复杂。

在上述第一代 MOEAs 的三个经典算法中,以 MOGA 最为有效,并且性能要优于其他两个多目标遗传算法^[3,4]。在第一代 MOEAs 的发展期间,许多问题不断呈现出来,引起了研究者的重视,使得多目标进化领域的研究趋势发生了巨大变化,以精英解保留机制为特征的第二代 MOEAs 相继被提出。这个时期仍然是对多目标遗传算法进行研究,代表性算法有 Zitzler 和 Thiele 提出的强度 Pareto 算法(Strength Pareto Evolutionary Algorithm, SPEA)及其改进版本 SPEA2^[31,32], Knowles 和 Corne 提出的 Pareto 存档进化策略(Pareto Archived Evolution Strategy, PAES)^[33]及其改进版本——Pareto 包络选择算法(Pareto Envelope-Based Selection Algorithm, PESA)^[34]和 PESA-II^[35], Erichson, Mayer 和 Horn 提出的 NPGA 的改进版本 NPGA2^[36], Coello Coello 和 Pulido 提出的微遗传算法(Micro-Genetic Algorithm, Micro-GA)^[37], Deb 等提出的 NSGA 的改进版本 NSGA-II^[38]。下面对这些算法中最为经典的算法进行简单介绍。

SPEA 可以说是在 MOEAs 研究中具有里程碑意义,因为该算法的提出使得精英解保留机制在多目标进化领域流行起来,从而开创了一个不同于早期 MOEAs 研究的时代。该算法采用一个称为 archive 的外部集合保存当前已经找到的非支配解,即在每一代,非支配个体都被保存在这个外部集合中,并且计算集合中每个个体的“强度”值。该算法中非支配解的“强度”概念与前面介绍的 MOGA 中的个体的 Pareto 排序数类似,用个体所支配的群体中的个体数目所占比例来表示。而当前种群中(去掉非

支配个体后的种群)的每一个个体的适应度,根据外部集合中所有非支配解对其的支配强度来确定,首先计算外部非支配个体集中支配它的所有个体的强度和,然后再将该强度和加1得其适应值。当外部集合中的个体数目超出约定值时,则采用聚类技术减少该集合里不断增加的个体。这种适应度分配方法,可以在一定程度上保证对真实 Pareto 前沿逼近的同时保持解集的分布性。然而该算法在分配适应度时没有把所有的个体都考虑进去,当种群中的某些个体仅被外部集合中的同一个体控制时,那么这些个体具有相同适应度值。特别地,当外部集合中仅有一个个体时,那么种群中除该非支配个体所有个体将拥有同一适应度值。因此,该算法适应度分配机制会导致对同一非支配级的个体所分配的适应度有偏差,从而可能导致算法过程的完全随机化,出现搜索效率降低,算法收敛速度减慢的现象,同时该算法采用的聚类法在有些情况下还会导致边界个体的丢失。

SPEA2 与 SPEA 相比,主要在以下三个方面进行了改进:首先 SPEA2 采用了更“精细”的适应度分配方式,对外部集合和进化种群中的每个个体所支配的个体数都要计算。然后对每个个体,分别计算支配它的所有个体的强度和,将其作为个体的原始适应值作为适应度值的一部分。其次,对于具有相同适应度值的个体,该算法采用了一种 k 最小近邻密度估计方法,即对于具有相同适应度值的每个个体,计算由该个体到它的第 k 个邻近个体的距离决定的拥挤度,将其作为适应度值的另一部分。第三部分的改进是对于外部集合更新过程的调整,提出了一种新的区域剪裁方法以保证边界个体不丢失。该算法被证明具有较好的收敛性和分布性,然而多样性的维持仍然需要较大的种群规模。

PAES 是一个优秀的局部搜索算法。该算法采用 1+1 进化策略,维持一个用来记录先前找到的非支配解的历史外部集合,作为变异个体比较的参考,将变异前后的支配关系作为比较的度量,采用精英解保留策略保留其中较好的个体。值得一提的是该算法引入了超网格机制来保持种群的多样性,将每个个体分配进一个(超)格子,从而简单方便地说明了个体密度信息。

NSGA-II 是对 NSGA 的改进,主要表现在以下三个方面:首先采用了快速非劣排序方法构造非支配个体集合,将计算复杂度由原来 NSGA 中的 $O(MN^3)$ (其中 M 为目标数, N 为种群大小)降低到 $O(MN^2)$ 。其次采用拥挤距离比较算子代替原来 NSGA 中的适应度共享方法,即对于具有不同非劣级别的个体,选择低级别的个体;对于具有相同非劣级别的每一个个体,计算该个体周围基于每一目标的最近两个个体之间的某种平均距离,作为该个体的拥挤距离,从而保证了该算法能够形成均匀分布的 Pareto 前沿。第三个方面的改进就是引入了精英解保留机制,以此保持优良个体,提高整个种群的进化水平。除了对原有 NSGA 算法进行改进,NSGA-II 还给出了一种多目标优化问题中对约束的处理方法,成为后来约束 MOEAs 的一个重要参考。

总结以上具有代表性的 MOEAs 的特征,可以得到以下几个成功经验:

1. 根据个体间的 Pareto 支配关系确定个体适应值;
2. 构造归档集 (archive) 保留算法的非支配个体;
3. 当非支配个体数量大于 archive 集的限定规模时,采用一定的方法对其进行修剪。

同时,这些 MOEAs 也存在一些不足之处:

1. 维护较好的种群多样性需要较高的时间复杂度与较大的计算量;
2. 有些方法对参数的设置依赖性很强,而许多参数需要根据给定问题的先验知识和经验进行调整。

为了解决上述问题,研究者们对 MOEAs 进行了改进,同时也在寻找其他的优化算法,将其引入到多目标优化领域。

1.3.2 多目标进化算法的关键技术

MOEAs 的关键技术包括:个体适应值的选取,群体多样性保持,精英解保留策略以及选择算子的确定。上述关键技术相辅相成,确立了 MOEAs 的特点,下面对其进行分析。

1. 个体适应值的选取

个体适应值的选取是对个体评价的一种实现机制,是衡量个体优劣的重要准则。与单目标优化算法不同,多目标优化算法中个体适应度的选取必须同时考虑多个目标,所以现在的 MOEAs 中应用较为广泛的个体适应度选择策略通常都是基于传统 Pareto 支配关系的。这类个体适应度选取方法的基本思想是基于 Pareto 支配概念在群体中建立了一种个体间赖以比较的偏序关系,从而反映出一个个体与群体中其他个体的相关性。如前面 1.3.1 节中的 MOGA, NSGA, SPEA, NSGA-II, SPEA2 等算法中的个体适应度选择都属于这类方法。由于基于 Pareto 支配概念,使得不存在支配关系的个体无法判断优劣程度,所以仅仅依靠 Pareto 支配概念来定义适应度是不全面的,研究者们为此设计了多种方式与该类方法结合^[38-46],例如 NSGA-II 将快速非劣排序法与个体拥挤程度相结合,将群体按照 Pareto 支配关系分成不同的层次,对于同一层次中的个体,参考其拥挤程度来确定个体的适应度值;SPEA 和 SPEA2 将个体支配的其他个体的数量与个体被支配的个体数量结合在一起,从而确定个体的适应度值。

随着人们对 MOEAs 的深入研究,算法需要解决的多目标优化问题规模不断扩大,导致非支配解急剧增长。因为随着目标的增长,个体间的关系就会变得愈加复杂,在传统 Pareto 支配的意义下,个体间的关系更趋近于“等价”,因此需要新的判别准则来评判个体间的优劣。所以个体适应度的选取策略从基于传统的 Pareto 支配概念转向了新的个体间的优化关系^[47-52],现在较为常用的是 Laumanns 和 Deb 等 2002 年提出

的 ε -dominance^[47]。但由于参数 ε 对算法影响较大, 且较难确定, ε -dominance 并未占主导地位, 所以新型支配关系的研究正在探索中。

2. 种群多样性的保持

对于多目标进化算法, 群体多样性的保持是一个非常重要的研究方面, 其对 Pareto 最优解的性能具有重要影响。综合前面提到的经典 MOEAs 及近年来发展起来的一些新的 MOEAs^[53-59], 对于群体多样性的保持大致有如下几种较为有效的方法:

(1) 基于小生境技术的共享函数法。这类方法通常是设置一个小生境半径, 计算该半径内个体的相似程度, 从中选择优秀个体生成新的下一代。MOGA, NPGA, NSGA 等都是采用这类方法来保持群体的多样性。然而这类方法对小生境半径的依赖程度很大。

(2) 基于拥挤距离法。这类方法中的拥挤距离是指与某个个体相邻的两个个体基于每一目标的距离差的和。如 NSGA-II 就是采用的这类方法来保持群体的多样性。这类方法首先将所有个体按照每一个目标函数值进行升序排列。然后再根据下面的公式在每个目标上都进行一次计算, 从而得出每个个体的拥挤距离值:

$$distance'_j = \begin{cases} \infty & \text{if } i \text{ is boundary individual} \\ distance'_{j-1} + \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{\max} - f_j^{\min}} & \text{if } i \text{ is not boundary individual} \end{cases} \quad (1.11)$$

为了防止边界解丢失, 对每一目标函数, 将边界(具有最大和最小函数值)的非劣个体所对应的拥挤距离值设置为无穷大。而对于中间个体, 为了平衡目标之间的差别, 所以在计算每个目标下距离个体 i 相邻的两个个体的距离差时, 要除以该目标下目标函数的最大和最小值之差。这类方法不涉及到小生境参数的选择, 但计算拥挤距离前需要对每个目标下的个体函数值进行排序, 对于具有 N 个个体的种群而言, 在最坏情况下, 对 k 个目标分别进行排序的时间为 $O(kN \log N)$ 。

(3) 基于聚类技术法。这类方法首先计算个体或类间的相似度, 然后通过聚类方法将整个非支配解集中的非支配解分成若干个聚类, 当聚类的个数超过一定数量时将距离最近的两个聚类合并, 将合并后的中心点作为新类的位置, 即通过聚类算法降低非支配集的大小, 来保持多目标进化群体的多样性。SPEA 等算法都采用这类方法保持群体的多样性。

(4) 基于网格法。这类方法以众多不同的方式被 MOEAs 所采纳, 用来保持进化群体的多样性。这类方法首先将目标空间分割成若干小区域, 每个小区域被称为(超)立方体, 具体的分割个数取决于进化群体的大小及待优化的目标数量。然后将每个(超)立方体进行标识, 并判断个体落在哪个(超)立方体中。为保持进化群体的多样性, 通常选择聚集密度较大的(超)立方体中的个体进行删除。PESA 等算法都采用这类方法保持群体的多样性。

3. 精英解保留策略

MOEAs 一个非常成功的经验就是对精英解进行保留, 这样大大提高了算法的性能。由于 MOEAs 大多都是随机性优化算法, 所以在进化过程中, 为了不使性能优良的解丢失, 往往采用一个辅助群体将其保存, 保存的精英解随着搜索的进行而不断更新。现在很多的 MOEAs, 充分利用 Pareto 支配关系和个体密度信息, 对个体进行评价, 将优良个体进行保存。

1.3.3 多目标优化算法的性能评价

对于一个 MOEA 的优劣, 往往是通过对其性能进行评价来反映的, 即采用一套能够客观反映算法性能优劣的评价方法, 在一组比较具有代表性的测试问题上进行测试, 通过计算 MOEA 求得一个 MOP 的 Pareto 非劣解集的时间花费以及评价所求得的解集的质量, 考查 MOEA 的执行效率以及优化后的效果。对于 MOEAs 的性能评价可以通过理论分析及实验测试两种方式进行, 通常 MOEAs 算法的运行效率及收敛性可以通过理论分析的方法来解决, 而其他性能评价往往采用实验手段, 通过比较分析的方法来实现。对 MOEAs 评价方法很多, 主要用来评价 MOEAs 对真实 Pareto 前沿的逼近程度和解集的分布性。

1. MOEAs 收敛性评价

目前, 通常使用基于世代距离的趋近度 (Generation Distance, GD) 衡量一个多目标优化算法的收敛性。GD 定义如下:

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (1.12)$$

其中, n 是通过算法得到的非劣解的个数, d_i 是算法中第 i 个解到真实 Pareto front (PF_{True}) 的最小距离。该指标反映算法得到的 Pareto front (PF_{Know}) 与 PF_{True} 的逼近程度, GD 越小说明通过算法得到的非劣最优解集越逼近真实解集。

2. MOEAs 解集分布性评价

对于多目标优化算法, 目前通常使用距离 (Spacing, SP) 衡量算法的收敛性。

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (1.13)$$

$$d_i = \min_j \left(\sum_{k=1}^n |f'_k(x) - f'_k(x)| \right)$$

其中 $i, j = 1, \dots, n, j \neq i$, $\bar{d}$ 为 d_i 的平均值。该指标反映算法得到的 Pareto front 的均匀性, SP 越小说明算法得到的非劣最优解集分布性越好。

1.3.4 目前多目标进化算法存在的问题

首先,由于进化算法遵循生物进化机理的启发机制,因此单一的遗传算法收敛速度并不高。多目标优化问题本身往往比较复杂,在算法的设计过程中,适应度函数的选取,多样性保持等方面要充分体现出多目标优化问题特征,所以多目标进化算法计算复杂度较高,时间花销较大,效率较低。并且有些 MOEAs 对参数的设置依赖性很强,而许多参数需要根据给定问题的先验知识和经验进行调整。

其次,目前多目标进化算法的研究主要围绕无约束多目标进化算法进行,相对而言,约束多目标进化算法研究较少。现有的大多数约束多目标进化算法的研究,主要集中于为了结合约束条件而扩展 Pareto 优化关系,例如约束保持法,即不可行解被忽略,即使它是非支配解。罚函数也是一种非常容易与多目标进化算法融合的约束处理方式,然而其系数确定较为困难,往往需要先验知识。多目标优化问题约束搜索空间中的有效处理机制实际上仍未开发出来。

为了解决上述问题,研究者们对多目标进化算法进行了改进,同时也在寻找其他的优化范例,将其引入到多目标优化领域。微粒群优化(Particle Swarm Optimization, PSO)算法由于其收敛速度较快,参数设置较少的特点^[60],被引入到多目标优化领域,并取得了许多研究成果。

1.4 多目标微粒群算法

随着演化计算的发展,一些基于群体智能的算法被开发出来,并被引入到多目标优化的领域,其中以多目标微粒群优化算法最为常见,成为基于群体智能的多目标优化算法的代表。1999 年 Moore 和 Chapman 首次提出将 PSO 算法应用于解决多目标优化问题^[61],但这个思想未公开发表,从此后用 PSO 解决多目标优化问题开始得到研究人员的关注,但直到 2002 年 Coello 等和 Ray 等先后发表了多目标微粒群优化(Multi-Objective Particle Swarm Optimization, MOPSO)算法的论文^[62,63],继之多目标微粒群优化算法才逐渐被研究者们重视,出现了大量研究成果^[64-78]。应用 PSO 优化算法解决多目标优化问题已经存在许多版本,早期的 MOPSO 算法主要关注与对多个优化目标处理方式的选择,直至最近,研究才关注于在已存在的算法上产生新的变化,比如关注在 MOPSO 中 PSO 算法的主要组成部分所起到的作用、最好解的选择机制和进化方程的参数选择机制等,这些对 MOPSO 算法的性能起关键作用。

1.4.1 多目标微粒群算法分类

根据对多个优化目标处理方式以及种群最优解选取方式,多目标微粒群算法大致

可以分为以下几类：聚集函数法、基于目标函数排序法、子群法、基于 Pareto 支配算法和其他方法。

1. 聚集函数法

聚集函数法是解决多目标优化问题的一种最直接的方法, 在传统多目标优化算法中较为常用, 对其介绍见 1.2.1 节。较具有代表性的基于聚集函数的 MOPSO 算法是文献[79], [80]和[81]。

文献[79]分析了固定权重、突变权重和动态权重这三种权重聚集方法。固定权重在优化过程中权重是固定不变的, 但计算量很大, 并且无法找到非凸解。突变权重和动态权重可以克服这个缺点, 只是突变权重一般是周期函数的复合函数, 变化比较频繁, 变化的尺度由周期函数的外层函数决定。动态权重在突变权重的基础上将复合函数变成了周期函数的绝对值函数, 这样变化的尺度相对突变权重要缓和一些。

文献[80]使用了一种线性权重聚集法。在这类方法中, 整个群体被平均分成了若干个子群体, 每个子群体被赋予不同的权重, 每个子群体中的微粒都朝着自己所处子群中的最优解“飞行”, 然后通过线性权重进行聚集, 得到总目标下群体全局最优微粒。

文献[81]使用了一种类似于权重聚集的方法。首先找到在每个目标下群体的全局极值和每个微粒在每个目标下的个体极值, 然后用各目标下全局极值的均值作为整个群体在总目标下的全局极值, 并通过判断每个微粒在每个目标下相对于该目标下全局极值的离散程度来选取该粒子在总目标下的个体极值。

以上 3 种方法都比较简单, 容易实现。但是算法每迭代一次只能产生一个“最优解”, 所以要得到接近 PF_{True} 的 Pareto 前沿则需要的迭代次数较大, 从而带来的计算量就会较大。并且这 3 种方法都没有很好地体现出 Pareto 支配的概念, 没有精英解保留机制, 这样使解的质量受到影响。

2. 基于目标函数排序方法

在这种方法中目标函数通常要根据某个标准进行排序, 然后按照此顺序对目标函数进行优化。比较具有代表性的方法是 Hu 和 Eberhart 的动态邻域策略及其扩展^[82,83]。前者主要针对两个目标函数在环形邻域上进行的优化, 没有采用外部存储体保存精英个体, 后者在前者的基础上引入了精英解的保留机制。这类方法的主要思想是首先根据第一个目标来计算当前考查微粒与其他微粒的距离, 并根据距离建立若干个邻域, 然后根据第二个目标函数选取邻域最优解。

这类方法中目标函数的顺序是非常重要的, 并且这种方法处理的目标函数往往较少。

3. 子群法

这类方法的基本思想是将整个群体划分为若干个子群, 每个子群中进行单目标优

化,然后通过子群间交换信息或将子群信息组合来产生群体的最优解。比较具有代表性的方法有文献[79]及其改进算法 VEPSO^[84],文献[85]等。

文献[79]针对两个优化函数的多目标优化问题提出了用两个子群体分别进行单目标优化,然后把第一个子群得到的最优解作为第二个子群更新速度的全局最优解,将第二个子群的最优解作为第一个子群更新速度的全局最优解。文献[84]将此思想进行改进,把每一维目标作为一个子群,每个子群进行单目标优化得到最优解,并将其作为邻接子群更新速度的全局最优解。

文献[85]对两个目标函数同时进行优化,根据多目标优化问题的特点将进化群体划分为第1个目标下的全优,第1个目标下的半优,第2个目标下的全优,第2个目标下的半优以及两个目标下的全劣这几个子群体。然后根据支配关系找出每个子群体中的全局最优个体,它们构成一个非支配集,从中选出群体的全局最优解。

文献[86]利用 Pareto 支配概念确定微粒的“飞行”方向,采用聚类的方法将种群分为几个子群,每个子群体产生非支配解集,从其中随机选择一个作为每个子群执行 PSO 算法时的全局最优解,子群间通过移植不同子群中的非支配解交换信息。

4. 基于 Pareto 的方法

现在的大多数研究主要集中于这种基于 Pareto 的方法。借鉴以前用进化算法解决多目标优化问题的一些成功经验,在进化过程中经常更新两个集合,一个是用来存放进化的基本种群 *population*,另一个是用来存储进化过程中的精英个体 *archive*。对于基本种群 *population*,首先确定该种群中个体的优劣,然后确定种群的全局最优个体和每个个体的历史最好位置,利用 PSO 公式进行更新,得到下一代基本群体。对 *archive* 一般要进行两种操作: *update* 和 *truncate*,前者用来更新 *archive* 中的个体保持为进化中的最优个体,后者是当要进入 *archive* 中实际个体的数目大于 *archive* 容量时,对其中个体进行修剪剔除。

该类方法中,比较具有代表性的算法有 Coello 等提出的 MOPSO^[62,87]以及后来研究者对其进行的改进^[88-90]。

MOPSO 算法是基于 PSO 求解多目标优化问题中最为经典的算法,在这个算法中 Coello 等首次提出了除基本微粒群引入第二个群体来保存生成的非支配解,并第一次用(超)网格来保存精英解。文献[87]是对文献[62]的改进,将原来固定的(超)网格变为自适应(超)网格,并使用了变异算子来更新种群微粒和决策变量的范围,变异尺度与进化代数成比例。这种方法成为后来研究者们改进的基础,也是研究者们对算法优劣的重要参照。

文献[88]在文献[62]的基础上借鉴 SPEA2 算法中环境选择和配对选择策略,并使用了密度估计技术,提高了解的多样性和搜索精度。但该算法中有3个群体,计算量较大。

文献[89]在文献[62]的基础上对算法进行了两方面的改进：①算法迭代到一定次数时对某个重要目标进行单目标优化，以此来解决 MOPSO 中出现的难以找到边界点的问题；②对超出边界的位置变量用一个随机数来取代。这两方面的改进主要提高了 Pareto 解集的分布性，但要求事先知道各优化目标的重要程度。

文献[90]对文献[87]中的自适应网格进行了改进，加入了非支配解集中微粒密度估计信息，在剔除密度较大网格中微粒的时候，确定了需要剔除的微粒的数量，给出了计算公式。并且该算法在全局最优解的选取方式上也进行了改进。

5. 其他方法

文献[91]提出了一种新的多目标微粒群算法：EMOPSO，该算法在 MOEA 的基础上进行了修改。对非支配解集分布性的保持提出了一种新颖的“hyper-plane”方法，该方法在文献[87]超网格的基础上，针对两个目标函数首先找到每一个目标的最小值 A 和 B，然后将这两点用直线连接，如果需要找到 n 个非支配解，则在此线段上均匀地再选择 $(n-1)$ 个点，接着在这 $(n-1)$ 个点上作直线 AB 的垂直线，非支配解中距离此垂直线最近的点被选中，其余各点被删除。此算法为了避免早熟收敛而加入了扰动因子，随着迭代次数的增加，扰动因子逐渐减小。此算法对约束处理方法以及参数自适应调整等方面都进行了研究。

对于上述所分析的 MOPSO 算法及其特征如表 1.1 所示。

表 1.1 经典 MOPSO 算法及其特征

算法		archive	修剪 archive	变异算子	Gbest 选取
聚 集 法	Parsopoulos 等[79]	无	——	无	单目标
	Baumgartner 等[80]	无	——	无	单目标
	张利彪等[81]	无	——	无	评估选取
排 序 法	Hu and Eberhart[82]	无	——	无	单目标
	Hu and Eberhart[83]	有	——	无	单目标
子 群 法	Parsopoulos 等[79]	无	——	无	单目标
	Parsopoulos 等[84]	有	——	无	单目标
	张利彪等[85]	有	——	有	——
	Pulido 等[86]	无	——	无	子群最优解交换
基 于 Pareto 方	Coello 等[62]	有	(超) 网格	有	网格中微粒密度
	Coello 等[87]	有	自适应网格	有	网格中微粒密度
	熊盛武等[88]	有	密度和优劣信息	无	轮盘赌
	Chamaani 等[89]	有	网格	有	网格和单目标

法	杨俊杰等[90]	有	自适应网格	无	密度和所支配的微粒 个数
其他	Pulido 等[91]	有	hyper-plane	有	——

1.4.2 多目标微粒群算法的关键技术

用 PSO 解决多目标优化问题往往借鉴 MOEAs 中的成功经验,但 MOPSO 既不同于 MOEAs,又不同于用 PSO 解决单目标优化问题, MOPSO 算法的关键技术研究一般围绕以下几点进行:如何选择非支配解、如何剔除 *archive* 集中的个体以保持其良好的分布性、如何保持解集多样性和 g_{best} 和 p_{best} 如何选取等。

1 非支配解的选择

对于这个问题的研究与前面 1.3.2 节中介绍的 MOEAs 的适应值的选取相似,涉及到传统 Pareto 支配的概念及后来研究者们对其发起的挑战。现有的 MOPSO 算法中非支配解的选取大都是基于传统 Pareto 支配机制的,即个体支配机制。2002 年, Laumanns 和 Deb 等提出 ε 支配概念后其思想被借鉴与改进^[47],2003 年 Mostaghim 等将其引入到 MOPSO 算法中^[69]。文献[92]为了保持解集良好的分布性,针对原有 ε 支配可能会丢失边界点的情况提出了强 ε 支配概念,在原有 ε 支配概念的基础上又加入了个体支配的概念。

2. *archive* 集的修剪

当前 MOPSO 算法中绝大多数都用到了容量受限的 *archive* 集来保存精英解,所以一般要对 *archive* 集进行修剪,剔除掉一部分非支配解, *archive* 集中的结果就是将来要输出的解,所以在剔除的过程中要考虑到多目标优化问题的特点。在现有的 MOPSO 算法中,大多采用 MOEA 中的经验,考虑较多的是如何剔除 *archive* 集中的个体,保持解集良好的分布性。通常采用的方法有:随机选择^[71],小生镜技术^[72,73],网格技术^[62,74,87,90],拥挤距离^[70,75,76],密度熵^[77],聚类技术^[91]等。

3. 保持解集多样性

PSO 算法最突出的一个特点就是收敛速度快,但这个特点同时也带来了算法早熟收敛的问题。为了增加解的多样性,避免早熟收敛,将 PSO 应用于解决多目标优化问题时往往借鉴多目标进化算法中的变异算子,有的 MOPSO 算法对微粒进行变异的同时对决策变量的范围进行变异,有的研究采用对“飞”出可行域的微粒进行重新初始化的方法来保持非支配解集中解的多样性。

文献[62]提出了一种使用变异因子来更新基本种群中微粒和决策变量范围的方法。该方法在搜索开始时对所有微粒进行变异,随着迭代次数的进行,种群中变异的微粒数量迅速下降,对每个决策变量范围也采用同样操作。这样,算法可以有一个很

强的搜索能力,随着迭代次数的增加,变异算子的作用逐渐减小,从而避免了算法早熟。文献[94]也采用了类似的方法来保持解的多样性。

文献[72][76]使用了小概率随机变异的方法来保持解集的多样性。这类方法的基本思想是在基本种群每次迭代的过程中随机选择(或从被支配解中选)很少一部分(例如变异概率为5%)微粒进行变异,这样以增强算法的全局搜索能力。

文献[89]对超出边界的微粒不用传统方法中将其拉至边界的处理方式,而是用一个可行域内的随机数将其替代,这样以增加解的多样性。

4. p_{best} 和 g_{best} 的选取

在 PSO 算法中,微粒是在种群全局最优位置 g_{best} 及个体历史最优位置 p_{best} 的指导下“飞行”的。将 PSO 算法应用于解决多目标优化问题时,每次迭代不再是仅仅产生一个单独的 p_{best} 值和 g_{best} 值,而是一组非支配解,如何选择合适的 p_{best} 和 g_{best} 来指导种群中微粒的“飞行”就显得非常重要了。同时 p_{best} 和 g_{best} 的选取充分体现了 MOPSO 自身的特点,对 MOPSO 算法的性能起关键作用,为这个领域研究打开了新的路子。由于 p_{best} 和 g_{best} 是 PSO 算法中特有的参数,因此它们的选取不能直接借鉴原有的 MOEAs 的成功经验。

通常对于 p_{best} 的选取一般都基于 Pareto 支配概念,将新产生的 p_{best} 与原来的 p_{best} 进行比较,选择非支配者作为算法迭代中的 p_{best} ,若二者不存在支配关系,则随机选择其中之一即可,但也有算法根据所支配的群体中微粒的个数来选择。

g_{best} 的选取是 MOPSO 算法设计中的关键, g_{best} 的选取在很大程度上与 archive 集的修剪及更新方式相关。 g_{best} 选取最简单最直接的方法就是从 archive 集中随机选取一个微粒作为基本种群更新的全局最优解^[74,77]。文献[72]采用轮盘赌概率方法从 archive 中选取精英微粒作为 g_{best} ,文献[75]通过随机从 archive 中选取两个微粒,从多个目标函数中随机选取一个目标进行比较的方法来选取 g_{best} 。张利彪等使用最优解评估选取的方式来选择种群更新的 g_{best} ^[81],文献[93]也使用了这种方法。

现有 MOPSO 算法中, g_{best} 的选取除了随机选择外通过密度信息来进行选择的较多。文献[62][87]使用(超)网格中微粒的密度信息,密度小者优先的方法来选取 g_{best} ,后来在其算法基础上进行改进的一些算法一般沿用了这种方式,或对其稍作调整^[90]。还有其他的一些选取 g_{best} 的方法。文献[92]使用了一种极值变异的方法来选择 g_{best} ,首先从 archive 中随机选择一个非支配解,然后将此解与当前 g_{best} 比较,较好者作为此次迭代的 g_{best} ,若两者一样好,则根据极值变异概率来决定 g_{best} 是否需要重新从 archive 中随机选择。

1.4.3 目前多目标微粒群算法存在的问题

首先, MOPSO 算法在对多目标优化问题的处理方面,基本上采用了 MOEAs 的

成功经验,例如适应值的选取方式,多样性的保持,非劣最优解集的修剪等,这样 MOEAs 中存在的计算复杂性的问题在 MOPSO 算法中并未得以解决。

其次,PSO 算法本身存在有早熟收敛,容易陷入局部最优的不足,在解决 MOPs 中,这种问题不可避免的呈现出来。在一些改进的 MOPSO 算法中,依靠引入其他算法或采用变异策略来解决这个问题,这样往往增加了算法的复杂性和计算量。

第三,对于 MOPSO 算法而言,参数间具有非常高的非线性作用,而自适应参数的设计,可以使得 MOPSO 算法摆脱人为干涉,但这个方面还很少有人设计。

鉴于 MOEAs 及 MOPSO 算法存在的问题,研究者们对其进行了改进,同时也在寻找其他优秀的优化算法,将其引入到多目标优化领域。

1.5 本论文的研究意义及研究内容

1.5.1 论文的研究意义

基于种群的优化算法具有隐并行性以及可以在单轮模拟过程中得到多个解等特点,非常适合求解多目标优化问题。20 世纪 80 年代中期,进化算法开始应用于解决多目标优化问题,表现出了强大的求解能力,在许多实际问题求解中获得了成功,这极大地提高了人们在多目标优化领域的认知水平及解决实际问题的能力。然而当前多目标进化领域仍存在一系列问题:

遗传算法本身收敛速度并不高,多目标优化问题本身比较复杂,所以 MOEAs 计算复杂度较高,时间花销较大,效率不高。

MOPSO 算法由于微粒群优化算法本身存在有早熟收敛等不足,所以单一的 MOPSO 算法容易陷入局部最优。在一些改进的 MOPSO 算法中,引入其他算法或采用变异策略来解决这个问题,这样往往增加了算法的复杂性。

约束处理,目前的大多数研究集中于为了结合约束条件而扩展 Pareto 优化关系,或采用罚函数作为约束处理方式。然而前者对一些等式约束不能有效处理,后者往往需要先验知识。

拟态物理学优化(Artificial Physics Optimization, APO)算法是一种新近提出的优化算法^[95-100],在解决单目标全局优化问题时,表现出了相对传统进化算法及 PSO 算法更优良的性能。借鉴已有的 MOEAs 和 MOPSO 算法的成功经验,将 APO 算法引入到多目标优化领域,借助于 APO 算法自身机制特点,解决传统 MOEAs 及 MOPSO 算法中的一些不足,为解决多目标优化问题提供新途径。因此,本文对多目标拟态物理学优化(Multi-Objective Artificial Physics Optimization, MOAPO)算法进行理论研究及性能分析,同时将其应用于移动机器人路径规划中。本文的研究将为解决多目标

优化问题提供新途径,为丰富基于群体智能的多目标优化方法提供新思路,对多目标优化领域的发展具有重要的理论和现实意义。

1.5.2 论文的研究内容

本文的研究目标在于利用拟态物理学优化算法,构造一种新的多目标优化算法(若无特别说明,本文讨论的都是最小化多目标优化问题),研究内容包括多目标拟态物理学优化算法可行性分析,无约束多目标拟态物理学优化算法的基本框架建立,有效性研究,多样性研究,适应值选择,收敛性分析,带约束多目标拟态物理学优化算法研究及其收敛性分析,以及多目标拟态物理学优化算法在移动机器人路径规划中的应用研究。

鉴于拟态物理学优化算法在解决单目标全局优化问题中表现出良好的性能,通过比较拟态物理学优化算法与典型的基于种群的优化算法之间特征的异同,阐述了基于拟态物理学优化算法解决多目标优化问题的可行性,分析了将拟态物理学优化算法应用于多目标优化领域需要解决的关键问题,建立了多目标拟态物理学优化算法的基本框架。在此基础上,基于聚集函数法提出一种多目标拟态物理学优化算法,通过与经典多目标进化算法及多目标微粒群优化算法在一系列标准的、复杂的多目标优化测试函数上的比较结果,说明了多目标拟态物理学优化算法的有效性。

多样性是多目标优化算法性能好坏的一个重要指标。本文从拟态物理学优化算法自身特点出发,分析了个体作用于种群中其他个体的虚拟力之和与种群多样性的关系,提出了一种基于虚拟力排序的多目标拟态物理学优化(VFMOAPO)算法,基于概率论的基本知识,对其收敛性进行了理论证明,通过数值实验分析了该算法的性能,并且通过与经典多目标进化算法及多目标微粒群优化算法的比较,说明了该算法在保持 Pareto 解集多样性方面具有较好性能。

适应值的选取是多目标优化算法研究的一个重要方面。本文从多目标优化问题的特点出发,将基于 Pareto 的序值概念及基于邻域半径的共享技术用于多目标拟态物理学优化算法中的质量函数的确定,提出一种基于序值的多目标拟态物理学优化(RMOAPO)算法。通过与经典多目标进化算法、多目标微粒群优化算法及标准的多目标拟态物理学优化(SMOAPO)算法比较的数值实验,分析了该算法性能。

现实世界中存在的多目标优化问题大多数是带约束的,然而相对于无约束多目标优化算法的研究,约束多目标优化算法的研究相对较少。本文首先利用可行规则法作为约束处理机制,设计了可行个体及不可行个体的质量函数及虚拟力作用规则,在此基础上提出一种基于虚拟力减小的约束多目标拟态物理学优化(VDCMOAPO)算法,有效处理了可行个体进入不可行域的情况,从而较好解决了 Pareto 最优解集位于可行域及不可行域边界的约束多目标优化问题,并且利用概率论的基本理论对该算法的收

敛性进行了分析与证明,通过仿真测试说明了该算法的性能。然后,本文将约束保持法的基本思想与RMOAPO算法相结合,提出了一种基于序值的约束多目标拟态物理学优化(CRMOAPO)算法。通过数值实验,分析了该算法性能。

应用MOAPO算法解决移动机器人路径规划问题。针对具有多个优化目标的移动机器人全局路径规划问题,本文提出了一种基于约束保持法的约束多目标拟态物理学优化算法和一种基于可行规则法的约束多目标拟态物理学优化算法。首先根据障碍物信息建立机器人工作空间模型,将移动机器人路径规划问题转化为连续空间的约束多目标优化问题。然后构造移动机器人路径规划的适应值函数,将约束保持法作为约束处理机制,利用基于序值的约束多目标拟态物理学优化算法对移动机器人的路径进行规划。通过与经典算法的仿真实验说明了将MOAPO算法应用于移动机器人路径规划的可行性与有效性。最后采用可行规则法作为约束处理机制,利用基于虚拟力减小的约束多目标拟态物理学优化算法对移动机器人路径进行规划,通过与基于序值的约束多目标拟态物理学优化算法得到的路径进行比较,分析了两种算法所适用的工程领域范围。

1.6 论文的组织结构

本文分为七章。第一章主要介绍多目标优化的基本概念、传统多目标优化方法、多目标进化算法及多目标微粒群优化算法的研究动态和关键技术,以及目前存在的问题等,提出本论文的研究目标及研究内容。

第二章首先介绍拟态物理学优化算法的基本理论,分析将拟态物理学优化算法引入到多目标优化领域的可行性,在此基础上建立了多目标拟态物理学优化算法的基本框架。

第三章提出了一种基于聚集函数法的无约束多目标拟态物理学优化算法,通过理论分析和仿真测试说明了将拟态物理学优化算法应用于解决多目标优化问题的有效性。

第四章提出了一种基于虚拟力排序的多目标拟态物理学优化算法,主要从群体多样性方面对算法进行研究,通过理论分析和仿真测试说明了多目标拟态物理学优化算法自身在非劣解集多样性方面具有良好的性能。

第五章对无约束多目标拟态物理学优化算法适应值函数进行研究,提出了一种基于序值的多目标拟态物理学优化算法,利用多目标优化领域中的序值概念来定义质量函数,以充分体现出多目标优化问题自身的特点。并通过仿真结果进行了性能分析。

第六章对约束多目标拟态物理学优化算法进行研究,首先对约束多目标优化算法的发展状况进行了介绍。其次基于可行规则法对算法的质量函数及虚拟力作用规则进行了设计,提出了一种基于虚拟力减小的约束多目标拟态物理学优化算法。然后基于

约束保持法提出了一种基于序值的约束多目标拟态物理学优化算法,并分别通过仿真实验对两种约束多目标拟态物理学优化算法进行了性能分析。

第七章基于多目标拟态物理学优化算法设计了一种移动机器人路径规划算法。首先简要介绍了现有移动机器人路径规划方法,其次建立了移动机器人的工作环境模型,然后建立了移动机器人路径规划的多目标优化模型,利用基于序值的约束多目标拟态物理学优化算法对移动机器人路径进行规划,通过仿真实验测试及与经典算法的比较说明了该算法的可行性和有效性。接下来利用基于基于虚拟力减小的约束多目标拟态物理学优化算法对移动机器人路径进行规划,通过与基于序值的约束多目标拟态物理学优化算法比较的仿真测试,分析了两种算法工程应用的领域面向。

结论部分对全文工作进行了回顾,对研究的后继工作进行了展望。

第2章 拟态物理学优化算法及多目标拟态物理学优化算法的基本框架

本章首先介绍了拟态物理学优化(APO)算法的基本原理和框架,以及在解决单目标全局优化问题中影响其性能的主要因素,然后将拟态物理学优化算法与典型的基于种群的多目标优化算法进行了比较,分析了应用拟态物理学优化算法解决多目标优化问题的可行性,并在此基础上给出了多目标拟态物理学优化算法的基本框架。

2.1 拟态物理学优化算法

2.1.1 APO 算法提出

拟态物理学最早是由 Spear 等提出^[94],主要用于解决多机器人系统的分布式控制。本质上该方法模拟了牛顿第二定律($F=ma$)。在这种方法中,机器人被抽象为具有质量 m 、位置 X 和速度 V 的个体。个体在空间的连续运动用若干个小的离散时间片 Δt 内的位移量 ΔX 来近似描述,即

$$\Delta X = V \cdot \Delta t \quad (2.1)$$

速度变化量为

$$\Delta V = (F/m) \cdot \Delta t \quad (2.2)$$

其中 F 为个体受到其他个体和环境作用力的合力。因此,个体在 t 时刻的位置为

$$X(t) = X(t-1) + V(t) \cdot \Delta t \quad (2.3)$$

速度为

$$V(t) = V(t-1) + (F/m) \cdot \Delta t \quad (2.4)$$

用 $F_{\max}$ 限定每个个体所受到的最大力,从而限制了个体的加速度。用 $V_{\max}$ 限定个体运动速度。

谢丽萍和曾建潮对拟态物理学进行了研究,试图从拟态物理的角度模拟生物群体智能的涌现,首次将拟态物理学成功应用于解决优化问题,提出了拟态物理学优化算法(APO)^[95-100]。该算法中每个个体是解空间中的一个可行解,每个个体根据自己的惯性及所受其他个体的合力作用来调整自己的运动,整个群体所经历的最好位置就是目前找到的全局最优解,个体的好坏由优化问题的适应值来评价。每个个体通过

全局最好、最差适应值和自身适应值不断更新质量,随着质量的变化个体所受到的虚拟力也在改变,从而更新个体的速度和位置。

对于全局最小化问题

$$\min\{f(X) : X \in \Omega \subset R^n\} \quad (2.5)$$

其中 $\Omega = \{X \in \Omega | x_k^{\min} \leq x_k \leq x_k^{\max}, k=1, \dots, n\}$

APO 算法中,若群体规模为 n ,则第 i 个 ($i=1,2,\dots,n$) 个体的质量表示为 m_i ,第 k 维的速度用 $v_{i,k}$ 来表示,第 k 维的位置表示为 $x_{i,k}$,受到第 j 个 ($i \neq j$) 个体的虚拟力为 $F_{ij,k}$,受到群体中所有其他个体总的作用力为 $F_{i,k}$ 。个体的质量 m_i 由下面公式决定:

$$m_i = e^{\frac{f(X_{best}) - f(X_i)}{f(X_{worst}) - f(X_{best})}} \quad (2.6)$$

这里, X_{best} 表示当前种群中的最优个体, X_{worst} 表示当前种群中的最差个体, $f(X_{best})$ 表示最优个体的目标函数值, $f(X_{worst})$ 表示最差个体的目标函数值, $f(X_i)$ 表示个体 i 的目标函数值。从公式 (2.6) 可以看出,个体质量在区间 $(0,1]$ 内变化,即 $m_i \in (0,1]$, 并且满足 $m_{best} = 1$ 。质量函数是一个非负有界单调递减函数,这样,个体的目标函数值越小,个体的质量就越大;反之,个体的目标函数值越大,个体的质量越小。因此,个体的质量可以反映出个体适应值的优劣。

个体 i 在第 k 维上受到个体 j 的虚拟作用力 $F_{ij,k}$ 及个体 i 在第 k 维上受到其他个体的合力 $F_{i,k}$ 由下列公式决定^[98]:

$$F_{ij,k} = \begin{cases} Gm_i m_j (x_{j,k} - x_{i,k}), & f(x_i) > f(x_j) \\ -Gm_i m_j (x_{j,k} - x_{i,k}), & f(x_i) \leq f(x_j) \end{cases} \quad \forall i \neq j \text{ and } i \neq best \quad (2.7)$$

$$F_{i,k} = \sum_{\substack{j=1 \\ i \neq j}}^n F_{ij,k} \quad \forall i \neq best \quad (2.8)$$

其中 G 是引力常数, $x_{j,k} - x_{i,k}$ 表示个体 j 到个体 i 在第 k 维上的距离,且 $F_{ii,k} = 0$ 。适应值好的个体对适应值差的个体有引力作用,适应值差的个体对适应值好的个体有斥力作用,作用力的大小和方向由公式 (2.7) 给出。同时,全局最优个体不受其他个体的作用。公式 (2.8) 描述了种群中除最优个体外的任一个体 i 在第 k 维上受到其他个体的合力。

个体 i 根据下面公式来更新自己的速度和位置:

$$v_{i,k}(t+1) = wv_{i,k}(t) + \lambda F_{i,k} / m_i, \quad \forall i \neq best \quad (2.9)$$

$$x_{i,k}(t+1) = x_{i,k}(t) + v_{i,k}(t+1), \quad \forall i \neq best \quad (2.10)$$

其中 w 是惯性权重,取值在 $(0,1)$ 间, λ 是服从 $\lambda \sim N(0,1)$ 的随机数。个体的运动空间和速度均受到限制,即 $X_i \in [X_{\min}, X_{\max}]$, $V_i \in [V_{\min}, V_{\max}]$ 。从上面公式 (2.9) 和公式 (2.10) 可以看出,对于全局最优个体,其速度和位置不进行更新,而是被保留下

来, 进入下一次的迭代。

公式 (2.9) 和 (2.10) 与 PSO 算法的进化方程相似, APO 算法与 PSO 算法具有以下类似特征:

- (1) 都是基于种群的算法;
- (2) 都是通过速度和位置迭代更新进行优化, 且都是离散行为;
- (3) 种群个体具有完备的空间位置信息, 个体是全局定位的。

二者的不同点主要在于:

- (1) APO 算法中的个体具有质量属性, 而 PSO 算法中微粒不具备质量属性;
- (2) APO 算法中个体行为与群体中其他个体相关, 而 PSO 算法中微粒的行为只与微粒本身的个体最优值和群体的全局最优值相关, 与其他的粒子无关。

PSO 算法中每个微粒的运动受到自身历史最好点和种群最优微粒的吸引, 整个群体具有向更好搜索区域运动的能力。然而, 当某个微粒陷入了局部最优解, 并且这个解足够好, 那么该微粒将吸引其他微粒陷入到这个局部极值, 而不去探索其他未发现的解空间。这样会导致种群的搜索空间有限, 多样性较差, 算法易陷入局部极值。

APO 算法中, 每个个体的运动除了受全局最好个体的吸引, 还要受到群体中其他个体的作用力: 比它好的个体对该个体具有引力, 比它差的个体对其具有斥力。如果将这种引斥力作用看作是个体间的交互和个体的认知与社会学习, 那么 APO 算法中个体获得周围个体的信息就更全面, 从而可以有效避免陷入局部极值的危险。文献 [95-100] 的研究结果表明, APO 算法与 PSO 算法及传统进化算法相比, 具有较好的稳定性、分布性和鲁棒性。

2.1.2 标准 APO 算法流程

标准 APO 算法主要包括三个部分: 初始化种群, 计算个体所受合力, 个体运动更新^[99]。算法步骤如下:

Step1. 初始化种群: 在问题可行域中随机产生种群个体的初始位置和速度; 计算每个个体适应值, 找到全局最优个体及全局最差个体;

Step2. 计算个体所受合力:

Step2.1. 根据式(2.6)计算个体质量;

Step2.2. 根据式(2.7)计算个体所受其他个体的虚拟力;

Step2.3. 根据式(2.8)计算个体所受合力;

Step3. 个体运动更新

Step3.1 根据式(2.9)计算个体的下一代速度;

Step3.2 根据式(2.10)计算个体的下一代位置;

Step3.3 计算个体适应值, 更新种群最优个体、最差 及其相应的适应值。

Step4. 判断是否满足结束条件, 若满足, 则停止计算, 并输出最优结果; 若不满足, 则返回步骤Step2, 进行下一轮迭代。

2.1.3 影响 APO 算法性能的关键因素

APO 算法中, 质量函数及引力常数 G 对算法的性能具有重要作用, 是影响 APO 算法的关键因素。

1. 质量函数

在 APO 算法中, 质量不是一成不变的, 而是一个重要属性, 质量函数反映了个体适应值的优劣: 个体适应值越好, 个体的质量就越大; 反之, 个体适应值越差, 个体质量就越小。即质量是个体适应值的函数, 随着个体适应值在群体中的优劣变化而改变, 个体质量的大小是该个体适应值在整个种群中优劣的一个评价。文献[98]和文献[99]的第四章对 APO 算法的质量函数进行了分析, 构造了直线、凹曲线、凸曲线及先凸后凹曲线四种质量函数, 通过仿真实验得出了凹曲线质量的 APO 算法收敛速度快, 搜索性能最好的结论。

2. 引力常数

引力常数 G 是影响 APO 算法的另一个重要参数, 可以通过参数 G 的选择策略来控制种群中作收敛运动和发散运动的个体的数量。文献[99]第六章对 APO 算法的收敛性进行了分析, 得出了有关参数 m_i , w 和 G 的解序列收敛的充要条件。根据该收敛条件及问题特征, 可以对引力常数进行设计, 从而增强算法的种群多样性。

从公式 (2.7) 可以知道, 引力常数 G 和个体的质量共同作用于个体受到其他个体的虚拟力, 然后通过进化方程影响个体的速度与个体位置。

2.2 应用 APO 算法解决多目标优化问题可行性分析

受自然界规律和生物群体智能行为启发的各类优化方法为多目标优化问题的求解提供了很多新颖有效的思路, 其中以多目标进化算法、多目标人工免疫算法、多目标微粒群优化算法等为代表。针对多目标优化问题, 基于种群的启发式多目标优化算法的一般设计思路是将问题域中随机采样的样本点看作个体或粒子, 个体或粒子在最优适应值引导下, 通过个体或粒子的不断更新逐步求得给定问题的非劣解集。拟态物理学优化方法中个体按照预定的个体间的引斥力规则相互作用, 通过个体间虚拟力的作用搜索潜在的最优解。

APO 算法与 PSO 算法非常相似, 对于二者的比较在前面 2.1 节中已有介绍, 在此不再赘述。目前在多目标进化领域, 经典的算法大都是基于遗传算法的, 因此将 APO 算法与遗传算法进行比较, 从而分析应用 APO 算法解决多目标优化问题的可行性。

APO 算法与遗传算法两者之间有以下相似之处:

- (1) 都是基于种群的方法;
- (2) 都是通过种群中个体的不断更新迭代来实现问题求解的;
- (3) 都具有潜在的并行性;
- (4) 都属于全局优化算法;
- (5) 都属于随机搜索算法。

二者之间也有不同之处:

(1) APO 算法是从拟态物理的角度来模拟生物觅食行为, 而遗传算法是通过模拟自然进化过程搜索最优解的方法。

(2) APO 算法中的个体有质量属性, 并且每一个个体在确定的时刻具有唯一的质量属性; 而遗传算法中的个体是质量为零的理想个体, 具有适应值属性, 其适应值一般通过计算给定的被优化函数值获得, 在确定的时刻, 每个个体对应一个适应值;

(3) APO 算法是通过速度和位置的迭代来更新种群, 而遗传算法是通过遗传算子来更新种群的;

(4) APO 算法的速度进化方程中 λ 是随机数, 因此 APO 算法是随机搜索算法, 而遗传算法中的变异是基于概率的, 因此遗传算法是基于概率的随机搜索算法。

从上面 APO 算法与遗传算法的比较以及前面 2.1 节中 APO 算法与 PSO 算法的比较可知, APO 算法与这两种典型的基于种群的算法在许多方面都是类似的, 因此可以将多目标进化算法和多目标微粒群优化算法中的研究成果引入到多目标拟态物理学优化算法的设计中。诚然, APO 算法与这两种典型的基于种群的算法在一些具体方面有不同之处, 但可以通过分析拟态物理学优化算法特点, 设计一些策略, 使其更适合解决多目标优化问题, 而这些策略正是多目标拟态物理学优化算法设计的关键。

通过上述分析可知, 将 APO 算法应用于解决多目标优化问题领域, 需要解决以下几个关键问题:

1. 怎样设计个体的质量函数来表达多个子目标的适应值, 从而评价个体之间的优劣关系。多目标优化问题不同于单目标优化问题, 在单目标优化领域, 个体质量可以与寻优粒子的适应值建立起直接的映射, 使得在求解最大化问题时, 个体适应值越大, 个体越优, 个体的质量越大; 在求解最小化问题时, 个体适应值越小, 个体越优, 个体质量越大。在多目标优化领域, 每个个体具有若干个子目标, 需要根据这若干个子目标的适应值, 结合 Pareto 最优解概念来确定个体的质量, 从而正确评价个体的优劣程度: 较优个体具有较大质量, 因此可以对比它差的个体产生较大的引力; 较差个体具有较小的质量, 则对其他个体的影响相对较小。这样较优个体可以对其他较差个体产生较大的吸引力, 使那些个体可以很快地运动至该较优个体的邻域内进行搜索, 从而使得种群会很快地收敛至 Pareto 最优个体周围。

2. 怎样分别从 APO 算法自身特点及多目标优化问题特点两个方面考虑, 来保持 Pareto 最优解集的多样性。如前所述, 种群多样性保持是多目标优化问题的一个重要方面。在 APO 算法中, 由于每个个体除受最好个体的作用, 还要受种群中其他个体的作用, 所以每个个体对于周围环境的信息较为全面, 这样群体具有较好的多样性。将 APO 算法应用于多目标优化领域, 要充分体现其优点, 将其个体间的虚拟力作用作为保持种群多样性的一个重要指标: 作用于其他个体虚拟力之和较大的个体, 必然拥有较大质量或处于个体较为稀疏区域, 故而这样的个体在同等优劣关系的个体中应优先被选择。另一方面, 从多目标优化问题本身特点出发, 由于多目标优化问题往往不会只有一个最优解, 所以希望产生的 Pareto 最优解集分布较为均匀, 这就需要在优化过程中保持种群多样性, 可以通过定义个体共享半径或参照个体间拥挤距离来完成这一要求, 即考查的个体邻域半径内密度较小者较优, 或者个体间拥挤距离较大者较优。

3. 对于具有最优解保留机制的算法, 怎样修剪 *archive* 集合, 从而保证算法找到的 Pareto 前沿光滑均匀分布。从 1.3.3 节可知, 具有最优非支配解保留机制的算法, 需要对 *archive* 集进行修剪, 从而保障该集合中存放的个体具有相同的优劣关系, 并且该集合大小不能超过规定的规模, 对于超出规定大小时需要删除一些个体, 这样就需要一定的策略确定删除哪些个体。由于对于多目标优化问题, 往往需要得到均匀分布的 Pareto 前沿, 所以所处区域密度较大的个体通常被删除, 表现在 MOAPO 算法中, *archive* 集合中那些作用于其他个体上虚拟力之和较小的非支配解被删除。

通过上述分析可知, 只要合理的解决以上关键技术, 构造、设计多目标优化拟态物理学优化算法是完全可行的。

2.3 多目标拟态物理学优化算法基本框架

对于最小化多目标优化问题 (1.1), MOAPO 算法主要包括以下几个部分:

算法初始化, 非支配解的选取, 计算个体所受合力, 个体运动更新, 非支配解集更新, 其框架如图 2.1 所示。其中涉及到的参数说明如下:

dim: 问题空间的维数;

m: 子目标的个数, 亦即目标空间的维数;

PopSize: 种群规模;

MAXITER: 最大迭代次数;

Iter: 当前迭代次数;

archive: 用来存储非支配解的外部集合;

archivesize: 非支配解集大小;

$x_k^{\max}$ 和 $x_k^{\min}$: 问题可行域第 k 维的上界和下界;

$V_{\max}$: 最大速度;

G : 引力常数;

Algorithm MOAPO Framework()

1. **Begin**
 2. **Initialize**
 3. 初始化参数: $PopSize, archivesize, MAXITER, G, x_k^{\max}, x_k^{\min}, V_{\max}$ 等, 计数器 $t=0$;
 4. 随机初始化种群中个体位置和速度;
 5. **End Initialize**
 6. **For** $i=1$ To $PopSize$
 7. 计算每个子目标下个体函数值及总目标下个体函数值;
 8. **End For**
 9. 找出非支配解放入 $archive$ 集合, 找出最优个体函数值 $f(X_{best})$ 和最差个体函数值 $f(X_{worst})$;
 10. **While** ($Iter \neq MAXITER$)
 11. **Do**
 12. 计算种群中每个个体所受合力;
 13. 计算种群中每个个体的速度和位置;
 14. 计算种群中每个个体在每个子目标下函数值, 找出非支配个体, 更新 $archive$ 集合;
 15. 计算种群中每个个体在总目标下个体函数值, 找出最优个体函数值 $f(X_{best})$ 和最差个体函数值 $f(X_{worst})$;
 16. $t=t+1$;
 17. **End While**
 18. 输出 $archive$ 集合中元素在每个子目标下的函数值;
 19. **End.**
-

图 2.1 MOAPO 算法基本框架

2.3.1 算法初始化

MOAPO 算法的初始化包括两部分: 算法参数的初始化和种群的初始化。在算法伊始, 首先要初始化算法中涉及到的参数。指定种群的规模 $PopSize$, 外部存储体 $archive$ 集的规模 $archivesize$, 问题空间的维数 dim , 目标空间的维数 m , 个体运动空间的约束范围 $[x_{\min}, x_{\max}]$, 个体运动速度的约束范围 $[v_{\min}, v_{\max}]$, 引力常数 G , 惯性权

重 w 的变化范围 $[w_{\min}, w_{\max}]$, 最大迭代次数 $MAXITER$, 计数器置零。接下来初始化种群。在初始状态下, 个体在问题的可行域内随机均匀分布, 在运动速度约束范围内随机产生初始速度, 此时个体受到其他个体的虚拟力作用为零, 从而个体受到的合力也为零。

2.3.2 非支配解的选取

在种群初始化之后, 计算每个个体在每个优化子目标下的函数值, 根据前面 1.1 节中的 Pareto 支配概念, 找出非支配个体, 并将其放入到外部存储体 *archive* 集合中。同时计算每个个体在优化的总目标下的函数值, 该值能够从某种程度上反映个体的相对优劣, 从而作用于算法中的质量函数, 使得个体相对越优其质量就越大, 个体相对较差, 其质量就较小。

在这里我们将个体总目标函数表示为 $fitness(x_i) = g(fitness_j(x_i))$, 其中 $fitness_j(x_i)$ 为个体 i 在子目标 j 下的函数值, $fitness(x_i)$ 为个体 i 在总目标下的函数值, $g(\cdot)$ 是将子目标转化成总目标的函数, 该函数的不同导致了不同的 MOAPO 算法, 本文中应用了聚集函数法和序值法来计算总目标下的个体适应值。有关个体适应值函数的研究将在第五章中详细叙述, 在此不再赘述。然后, 根据个体在总目标下适应值的大小, 选出全局最优个体及全局最差个体。

2.3.3 个体所受合力的计算

种群中每个个体受到其他个体的虚拟力作用, 这些虚拟力作用的和就是该个体受到的虚拟力之和。从公式 (2.7) 可知, 个体所受到的其他个体的虚拟力与个体的质量与个体间的距离相关, 个体质量的计算是 MOAPO 算法设计需要解决的一个关键问题, 个体质量要充分体现出性能较好的个体具有较大的质量, 性能相对较差的个体具有较小的质量, 所以个体质量的大小要能够反映出个体的优劣, 这样个体质量就与个体适应值函数的选取及个体间的支配关系息息相关。在本文中, 我们只考虑基于 Pareto 概念的个体间的支配关系, 而适应值函数的选取在第五章中有详细介绍。

2.3.4 个体运动更新及 *archive* 集更新

每个个体在自身惯性及所受到的虚拟力作用下根据公式 (2.9) 及 (2.10) 不断更新自己的速度和位置。对于多目标进化算法, 一般希望在前期算法具有较强的全局搜索能力, 在后期算法具有相对较强的精细搜索能力, 所以在个体运动更新过程中, 惯性权重一般不是一个固定值, 而是随着迭代次数的增加逐渐变化的, 其变化范围限制在 $[w_{\min}, w_{\max}]$ 。

在个体运动更新之后, 需要根据个体在每个子目标下的适应值选取非支配个体, 然后将其与 *archive* 集合中已经存在的非支配个体进行比较, 删除被支配个体, 将非支配个体存储在 *archive* 集合中。有时, 需要存储的非支配个体的数量会超出 *archive* 集合规模, 这时需要对 *archive* 集合进行修剪, 其修剪方法也是 MOAPO 算法设计的一个关键, 可以根据非支配解集的分布性来更新 *archive* 集合, 这方面的研究将在第四章详细叙述。

2.3.5 MOAPO 算法流程

MOAPO 算法步骤如下:

Step1. 初始化种群: 种群个体每一维的初始位置和速度在问题可行域中随机产生;

Step2. 选取非支配解:

Step2.1. 计算个体在每个子目标下适应值, 并找出非支配个体;

Step2.2. 计算每个个体在总目标下的适应值;

Step2.3. 选出全局最优和最差个体;

Step3. 计算个体所受合力:

Step3.1. 根据式(2.6)计算个体质量;

Step3.2. 根据式(2.7)计算个体所受其他个体的虚拟力;

Step3.3. 根据式(2.8)计算个体所受合力;

Step4. 个体运动更新及 *archive* 集更新:

Step4.1. 根据式(2.9)计算个体的下一代速度;

Step4.2. 根据式(2.10)计算个体的下一代位置;

Step4.3. 计算个体在每个子目标下的适应值, 找出非支配个体, 将其与 *archive* 集中个体比较, 更新 *archive* 集, 若非支配个体数量超出 *archive* 集规模, 修剪 *archive* 集合;

Step4.4. 计算个体在总目标下的适应值, 找出最优及最差个体适应值;

Step5. 判断是否满足结束条件, 若满足, 则停止计算, 并输出最优结果; 若不满足, 则返回步骤Step3, 进行下一次迭代。

2.4 本章小结

本章首先简要介绍了 APO 算法及其影响该算法性能的关键因素, 将 APO 算法与目前多目标领域内比较流行的处理方法——PSO 优化算法及遗传算法进行了比较, 分析了 APO 算法与 PSO 算法及遗传算法的相同之处与不同之处, 指出了将 APO 算法引入多目标优化领域需要解决的关键问题, 阐述了将 APO 算法应用于解决多目标优化问题的

可行性。在此基础上建立了 MOAPO 算法的基本框架,并对其中的每一部分进行了简单介绍。

第3章 基于聚集函数法的多目标拟态物理学优化算法

3.1 引言

根据第二章的分析可知,将 APO 算法应用于解决多目标优化问题是可行的,并且确定个体质量是算法的关键。在 APO 算法中,个体质量是由全局最好目标函数值、全局最差目标函数值及当前个体的目标函数值决定。但多目标优化算法不同于单目标优化算法,多目标优化算法是对多个子目标同时进行优化,每次迭代中每个子目标均对应于一个全局最好及最差目标函数值,所以在多目标优化算法的每次迭代中,会产生多个针对于不同子目标的全局最好及最差函数值,而单目标优化算法中只有一个全局最好及最差函数值。因此本章首先借鉴聚集函数法的思想,通过给每个子目标下的全局最好函数值和全局最差函数值赋予随机权重的方式,将多目标优化问题中多个针对不同子目标的全局最好及最差目标函数值转换为单个全局最好和最差目标函数值,然后利用 APO 算法实现对多目标优化问题中 Pareto 最优解集的搜索。最后通过数值实验及与经典多目标进化算法和基于 PSO 的多目标优化算法进行比较,分析了 MOAPO 算法在对真实解的逼近程度及分布性等方面的性能特征,说明了将 APO 应用于多目标优化问题的有效性。

3.2 算法思想与算法流程

3.2.1 适应值函数设计及算法思想

对于基于种群的多目标优化算法,适应值函数的设计是算法的关键,种群中个体的适应值应该能够充分说明个体的好坏。在 APO 算法中,质量是个体所具有的一个属性,是个体好坏的体现,所以个体质量是其适应值大小的表现形式。在 MOAPO 算法中,质量仍然是评价个体优劣的一个重要属性,所以我们将质量函数作为 MOAPO 算法的适应值函数。从公式(2.6)可以看出, APO 算法中个体的质量与全局最好目标函数值、最差目标函数值及个体自身的目标函数值相关,因此怎样将 MOAPO 算法中多个子目标函数值转换成单个的总目标函数值是算法适应值函数设计的关键问题。

聚集函数法是多目标优化中最直接、最简单的方法^[101],通过给多个目标函数赋予权重,将多目标优化问题转换成单目标优化问题。借鉴聚集函数法的基本思想,

MOAPO 算法将多个目标函数下求得的全局最好解和最差解,通过随机选取每个目标权重的方法分别求其“均值”,并求出相应函数值作为更新个体质量的全局最好函数值和全局最差函数值。具体方法如下:

首先,对于种群中的每一个个体,计算其在每个子目标下的函数值 $Fitness[i][j]$ (表示第 j 个个体在第 i 个目标下的函数值),然后给个体在每个目标下的函数值赋予随机权重 r_i ,并且对于 r_i 存在

$$\sum_{i=1}^N r_i = 1 \quad (3.1)$$

其中, N 是子目标的数量。

对于种群中的每一个个体,通过上面公式 (3.1) 的随机权重可以求出其在每个目标下函数值的“均值”,作为该个体 j 在多目标优化问题中的个体函数值 $f(x_j)$,即:

$$f(x_j) = \sum_{i=1}^N (r_i * Fitness[i][j]) \quad (3.2)$$

然后找到每个子目标函数下的全局最好适应值 $f_{best}(i)$ 和全局最差适应值 $f_{worst}(i)$ 及其对应的个体位置 $gbest(i)$ 和 $gworst(i)$ 。对每个 $gbest(i)$ 和 $gworst(i)$ 取随机权重 r_j 和 r_k ,将随机加权作为全局最好解 $gBest$ 和全局最差解 $gWorst$,即

$$gBest = \sum_{j=1}^N (r_j * gbest(i)) \quad (3.3)$$

$$gWorst = \sum_{k=1}^N (r_k * gworst(i)) \quad (3.4)$$

计算 $gbest(i)$ 和 $gworst(i)$ 所对应的每个目标下的全局最好及最差适应值 $BestFit(i)$ 和 $WorstFit(i)$ 。最后利用赋予随机权重的方法分别求得 N 个全局最好适应值及 N 个全局最差适应值的均值 f_{best} 和 f_{worst} ,将其作为多目标优化问题中的全局最好及最差适应值。即

$$f_{best} = \sum_{i=1}^N (r_i * BestFit(i)) \quad (3.5)$$

$$f_{worst} = \sum_{i=1}^N (r_i * WorstFit(i)) \quad (3.6)$$

这样就可以根据公式(3.7)计算个体的质量 m_i 。

$$m_i = e^{\frac{f_{best} - f(x_i)}{f_{worst} - f_{best}}} \quad (3.7)$$

MOAPO 算法的基本思想如下:

对于最小化多目标优化问题 (1.1),首先根据上面描述的适应值函数的设计方法求出种群中每个个体的质量,然后再根据个体质量和个体间每一维距离计算非全局最

优个体在该维上所受到的其他个体的作用力：若个体 i 的适应值大于个体 j 的适应值，即 $m_i > m_j$ ，说明个体 i 要好于个体 j ，则个体 j 对个体 i 有斥力；若个体 i 适应值小于等于个体 j 的适应值，即 $m_i \leq m_j$ ，则个体 j 对个体 i 有引力，即：

$$F_{i,k} = \begin{cases} Gm_i m_j (x_{j,k} - x_{i,k}), & m_j \geq m_i \\ -Gm_i m_j (x_{j,k} - x_{i,k}), & m_j < m_i \end{cases} \quad \forall i \neq j \text{ and } i \neq best \quad (3.8)$$

计算出个体受到其他个体的虚拟力之后，根据公式 (2.8) 求出其他所有个体作用于该个体的虚拟力之和。然后根据公式 (2.9) 和公式 (2.10) 更新个体的速度和位置。

3.2.2 算法流程

以最小化 N 个目标函数为例，给出 MOAPO 算法流程如下：

- Step1 初始化群体：给定群体规模 $PopSize$ ，随机产生每个个体的初始位置和初始速度。
- Step2 计算每个个体在每个子目标下的函数值，再利用公式 (3.2) 计算每个个体目标函数值的“均值”。
- Step3 对于 N 个子目标，分别找出每个子目标下的全局最好函数值及其相对应的个体，每个子目标下全局最差函数值及其相对应的个体。
- Step4 根据公式 (3.3) 及公式 (3.4) 计算全局最好个体及全局最差个体，并计算其在每个子目标下函数值，根据公式 (3.5) 及公式 (3.6) 计算全局最好函数值和最差函数值。
- Step5 利用公式 (3.7) 计算每个个体的质量。
- Step6 利用公式 (3.8) 计算每个个体每一维受到的其他个体的作用力，根据公式 (2.8) 计算个体在该维上所受到的合力。
- Step7 用得到的质量 m_i 和合力 $F_{i,k}$ 根据公式 (2.9) 和 (2.10) 更新个体的速度和位置。
- Step8 若达到最大迭代次数则退出，否则返回 Step2。

3.3 数值实验及结果分析

为了充分说明 MOAPO 算法的有效性，我们利用 MOAPO 算法对一些标准的多目标测试函数进行了实验，利用 MOAPO 算法与文献[79]中提出的 MOPSO 算法以及文献[38]的 NSGA-II 算法进行对比研究。测试函数选择文献[81]中的测试函数 1 (Schaffer1)、测试函数 2 (Schaffer2) 和文献[79]中的测试函数 2 (ZDT1)、测试函数 3 (ZDT2)、测试函数 5 (ZDT3)，如表 3.1 所示。其中，Schaffer1 函数和 Schaffer2 函数主要对多目标优化算法的有效性进行测试，ZDT1、ZDT2 和 ZDT3 的决策变量的

维数为 30 维，分别对多目标优化算法解决 Pareto 前沿是凸、非凸和不连续的优化问题的能力进行测试。3 种算法的群体规模均为 50，迭代次数均为 100 次。每个算法对每个测试函数分别进行 30 次独立实验，将第 15 次的实验结果以图形形式输出。

表 3.1 基于聚集函数法 MOAPO 算法的测试函数

Problem	Variable bounds	Objective functions
Schaffer1	$x \in [-5, 7]$	$\min f_1(x) = x^2$ $\min f_2(x) = (x-2)^2$
Schaffer2	$x \in [-5, 10]$	$\min f_1(x) = \begin{cases} -x, & \text{if } x \leq 1, \\ -2+x, & \text{if } 1 < x \leq 3, \\ 4-x, & \text{if } 3 < x \leq 4, \\ -4+x, & \text{if } x > 4 \end{cases}$ $\min f_2(x) = (x-5)^2$
ZDT1	$x = (x_1, x_2, \dots, x_m)$ $x_i \in [0, 1]$ $i = 1, 2, \dots, m$ $m = 30$	$\min f_1(x) = x_1$ $\min f_2(x) = g(1 - \sqrt{f_1/g})$ $g(x_2, \dots, x_m) = 1 + 9 \sum_{i=2}^m x_i / (m-1)$
ZDT2	$x = (x_1, x_2, \dots, x_m)$ $x_i \in [0, 1]$ $i = 1, 2, \dots, m$ $m = 30$	$\min f_1(x) = x_1$ $\min f_2(x) = g(1 - \sqrt{f_1/g})$ $g(x) = 1 + 9 \sum_{i=2}^m x_i / (m-1)$
ZDT3	$x = (x_1, x_2, \dots, x_m)$ $x_i \in [0, 1]$ $i = 1, 2, \dots, m$ $m = 30$	$\min f_1(x) = x_1$ $\min f_2(x) = g(1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1))$ $g(x_2, \dots, x_m) = 1 + 9 \sum_{i=2}^m x_i / (m-1)$

MOAPO 算法中，惯性权重 w 表明个体的历史速度信息对当前个体速度的影响， w 较大，则全局搜索性能较强； w 较小，则个体的局部搜索性能较好。通常在算法初期往往需要较强的全局搜索能力，以便能够快速地搜索到全局最优解的周围，而算法后期，种群中的个体往往聚集于最优解周围，此时需要进行精细搜索，所以需要较强的局部搜索能力。为了平衡个体的全局搜索能力和局部搜索能力，MOAPO 算法中惯性权重 w 随着迭代的进行动态下降，但当迭代到一定次数时（本算法中是到最大次数

的 3/4) 停止减小。即

$$w(iter) = w_i - \frac{iter-1}{Maxit} * (w_i - w_f) \quad (3.9)$$

其中 $iter$ 是迭代次数, $Maxit$ 是最大迭代次数, w_i 是初始值, 在本算法中为 0.9, w_f 是终值, 在本算法中为 0.4。

MOAPO 算法中, 在计算个体所受到的力时要用到另一个参数 G , 在单目标的 APO 算法中, G 通常取常数 10。为了更好的平衡算法的全局和局部搜索能力, G 应随种群的进化而动态变化, 本算法中 G 的表达式为

$$G(iter) = G_i - \frac{iter-1}{Maxit} * (G_i - G_f) \quad (3.10)$$

其中 G_i 与 G_f 分别表示 G 的初值和终值。由于 w 初值较大, 所以为了平衡全局与局部搜索能力 G_i 也较大, 本算法中取经验值为 70, 当迭代进行到最后时, G 达到终值 1。

从前面 1.3.3 小节的介绍可知, 对于多目标优化算法的评价是通过逼近真实解程度与解集分布性方面来衡量算法的性能。多目标优化算法中常用的 GD (Generation Distance) 和 SP (Spacing) 来衡量算法对真实解的逼近程度和 Pareto 解集的分布性。本文采用这两个性能指标对算法性能进行对比。 GD 和 SP 的定义在公式 (1.13) 和公式 (1.14) 中已进行描述, 在此不再赘述。 GD 越小, 说明通过算法得到的非劣最优解集越逼近真实解集; SP 越小, 说明算法得到的非劣最优解集分布性越好。

图 3.1 至图 3.5 为 MOAPO, MOPSO 和 NSGA-II 对 5 个测试函数生成的 Pareto 前沿与真实 Pareto 前沿的对比图。表 3.2 与表 3.3 分别为 3 种算法关于 5 个测试函数的 GD 与 SP 值。

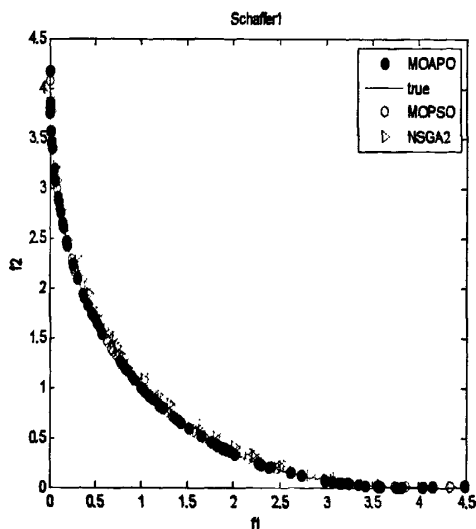


图 3.1 Schaffer 1 的 Pareto 前沿

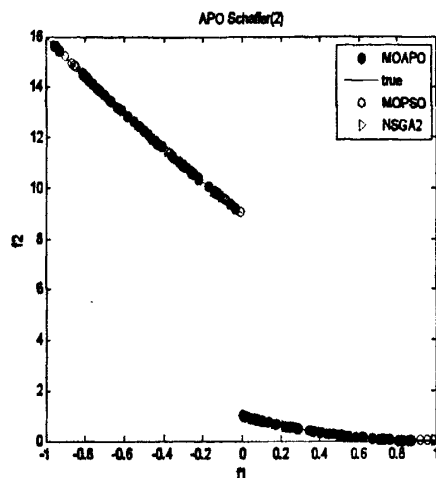


图 3.2 Schaffer 2 的 Pareto 前沿

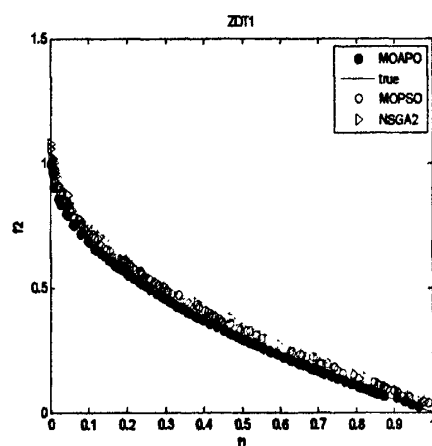


图 3.3 ZDT1 的 Pareto 前沿

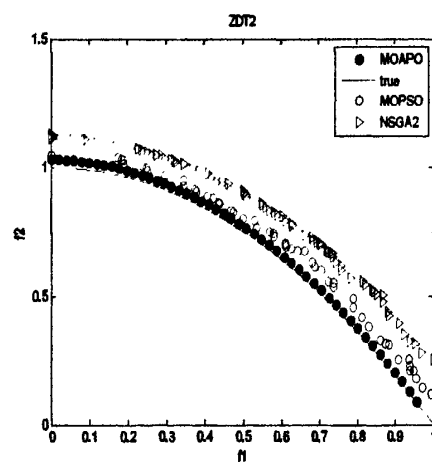


图 3.4 ZDT2 的 Pareto 前沿

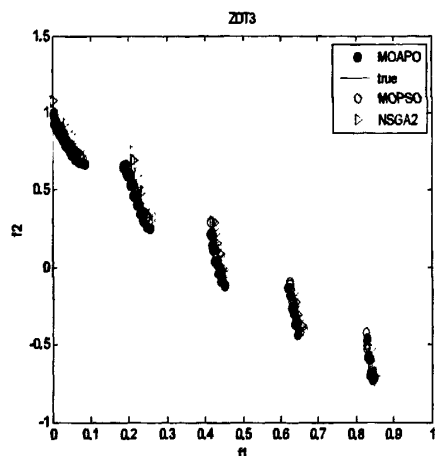


图 3.5 ZDT3 的 Pareto 前沿

从图 3.1 可以看出，对于 Schaffer(1)函数，3 种算法中 MOAPO 和 MOPSO 的解基本能覆盖真实 Pareto 前沿，由表 3.2 与表 3.3 可以看出，MOPSO 的 *GD* 值较小，但 MOPSO 的解的分布性较差；而 MOAPO 的 *GD* 值略大于 MOPSO 的 *GD* 值，比 NSGA-II 的 *GD* 值要好，但其解的分布性要比其他两种算法的解的分布性都要好。从图 3.2 可以看出，对于 Schaffer (2) 函数，3 种算法中 MOAPO 的解基本覆盖了真实 Pareto 前沿，无论是对真实 Pareto 前沿的逼近程度还是解集分布性都要好于 MOPSO 算法和 NSGA2，通过表 3.2 与表 3.3 我们也可以得到同样的结论。

对于 ZDT1, ZDT2 和 ZDT3 函数，从图 3.3 至图 3.5 中可以看出 MOAPO 算法得到的 Pareto front 无论从对真实 Pareto 前沿的逼近程度还是解集分布性，都要好于其他两种算法得到的 Pareto 前沿，从表 3.2 与表 3.3 中也可以看出，MOAPO 的 *GD* 值与 *SP* 值均要好于其他两种算法的相应指标，因此 MOAPO 算法对真实解集的逼近程度与 Pareto 解集的分布性要更好。

通过 5 个典型的多目标测试函数的优化实验可知，与 MOPSO 与 NSGA-II 相比，MOAPO 算法能更好地收敛到真实 Pareto 前沿，并且得到的解集分布更加均匀。

表 3.2 基于聚集函数法 MOAPO 测试函数的 *GD* 值比较

测试函数	GD	MOAPO	MOPSO	NSGA-II
Schaffer(1)	mean	0.0995	0.0962	0.1453
	best	0.0855	0.0758	0.0925
	std.	0.3066	0.0208	0.0094

Schaffer(2)	mean	0.0378	0.0488	0.0967
	best	0.0207	0.0322	0.0431
	Std.	0.0377	0.0392	0.0766
ZDT1	mean	0.0176	0.0553	0.1253
	best	0.0052	0.0502	0.0896
	std.	0.0190	0.0040	0.0155
ZDT2	mean	0.0528	0.0931	0.2159
	best	0.0055	0.0640	0.1497
	std.	0.0104	0.0091	0.0232
ZDT3	mean	0.0162	0.0271	0.0443
	best	0.0051	0.0077	0.0097
	std.	0.0163	0.0137	0.0112

表 3.3 基于聚集函数法 MOAPO 测试函数的 SP 值比较

测试函数	SP	MOAPO	MOPSO	NSGA-II
Schaffer(1)	mean	0.2004	1.2684	0.2143
	best	0.0338	0.4974	0.1067
	std.	0.0532	0.5987	0.0098
Schaffer(2)	mean	0.0741	0.2068	0.3075
	best	0.0552	0.1001	0.1069
	std.	0.0107	0.1377	0.2149
ZDT1	mean	0.0089	0.0171	0.0218
	best	0.0057	0.0088	0.0155
	std.	0.0025	0.0065	0.0031
ZDT2	mean	0.0106	0.0490	0.0374
	best	0.0031	0.0088	0.0217
	std.	0.0020	0.0119	0.0085
ZDT3	mean	0.0079	0.0084	0.0222
	best	0.0055	0.0067	0.0166
	std.	0.0014	0.0037	0.0033

3.4 本章小结

本章借鉴聚集函数的思想提出了一种多目标拟态物理学优化算法。首先对算法的适应值函数进行了设计,通过赋予随机权重的方法计算种群中每个个体的目标函数值以及全局最优个体的目标函数值和全局最差个体的目标函数值,从而得到个体的质量,然后利用 APO 算法对最小化多目标问题进行求解。在仿真实验中,讨论了惯性权重 w 和引力常数 G 的设计方法。通过与经典多目标进化算法以及多目标微粒群算法的比较可知,该算法具有较好的对真实 Pareto 前沿的逼近性能和较为均匀的解集分布性,充分说明了 MOAPO 算法的有效性。

第4章 基于虚拟力排序的多目标拟态物理学优化算法

4.1 引言

从前面两章的分析可知,将 APO 算法应用于解决多目标优化问题是可行的,通过对 MOAPO 算法的数值实验及与其他经典 MOEAs 和 MOPSO 算法的比较可知,用 MOAPO 算法来解决多目标优化问题是有效的。但该算法没有采用外部存储体来保留进化过程中生成的非劣解,没有采用有效方法来保持解集分布性,也没有对算法的收敛性从理论上进行分析。从第一章对多目标进化算法分析可知,基于外部存储体的非劣解保留机制是多目标进化算法的一个成功经验,可以大大提高 Pareto 解集质量。解集的分布性更是衡量多目标优化算法的一个重要指标,多目标优化算法的目标就是找到一组分布均匀的、尽可能逼近真实 Pareto 解集的集合。

现有的 MOEAs 及 MOPSO 算法通常采用 *archive* 集来保存当前搜索到的非劣解,利用非劣解集中的密度信息对非劣解集进行裁剪,以保持非劣解集良好的分布性,并从非劣解中选取全局最优解 g_{best} 。有关种群多样性技术在前面 1.3.2 节及 1.4.2 节分别有详细介绍,在此不再赘述。密度信息的获取往往通过随机选择^[71]、小生境技术^[72,73]、网格技术^[62,74,87,90]或聚类技术^[91]等来实现。显然,随机搜索的效率较低,并且所得解的多样性不是很好。小生境技术参数的确定较为困难;利用网格技术获得密度信息与网格的大小和数量息息相关,若每个网格中包含的微粒数都为 1 时,此时大多数使用这类方法的算法中非劣解集的修剪与全局最优解的选取都退化为随机搜索;聚类技术的计算复杂度较高。

针对以上问题,本章首先结合 APO 算法自身特点,从虚拟力的角度分析了 MOAPO 算法中个体的优劣,提出了一种基于虚拟力排序的 MOAPO (VFMOAOP) 算法。该算法利用快速排序思想,根据非劣解集中的个体作用于其他个体的虚拟力之和的大小进行排序,以此来保持 Pareto 最优解集的分布性,并将作用于其他个体的虚拟力之和的最大者作为全局最优个体。然后基于概率论的基本理论,对 VFMOAPO 算法的收敛性进行了理论分析与证明。最后通过数值实验说明了该算法的有效性。

4.2 算法思想及算法流程

APO 算法中,个体的质量 m 不是一个固定值,而是在每次迭代过程中变化的, m 越大说明该个体越好^[99]。同时,根据 APO 算法中虚拟力的计算公式可知,一个个体

作用于其他个体上的虚拟力的大小除与质量相关外还与两个个体间的距离相关, 距离越远, 两个个体间的作用力就越大。根据以上原则可知, 一个个体*i*对其他个体的作用力之和如果越大, 说明要么是该个体的质量大, 要么是该个体距离其他个体较远。如果是个体质量大, 那么说明该个体位置较优; 如果是个体距离其他个体较远, 说明该个体所处位置的个体密度较小。多目标优化问题中, 对于非劣解集的修剪往往根据个体的密度信息, 选择密度较小的个体进行保留, 而对密度较大个体进行裁剪, 并且对于全局最优解的选取一般也选择密度较小个体, 这样可以避免群体收敛于一点, 从而保持非劣解集的分布性。本章提出一种 VFMOAPO 算法, 该算法结合多目标优化问题特点, 利用 APO 算法本身特性, 根据个体作用于其他个体的虚拟力之和的大小进行排序, 进而修剪非劣解集和选取全局最优解。

4.2.1 算法思想

设 P 为进化群体, 初始化时随机产生 n 个个体, $archive$ 为非劣解集, 大小为 n , 初始时空。由于利用 APO 算法求解多目标优化问题时还要用到全局最差适应值, 所以这里再引入一个被支配解集合 Q , 大小为 n , 初始时空。挑选出 P 中的非支配解放入到 $archive$ 集中, 由于初始状态下个体所受到其他个体的作用力都为 0, 所以首先从 $archive$ 集中选择所有目标下的函数值的随机加权值最小的个体作为全局最优个体。挑选出 P 中的被支配解, 放入到 Q 中, 选择所有目标下的函数值的随机加权值最大的个体作为全局最差个体。根据公式 (2.6) 计算个体 i 的质量, 根据公式 (2.7) 计算在第 k 维上个体 i 受到个体 j 的作用力, 根据公式 (2.8) 计算在第 k 维上个体 i 所受到的其他所有个体的合力。在第 k 维上个体 i 作用于个体 $j(j \neq best)$ 的虚拟力为

$$F_{j,k} = -F_{i,k}, \quad i \neq j \text{ and } j \neq best \quad (4.1)$$

在第 k 维上, 个体 i 作用于其他所有个体的虚拟力之和为

$$F_{i,k} = \sum_{j=1}^n F_{j,k}, \quad i \neq j \text{ and } j \neq best \quad (4.2)$$

个体 i 作用于其他所有个体上的虚拟力之和大小为

$$F(i) = \sqrt{F_{i,1}^2 + F_{i,2}^2 + \dots + F_{i,k}^2} \quad (4.3)$$

利用快速排序方法, 根据个体作用于 P 中其他个体的虚拟力之和的大小对 $archive$ 集中的个体按照降序关系进行排序。这样作用于其他个体上虚拟力之和越大的个体就排在越前面, 作用于其他个体上虚拟力之和越小的个体就排在越后面, 因此将 $archive$ 集中的第一个个体作为全局最优个体 $gbest$ 。然后考查集合 Q 中的元素, 该集合中的个体都是被支配的, 要劣于 $archive$ 集中的个体, 所以全局最差个体就从该集合中产生。同样根据快速排序的思想和个体作用于 P 中其他个体的虚拟力之和的

大小对集合 Q 中的个体按照升序关系进行排序, 这样 Q 集合中越差的个体就排在越前面, 所以将该集合中的第一个个体作为全局最差个体 g_{worst} 。接下来在每个目标下找到全局最优个体 g_{best} 和全局最差个体 g_{worst} 对应的函数值, 利用给每个目标赋予随机权重的方法得到总目标下的全局最优个体的适应值 f_{best} 、全局最差个体的适应值 f_{worst} 以及每个个体在总目标下的个体适应值 $f(x_j)$ 。即

$$f(x_j) = \sum_{i=1}^m (r_i * f_i(x_j)) \quad (4.4)$$

其中, m 为子目标个数。然后再根据公式 (2.9) 和公式 (2.10) 更新个体速度和位置。

更新了个体的速度和位置之后, 再计算每个目标下个体的函数值, 从而确定哪些个体是非支配解, 计算该个体作用于其他个体的虚拟力之和, 将其与 $archive$ 集中的非支配解作用于其他个体的虚拟力之和进行比较, 然后插入到 $archive$ 集中相应的位置上。如果当前非支配解的个数大于 $archive$ 集大小, 则需要对 $archive$ 集进行修剪。作用在其他个体上虚拟力之和较小的个体, 即 $archive$ 集中排在后面的个体, 要么是个体质量较小, 要么处于个体密度较大的区域, 所以这样的个体将被裁剪掉, 而裁剪的数量根据当前非支配解个数超出 $archive$ 集大小的情况而定。根据同样道理修剪被支配解集合 Q 。 $archive$ 集合与 Q 集合的第一个个体就是全局最优及全局最差个体。

4.2.2 算法流程

对于最小化多目标优化问题, VFMOAPO 算法流程如下:

- Step1 初始化: 给定群体 P 的规模 n , $archive$ 集合大小 n , 集合 Q 大小 n , 最大迭代次数 $Maxit$, 随机产生每个个体的初始位置 $popul$ 和初始速度 vel , 个体间作用力 $F_{g,k} = 0$, $archive$ 集和集合 Q 均为空。
- Step2 计算每个个体在每个子目标下的函数值(用 $f_i(x_j)$ 表示第 i 个目标下第 j 个个体的函数值), 然后根据公式 (4.4) 计算每个个体在总目标下的函数值 $f(x_j)$ 。
- Step3 根据 Pareto 支配概念将非支配解放入 $archive$ 集中, 并将其中总目标下最小的函数值作为全局最优函数值 f_{best} ; 将被支配解放入 Q 集合中, 将其中总目标下最大的函数值作为全局最差函数值 f_{worst} 。
- Step4 利用公式 (3.7) 计算个体质量, 利用公式 (3.8)、公式 (4.1) 和公式 (4.2) 计算个体 i 在第 k 维上作用于其他个体的虚拟力之和 F_{ik} , 并根据公式 (4.3) 计算个体 i 作用于所有其他个体的虚拟力之和 $F(i)$ 。
- Step5 根据 $F(i)$ 大小, 利用快速排序的思想将 $archive$ 集合按降序关系进行排序, 将 Q 集合按升序关系进行排序。
- Step6 将 $archive$ 集中第一个个体作为全局最优个体, 求出其总目标下相应函数值

f_{best} , 将 Q 集合中第一个个体作为全局最差个体, 求出其总目标下相应的函数值 f_{worst} , 用公式 (3.7) 计算个体的质量 m_i , 用公式 (3.8)、公式 (4.1) 和公式 (4.2) 计算其受到其他个体的合力 $F_{i,k}$, 根据公式 (2.9) 和公式 (2.10) 更新个体的速度和位置。

Step7 根据公式 (4.4) 计算每个个体在总目标下的函数值, 对产生的非支配个体与 $archive$ 集中的个体比较, 根据作用于 P 中其他个体的虚拟力之和的大小将非支配个体插入相应位置, 若 $archive$ 集溢出, 则将 $archive$ 集中的最后面的元素删除。根据同样方法更新 Q 集合。

Step8 若达到最大迭代次数则退出, 否则返回 Step6。

4.3 算法收敛性证明

目前多目标优化问题的研究重点在于算法的设计和数值实验效果的比较上, 理论研究成果相对较少。并且现有的大多数研究只是沿袭了单目标优化算法的理论模式和研究成果, 忽视了多目标优化算法自身蕴含的特征。在理论研究方面, 比较具有代表性的研究有: 德国学者 Rudolph 对两个目标优化的进化算法的收敛行为进行了分析^[103], 但该分析过程较为复杂, 并且很多细节源于他对单目标遗传算法收敛性的理论分析结果; Hanne^[104]研究了实函数多目标进化算法的收敛性, 给出了多目标进化算法以概率 1 收敛的收敛性定理, 但该定理要求被优化的每个目标函数必须连续, 对搜索空间要求必须是非空整数闭区间, 目标函数必须是凸函数等, 而大多数算法并无法满足这一要求。近几年国内一些学者采用有限齐次 Markov 链作为分析工具, 对多目标优化问题的收敛性分析做了相关研究^[105,106], 也有极少数用概率论对多目标优化问题的收敛性进行分析^[107], 但这些研究都是针对多目标进化算法进行的, 引入到多目标领域的其他新的优化算法, 大多数都没有对其收敛性进行理论分析, 而进行必要的收敛性能的分析能够为设计高效的算法提供理论保障。

4.3.1 相关收敛性理论

设 $pop(n)$ 是 VFMOAPO 算法产生的目标空间群体序列, 即第 n 次迭代后 $archive$ 集中的元素序列。将目标空间网格化为若干个 m 维超长方体, 记为 $B_i (i=1, 2, \dots, s)$, s 为超长方体的个数, 超长方体主要是为了方便表述而引入的, 算法采用虚拟力排序来保持 Pareto 前沿分布性, 每一个超长方体内的 Pareto 最优解的个数不能超过 1。可以通过调整 s , 使得上述情况成立。 B_i 包含多目标优化问题的 Pareto 前沿 F_i , 设 $N_\varepsilon(F_i)$ 为 F_i 的 ε 邻域。

定义 4.1: 若 $\forall \varepsilon > 0, \forall n \geq m, \lim_{m \rightarrow \infty} P\{pop(n) \cap N_\varepsilon(F_i) \neq \emptyset\} = 1, 1 \leq i \leq s$, 则称 VFMOAPO 产生的群体 $pop(n)$ 强收敛于 F_i , 简称 VFMOAPO 强收敛于其 Pareto 前沿。

定义 4.2: 如果 $\forall \varepsilon > 0, \lim_{n \rightarrow \infty} P\{pop(n) \cap N_\varepsilon(F_i) \neq \emptyset\} = 1, 1 \leq i \leq s$, 则称 VFMOAPO 弱收敛于其 Pareto 前沿。

引理 4.1(Borel-Cantelli 引理): 对于任意的事件序列 $\{X_n\}$ 有

$$\sum_{n=1}^{\infty} P(X_n) < \infty \Rightarrow P\left(\bigcap_{m=1}^{\infty} \bigcup_{n \geq m} X_n\right) = 0 \quad (4.5)$$

4.3.2 算法收敛性

为了方便描述, 将迭代过程中产生的非劣解与 Pareto 前沿间的几种关系记为:

$$\alpha_n = P\{pop(n+1) \cap N_\varepsilon(F_i) = \emptyset | pop(n) \cap N_\varepsilon(F_i) \neq \emptyset\} \quad (4.6)$$

$$\beta_n = P\{pop(n+1) \cap N_\varepsilon(F_i) = \emptyset | pop(n) \cap N_\varepsilon(F_i) = \emptyset\} \quad (4.7)$$

$$\gamma_n = \beta_0 \beta_1 \dots \beta_n \quad (4.8)$$

对于第 $n+1$ 次迭代后产生的 *archive* 集不收敛的情况有两种: 第 n 次迭代产生的 *archive* 集收敛和第 n 次迭代产生的 *archive* 集不收敛, 即 α_n 与 β_n 中的事件序列。

定理 4.1: $\forall \varepsilon > 0$, 若 $\lim_{n \rightarrow \infty} \gamma_n = 0$, 则 $\lim_{n \rightarrow \infty} P\{pop(n) \cap N_\varepsilon(F_i) \neq \emptyset\} = 1$, 即 VFMOAPO

弱收敛于其 Pareto 前沿: 若 $\sum_{n=1}^{\infty} \gamma_n < \infty$, 则 $P\left(\bigcap_{m=1}^{\infty} \bigcup_{n \geq m} pop(n) \cap N_\varepsilon(F_i) = \emptyset\right) = 0$, 即

VFMOAPO 强收敛于其 Pareto 前沿。

证明:

$$\begin{aligned} & P\{pop(n+1) \cap N_\varepsilon(F_i) = \emptyset\} \\ &= P\{pop(n+1) \cap N_\varepsilon(F_i) = \emptyset | pop(n) \cap N_\varepsilon(F_i) \neq \emptyset\} * P\{pop(n) \cap N_\varepsilon(F_i) \neq \emptyset\} \\ &+ P\{pop(n+1) \cap N_\varepsilon(F_i) = \emptyset | pop(n) \cap N_\varepsilon(F_i) = \emptyset\} * P\{pop(n) \cap N_\varepsilon(F_i) = \emptyset\} \\ &= \alpha_n * P\{pop(n) \cap N_\varepsilon(F_i) \neq \emptyset\} + \beta_n * P\{pop(n) \cap N_\varepsilon(F_i) = \emptyset\} \end{aligned} \quad (4.9)$$

由于 VFMOAPO 采用了最优解保留机制, 所以在第 n 代收敛于其 Pareto 前沿, 不会在第 $n+1$ 代时不收敛, 所以显然 $\alpha_n = 0$ 。

$$\begin{aligned} (4.9) \text{式} &= \beta_n * P\{pop(n) \cap N_\varepsilon(F_i) = \emptyset\} \\ &= \beta_n * \beta_{n-1} * P\{pop(n-1) \cap N_\varepsilon(F_i) = \emptyset\} \\ &= \beta_n * \beta_{n-1} * \dots * \beta_1 * \beta_0 * P\{pop(0) \cap N_\varepsilon(F_i) = \emptyset\} \\ &= \gamma_n * P\{pop(0) \cap N_\varepsilon(F_i) = \emptyset\} \end{aligned} \quad (4.10)$$

$$\because \lim_{n \rightarrow \infty} \gamma_n = 0$$

$$\therefore \lim_{n \rightarrow \infty} P\{pop(n+1) \cap N_s(F_i) = \emptyset\}_n = 0$$

$$\text{即 } \lim_{n \rightarrow \infty} P\{pop(n) \cap N_s(F_i) \neq \emptyset\} = 1,$$

VFMOAPO 弱收敛于其 Pareto 前沿。

$$\because P\{pop(n+1) \cap N_s(F_i) = \emptyset\} = \gamma_n * P\{pop(0) \cap N_s(F_i) = \emptyset\}$$

且根据假设 $\sum_{n=1}^{\infty} \gamma_n < \infty$,

由于 $P\{pop(0) \cap N_s(F_i) = \emptyset\} \in [0, 1]$

$$\text{所以 } \sum_{n=1}^{\infty} (\gamma_n * P\{pop(0) \cap N_s(F_i) = \emptyset\}) < \infty$$

$$\text{即 } \sum_{n=1}^{\infty} P\{pop(n+1) \cap N_s(F_i) = \emptyset\} < \infty$$

根据引理 4.1 可知 $P\{\bigcap_{m=1}^{\infty} \bigcup_{n \geq m} pop(n) \cap N_s(F_i) = \emptyset\} = 0$, 即 VFMOAPO 强收敛于其

Pareto 前沿。

证毕。

定理 4.2: 若 VFMOAPO 中 $\lambda \sim N(0, 1)$, 多目标优化问题的决策空间 $\aleph$ 是 R^n 中的紧集, 目标函数 $f(X)$ 是目标空间 R^m 上的连续函数, 则 VFMOAPO 强收敛于其 Pareto 前沿。

证明:

由个体的速度和位置进化方程可得,

$$\Delta x = x(t+1) - x(t) = wV + \lambda F / m \quad (4.11)$$

令 $wV = \mu$, $F/m = \sigma$, 则

$$\Delta x = \mu + \lambda \sigma \quad (4.12)$$

其中, μ, σ 均为常数项。由于 λ 是服从正态分布的随机变量, 那么正态分布的线性组合也服从正态分布。因此

$$\Delta x \sim N(\mu, \sigma^2) \quad (4.13)$$

且 Δx 的 n 个分量 $\Delta x_1, \Delta x_2, \dots, \Delta x_n$ 之间相互独立。

设 $y_0 = f(X_0)$ 为 B_i 中 Pareto 前沿上的点, 其中 $X_0 = (x_0^1, x_0^2, \dots, x_0^n) \in \aleph$

$\because f(X)$ 是连续函数

$$\therefore \exists r > 0, \text{ 当 } \|X - X_0\|_{\infty} \leq r \text{ 时有 } \|f(X) - f(X_0)\|_{\infty} \leq \varepsilon$$

$$\therefore f(X) \in N_s(F_i)$$

记 $A_{x_0, r} = \{X \in \aleph \mid \|X - X_0\|_{\infty} \leq r\}$, 对于 $X = (x_1, x_2, \dots, x_n) \in \aleph$

$$\begin{aligned}
 & P\{(X + \Delta X) \in A_{x_0, r}\} \\
 &= \prod_{i=1}^n P\{|x'_i + \Delta x'_i - x'_0| \leq r\} \\
 &= \prod_{i=1}^n P\{x'_0 - x'_i - r \leq \Delta x'_i \leq x'_0 - x'_i + r\} \\
 &= \prod_{i=1}^n \int_{x'_0 - x'_i - r}^{x'_0 - x'_i + r} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(y - \mu_i)^2}{2\sigma_i^2}} dy
 \end{aligned}$$

所以 $0 < P\{(X + \Delta X) \in A_{x_0, r}\} < 1$

$\therefore P\{(X + \Delta X) \in A_{x_0, r}\}$ 是紧集 $\mathbb{N}$ 上的连续函数

$\therefore \exists X'_1, X'_0 \in \mathbb{N}$,

使得 $P\{(X'_1 + \Delta X') \in A_{x_0, r}\} = \min P\{(X + \Delta X) \in A_{x_0, r}\}$ 且 $0 < P\{(X'_1 + \Delta X') \in A_{x_0, r}\} < 1$

所以 $P\{(X + \Delta X) \in A_{x_0, r}\} \geq P\{(X'_1 + \Delta X') \in A_{x_0, r}\} = \min P\{(X + \Delta X) \in A_{x_0, r}\}$

$\therefore P\{pop(n+1) \cap N_\varepsilon(F_l) \neq \emptyset | pop(n) \cap N_\varepsilon(F_l) = \emptyset\} \geq P\{(X'_1 + \Delta X') \in A_{x_0, r}\}$

$\beta_n = P\{pop(n+1) \cap N_\varepsilon(F_l) = \emptyset | pop(n) \cap N_\varepsilon(F_l) = \emptyset\} \leq 1 - P\{(X'_1 + \Delta X') \in A_{x_0, r}\} = c$

$c \in (0, 1)$

$$\sum_{n=1}^{\infty} \gamma_n = \beta_0 + \beta_0\beta_1 + \dots + \beta_0\beta_1\dots\beta_n \leq c + c^2 + \dots + c^n = \frac{c}{1-c} < \infty$$

对于 $\forall \varepsilon > 0$, 由定理 1 可知, VFMOAPO 算法强收敛于其 Pareto 前沿。

证毕。

4.4 数值实验及结果分析

为了测试 VFMOAPO 算法的性能, 我们利用上述算法对三个典型的多目标测试函数进行了实验, 用 VFMOAPO 算法与文献[90]中提出的 MOPSO 算法以及经典的多目标进化算法 (NSGA, SPEA) 针对算法的收敛性和分布性通过仿真实验进行对比研究。

本章采用多目标优化算法中的典型测试函数 ZDT1, ZDT2 和 ZDT3 进行测试, 这三个测试函数的决策变量维数均为 30, 分别用来测试算法解决 Pareto 前沿是凸、非凸和不连续的优化问题的能力。实验中群体 P 和 *archive* 集的规模设置均为 100, 根据公式 (3.9), w 从初值 $w_i = 0.9$ 线性递减, 终值 $w_f = 0.9$, 当迭代次数达到最大迭代

次数的 75%时, w 停止变化。根据公式 (3.10), G 从初值 $G_i = 70$ 线性递减, 终值 G_f 取值为 5, 进化代数 为 50 代 (比较算法中的进化代数为 5000 代), 这样可以大大减少算法的计算量, 从而提高算法效率。图 4.1 至图 4.3 是利用 VFMOAPO 算法及其他三种经典算法对每个测试函数进行测试的 Pareto 前沿, 表 4.1 至表 4.6 是分别对三个测试函数进行了 30 次独立实验的评价结果。其中 NSGA、SPEA 的计算结果来源于资料[102], MOPSO 结果来源于文献[90]。

从图 4.1 至图 4.3 可以看出利用 VFMOAPO 对这三个测试函数进行多目标优化得到的解基本上都在真实的 Pareto 前沿上, 并且分布均匀, 特别是 ZDT2 函数的 Pareto front, 要明显好于其他各个算法得到的 Pareto 前沿。从表 4.1 至表 4.6 中可以看出, 对于 ZDT1 和 ZDT3 函数, VFMOAPO 算法得到的性能指标 GD 比 NSGA 和 SPEA 算法要好, 与 MOPSO 算法中的 GD 相差不多, 但对于这两个测试函数, 本算法得到的性能指标 SP 要远远好于其他三种算法。对于测试函数 ZDT2, VFMOAPO 算法的性能指标 GD 和 SP 均要好于其他三种算法。所以 VFMOAPO 算法在处理多目标优化问题中 Pareto 前沿是凸、非凸和不连续的优化问题时有较好的性能, 要优于其他几种经典多目标进化算法及多目标微粒群算法。

表 4.1 VFMOAPO 及比较算法对测试函数 ZDT1 的 GD 结果

GD	MOAPO	MOPSO	NSGA	SPEA
Best	0.0008	0.0016	0.1453	0.0893
Worst	0.0045	0.0048	0.2140	0.1181
Average	0.0018	0.0022	0.1715	0.0134
Median	0.0035	——	0.2140	0.1064
Std.	8.437e-4	——	0.0155	0.0066

表 4.2 VFMOAPO 及比较算法对测试函数 ZDT1 的 SP 结果

SP	MOAPO	MOPSO	NSGA	SPEA
Best	1.2864e-5	0.0084	0.0300	0.0155
Worst	0.0164	0.0203	0.0603	0.0295
Average	0.0049	0.0120	0.0409	0.0218
Median	0.0055	——	0.0512	0.0227
Std.	0.0050	——	0.0071	0.0031

表 4.3 VFMOAPO 及比较算法对测试函数 ZDT2 的 *GD* 结果

GD	MOAPO	MOPSO	NSGA	SPEA
Best	0.0027	0.0276	0.2159	0.0889
Worst	0.0042	0.0667	0.3100	0.1340
Average	0.0034	0.0509	0.2668	0.1088
Median	0.0028	—	0.3019	0.0961
Std.	4.1979e-4	—	0.0232	0.0124

表 4.4 VFMOAPO 及比较算法对测试函数 ZDT2 的 *SP* 结果

SP	MOAPO	MOPSO	NSGA	SPEA
Best	1.1974e-6	0.0117	0.0337	0.0217
Worst	0.0113	0.0218	0.1525	0.0573
Average	0.0037	0.0151	0.0725	0.0374
Median	0.0031	—	0.0830	0.0349
Std.	0.0032	—	0.0269	0.0085

表 4.5 VFMOAPO 及比较算法对测试函数 ZDT3 的 *GD* 结果

GD	MOAPO	MOPSO	NSGA	SPEA
Best	0.0034	0.0047	0.1044	0.1294
Worst	0.0077	0.0067	0.1620	0.2001
Average	0.0051	0.0052	0.1417	0.1537
Median	0.0068	—	0.1263	0.1546
Std.	0.0014	—	0.0137	0.0163

表 4.6 VFMOAPO 及比较算法对测试函数 ZDT3 的 *SP* 结果

SP	MOAPO	MOPSO	NSGA	SPEA
Best	6.4312e-6	0.0341	0.0276	0.0166
Worst	0.0163	0.0305	0.0770	0.0295
Average	0.0017	0.0276	0.0425	0.0222
Median	0.0078	—	0.0379	0.0211
Std.	0.0051	—	0.0112	0.0033

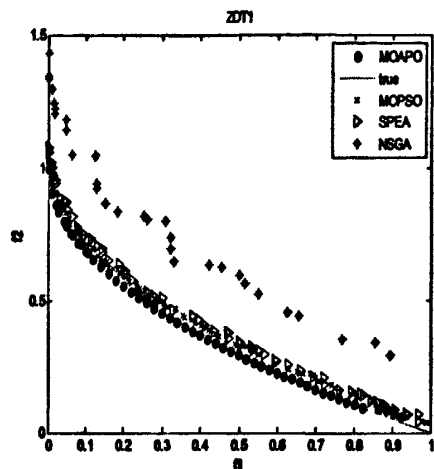


图 4.1 VFMOAPO 与比较算法产生的 ZDT1 的 Pareto 前沿

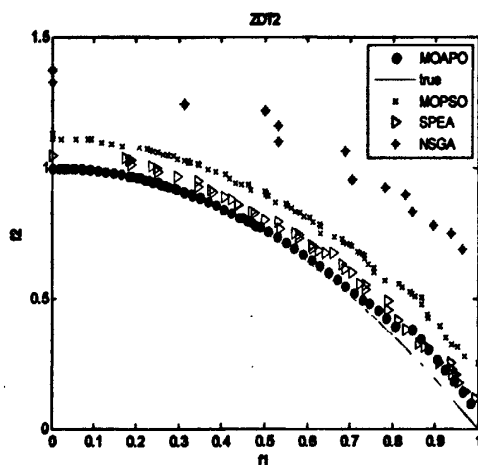


图 4.2 VFMOAPO 与比较算法产生的 ZDT2 的 Pareto 前沿

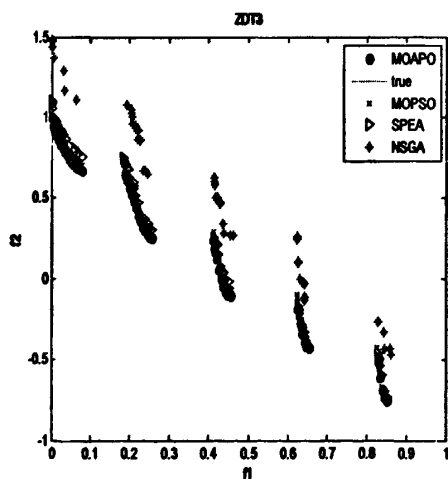


图 4.3 VFMOAPO 与比较算法产生的 ZDT3 的 Pareto 前沿

4.5 本章小结

为了解决基于聚集函数法的 MOAPO 算法没有最优个体保留机制的不足,本章在 MOAPO 算法中引入了 *archive* 集合来保留进化过程中产生的非劣解,提出了一种基于虚拟力排序的 MOAPO 算法。该算法通过 APO 自身进化机制的特征来保持 Pareto 解集良好的分布性,从而避免引入其他算法带来的额外的计算量。然后对所提出的 VFMOAPO 算法的收敛性进行了理论分析,利用概率论中的基本知识对其进行了证明。最后利用 VFMOAPO 算法对多目标优化问题的测试函数进行了数值实验,并与经典的 MOEAs 与 MOPSO 算法进行了比较,说明了本章所提算法的有效性,特别在 Pareto 解集分布性保持方面,具有一定的优势。

第5章 无约束多目标拟态物理学优化算法适应值函数

研究

5.1 引言

个体适应值分配是多目标进化算法中的关键技术之一,对算法的性能有着重要影响。多目标优化问题与单目标优化问题有着本质区别,所以在单目标优化领域中常用的适应值分配策略不再适用于多目标优化领域,不能简单的将其引入到多目标优化算法中,而应该根据多目标优化问题的特点,对搜索空间中的个体样本进行适应值分配。

本文中,第三章提出了 MOAPO 算法(我们称之为标准 MOAPO 算法,简称为 SMOAPO 算法),第四章提出了 VFMOAPO 算法,对很多多目标优化问题的测试函数,这两个算法表现出了较好的性能,特别在解集分布性保持方面,要优于经典的 MOEAs 和 MOPSO 算法。用 APO 解决多目标优化问题,在计算质量时需要群体中最好个体及最差个体的函数适应值,这时需要通过一定策略将多个子目标函数组合成一个单目标函数。SMOAPO 算法及 VFMOAPO 算法均采用基于聚集函数的策略来实现多个子目标到单个总目标的转化。通过这种方式得到的个体质量在一定程度上反映出了个体的优劣,但没有充分体现出多目标优化问题特点。本章提出一种基于序值的 MOAPO 算法(RMOAPO),该算法通过 Pareto 支配关系和拥挤程度为个体确定序号,用定义序号的方法替代 SMOAPO 算法及 VFMOAPO 算法中计算质量函数使用总目标函数值的方法,以此来解决上述问题。

本章的组织结构如下:

首先基于 Pareto 支配关系定义了个体序值的概念,给出了个体邻域内拥挤程度的定义,在此基础上提出了一种适应值函数的计算方法。然后根据这种适应值计算方法提出了一种基于序值的多目标拟态物理学优化算法,对算法思想及算法步骤进行了描述。最后通过数值实验及与经典 MOEAs、MOPSO 算法及 SMOAPO 算法的比较,充分说明了 RMOAPO 算法的有效性。

5.2 基于序值的质量函数的计算

5.2.1 个体序值的定义

一个进化种群是由若干个个体组成,个体间存在这样或那样的关系,比如支配与被支配关系,同一小生境内的聚集关系等,种群内部的各种性质就是由组成该种群的个体间的相互关系决定的^[9]。所以我们可以利用种群中个体间的相互关系将个体进行排序,以此来反映个体的优劣。序值是一个在多目标优化领域常用的概念,许多经典的 MOEAs 都采用序值的概念来定义算法的适应值函数。这里我们仍采用 Pareto 支配概念来定义个体间非支配或被支配关系,具体定义在前面 1.1.2 节中定义 1.3 已经给出,在此不再赘述。

定义 5.1: (个体序值的定义) 设在第 t 代生成的种群 $P(t)$ 由 N 个个体组成, $n_i(t)$ 表示 $P(t)$ 中支配个体 i 的所有个体的数量,则个体 i 在第 t 代的序值可以定义为 $r_i(t) = 1 + n_i(t)$ 。

5.2.2 邻域拥挤度

根据上述个体序值的定义,为种群中的个体分配序值,这样在第 t 代中可能会存在大量序值相同的个体,对这些个体我们采用邻域拥挤程度进行进一步处理。所谓邻域是指在给定个体的半径 ε 内的区域称为该个体的 ε 邻域^[9]。由于多目标优化算法设计的目标就是使进化过程朝 Pareto 最优解集搜索和保持非劣解集的多样性,所以根据上述序值的概念和个体所处邻域内的个体的拥挤程度,将个体从 1 到 N (N 为群体规模)进行编号。

RMOAPO 算法中,邻域半径 r 对算法起着重要作用。对于邻域半径的选择,通常由决策者根据 Pareto 最优解集中个体间的最小期望间距来确定。然而在实际应用中,人们对 Pareto 最优解集往往无法预知,并且如果要根据邻域内所含个体的数量而确定个体的优良程度,邻域半径太小,则容易存在大量邻域内个体数量为 0 的情况,这样对于个体优劣的评价较为困难,故本章中邻域半径 r 由等式 (5.1) 确定。

$$r = \frac{(\sum_{i=1}^n (\max |E_i|)^2)^{\frac{1}{2}}}{\text{archivesize} - 1} \quad (5.1)$$

其中 $\max |E_i|$ 为初始化时第 i 维上个体间最大的欧氏距离, archivesize 为非支配解集的大小。

如果具有相同序值的个体的拥挤程度相同,则要对邻域半径进行调整,即对邻域进行缩小或扩大:如果具有相同序值的个体邻域内的个体数量(包括该个体)小于等

于 2 时, 如果再对邻域缩小则意义不是很大, 故而这种情况下对邻域进行扩张, 将邻域半径增加到原半径的 1.5 倍, 然后再考查调整后的邻域内的个体数量; 如果具有相同序值的个体邻域内的个体数量大于 2 时, 则对邻域进行缩小, 将半径缩小至原来半径的一半, 然后再考查调整后的邻域内的个体数量。

5.2.3 质量函数的计算

MOAPO 算法中, 质量函数对算法的性能起着非常重要的作用。质量函数反映了个体适应值的优劣, 为优化搜索提供重要信息。个体适应值的大小就是通过个体质量来表现的: 较好的个体适应值对应着较大的个体质量; 较差的个体适应值对应于较小的个体质量。所以个体质量是个体适应值的函数, 是个体适应值在整个种群中优劣的评价。

RMOAPO 算法中, 根据个体的序值及邻域拥挤程度来评价个体的优劣: 个体的序值越小, 说明支配该个体的个体数量越少, 这样的个体相对而言较为优秀; 反之, 个体的序值越大, 说明支配该个体的个体数量较多, 这样的个体相对较差。对于具有相同序值的个体而言, 如果该个体邻域拥挤度较大, 说明该个体所处区域个体分布较为稠密, 个体密度较大; 如果该个体邻域拥挤度较小, 说明该个体所处区域的个体密度较小。对于基于种群的多目标优化问题, 为了保持种群的多样性, 个体密度越小, 个体相对较优, 应具有较好的适应值; 反之, 个体密度越大, 个体相对较差, 应具有较差的适应值。这样可以根据个体的序值及个体邻域拥挤度为每一个个体分配一个序号。如果具有相同序值的个体, 其邻域拥挤度也相同, 则对邻域半径进行相应的调整。如果调整半径之后, 具有相同序值的个体的拥挤程度仍然相同, 则说明这些个体的优劣程度相同, 将欲分配给这些个体的编号进行随机分配即可。使用这种方法给群体中的每个个体赋予了不同的序号, 从而对质量函数定义如下:

$$m_i(t) = e^{\frac{-rank_i(t)}{N}} \quad (5.2)$$

与将多个被优化的子目标转化成单个优化总目标的 SMOAPO 算法相比, RMOAPO 算法的质量函数不再是通过权重简单的将各个子目标进行聚集, 而是充分考虑到个体间的支配关系以及个体所处区域的密度, 这样更能体现出多目标优化问题自身的特点。

5.3 基于序值的无约束 MOAPO 算法

5.3.1 算法思想

RMOAPO 算法的基本思想是为每个个体分配一个不同的序号,遵循的规则是根据 5.2.1 节定义的个体序值的概念,按照其升序排列,如果存在多个个体具有相同序值的情况,就参照邻域内拥挤度来排列个体:拥挤程度较小的个体的序号较小,拥挤程度较大的个体序号较大,如果具有相同序值的个体的拥挤程度相同,则再调整邻域半径,根据调整半径后的邻域内个体的拥挤程度来排列个体。如果调整半径之后邻域内个体的拥挤程度仍然相同,则将欲分配给这些个体的序号进行随机分配即可。例如,在第 t 代生成的种群 $P(t)$ 中存在三个非支配解 A、B、C,根据序值的概念,这三个个体具有相同的序值,现在欲将自然数 1 到 3 映射为这三个个体的序号,映射的方法为:首先考查每个个体邻域内的个体数量,如果每个邻域内的个体数量不等,假设 A 邻域内共有 5 个个体(包括 A),B 邻域内有 7 个个体,C 邻域内有 3 个个体,则 A 的序号 $rank_A(t)=2$,B 的序号 $rank_B(t)=3$,C 的序号 $rank_C(t)=1$;如果有两个邻域内个体数量相等,假设 A 邻域内和 B 邻域内各自都包含 5 个个体,C 邻域内有 3 个个体,对原来邻域半径进行调整:将邻域半径缩小至原来一半,再考查 A 邻域和 B 邻域内的个体数量,例如 A 邻域内有 4 个个体,B 邻域内有 3 个个体,则 C 的序号 $rank_C(t)=1$,B 的序号 $rank_B(t)=2$,则 A 的序号 $rank_A(t)=3$;若邻域半径缩小一半后,A、B 邻域内的个体数量仍然相等,那么 A 和 B 的序号为 2 还是 3 则进行随机分配。对于被支配个体也采用相同的规则来定义个体的序号,如果第 t 代生成的种群 $P(t)$ 中有 n_1 个非支配解,则余下的个体的序号从 n_1+1 开始,可以暂时将 n_1 个非支配解从群体中移去,按照上述处理非支配解的方法处理剩余个体中的非支配解,然后将它们也暂时移去,如此循环,直至所有个体都被编号为止。

5.3.2 算法流程

- Step1** 初始化群体,给定群体规模 N ,随机产生每个个体的初始位置和初始速度,根据等式 (5.1),计算邻域半径 r 。
- Step2** 计算每个个体 $i(i=1,...,N)$ 在每个目标下的函数值,保留支配个体 i 的所有个体的数量 n_i ,找出非支配个体,将其存储在 *archive* 集中,并给每个非支配个体分配序值为 1,为其余个体分配序值 n_i+1 。
- Step3** 根据序值将群体中所有个体按升序排列。
- Step4** 计算具有相同序值的个体 r 邻域内的个体数量,按照该数量的升序将同序值个

体排序,对具有相同数量的同序值个体 i 进行标记 $flag(i)=1$ 。

Step5 如果 $flag(i)=1$,则根据上面 5.3.1 节中的方法调整邻域半径。如果调整后具有相同序值的个体的 r 邻域内的个体数量仍相同,对这些个体的顺序随机排列。

Step6 根据上述排序结果将每个个体赋一个序号,即为每个个体分配一个 1 到 N 间的自然数作为 $rank_i(i)$ 。

Step7 根据公式 (5.2) 计算个体质量,根据公式 (2.7) 和公式 (2.8) 计算个体受到其他个体在第 k 维上的作用力及在该维上所受到的合力,根据公式 (2.9) 和公式 (2.10),更新个体速度和位置。

Step8 计算每个个体在每个目标下的函数值,保留支配每个个体的个体数量 n_i ($i=1,\dots,N$),找出非支配个体,并给每个非支配个体分配序值为 1,为其余个体分配序值 n_i+1 。将每一个非支配个体与存储在 *archive* 集中的个体比较,若该个体既不被 *archive* 集中个体支配,也不支配 *archive* 集中的个体,则将该个体存入 *archive* 集中;若该个体支配 *archive* 集中的个体,则将被支配个体从 *archive* 集中删除,将非支配个体放入 *archive* 集中。若需要存储的个体超过 *archive* 集的规模,则将支配其他个体数量最少的个体删除。

Step9 若达到最大迭代次数则退出,否则返回 Step3。

5.4 数值实验及结果分析

我们利用 RMOAPO 算法对一组具有代表性的多目标测试函数进行了实验,将其与典型的多目标进化算法在评价指标 GD 和 SP 上进行比较,分析算法的各项性能。测试函数选择了 ZDT1、ZDT2、ZDT3、ZDT4、ZDT6,这五个测试函数的决策变量的维数均为 30 维,ZDT1 函数主要测试多目标优化算法解决 Pareto 前沿是凸的优化问题的能力,ZDT2 函数主要测试多目标优化算法解决 Pareto 前沿是凹的优化问题的能力,ZDT3 函数主要测试多目标优化算法解决 Pareto 前沿是不连续的优化问题的能力,ZDT4 函数是 ZDT 函数中最复杂的,有 21^9 个不同的局部 Pareto 前沿^[38],该函数主要测试多目标优化算法处理多峰函数和解决局部 Pareto 前沿众多的优化问题的能力,ZDT6 函数主要测试多目标优化算法解决 Pareto 前沿是不均匀的优化问题的能力。以上 5 个测试函数定义如表 5.1 所示。

实验中,RMOAPO 算法的群体规模及 *archive* 集均为 100,迭代次数为 100 次,引力常数 G 为 10,惯性权重 ω 从 0.9 到 0.4 随进化代数线性减少。用来进行比较的算法选择了文献[90]的 MOPSO 算法,比较数据由该文献作者提供。该算法中,群体规模及 *archive* 集大小均为 100,进化代数为 5000 代,学习因子 2.05,惯性权重从 0.9 到 0.4 随进化代数线性减少。我们还选择了经典多目标进化算法 NSGA 和 SPEA 算法

作为比较算法, 这两个算法的数据来源于文献[102]。我们将 RMOAPO 算法与 SMOAPO 算法也进行了比较, SMOAPO 算法中群体规模及 *archive* 集均为 100, 迭代次数为 100 次, 引力常数 G 为 10, 惯性权重 ω 从 0.9 到 0.4 随进化代数线性减少。

表 5.1 RMOAPO 算法测试函数

Problem	Variable bounds	Objective functions
ZDT1	$x = (x_1, x_2, \dots, x_m)$ $x_i \in [0, 1]$ $i = 1, 2, \dots, m$ $m = 30$	$f_1(x_1) = x_1$ $f_2(x) = g(1 - \sqrt{f_1/g})$ $g(x_2, \dots, x_m) = 1 + 9 \sum_{i=2}^m x_i / (m-1)$
ZDT2	$x = (x_1, x_2, \dots, x_m)$ $x_i \in [0, 1]$ $i = 1, 2, \dots, m$ $m = 30$	$f_1(x_1) = x_1$ $f_2(x) = g(1 - (f_1/g)^2)$ $g(x_2, \dots, x_m) = 1 + 9 \sum_{i=2}^m x_i / (m-1)$
ZDT3	$x = (x_1, x_2, \dots, x_m)$ $x_i \in [0, 1]$ $i = 1, 2, \dots, m$ $m = 30$	$f_1(x_1) = x_1$ $f_2(x) = g(1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1))$ $g(x_2, \dots, x_m) = 1 + 9 \sum_{i=2}^m x_i / (m-1)$
ZDT4	$x = (x_1, x_2, \dots, x_m)$ $x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2, \dots, m$ $m = 30$	$f_1(x_1) = x_1$ $f_2(x) = g(1 - \sqrt{f_1/g})$ $g(x) = 1 + 10(m-1) + \sum_{i=2}^m (x_i^2 - 10 \cos(4\pi x_i))$
ZDT6	$x = (x_1, x_2, \dots, x_m)$ $x_i \in [0, 1]$ $i = 1, 2, \dots, m$ $m = 30$	$f_1(x_1) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(x) = g(1 - (f_1/g)^2)$ $g(x_2, \dots, x_m) = 1 + 9(\sum_{i=2}^m x_i / (m-1))^{0.25}$

利用 RMOAPO 算法和其他经典多目标进化算法及 SMOAPO 算法对上述 5 个测试函数进行测试, RMOAPO 算法与 SMOAPO 算法分别独立运行 30 次, 将第 15 次

运行结果以图形形式输出, 同样, 分别将资料[102]中 30 个 NSGA 和 SPEA 结果中的第 15 个以图形形式输出, 由于文献[90]作者只提供了 MOPSO 算法的一次的运行结果, 故将其直接输出, 得到的 Pareto 前沿如图 5.1 至图 5.5 所示, 得到性能评价指标 GD 和 SP 的测试结果如表 5.2 和表 5.3 所示。从图 5.1 我们可以看到, 利用 RMOAPO 算法得到测试函数 ZDT1 的 Pareto 前沿基本都在真实解上, 表 5.2 和表 5.3 中的数据显示, RMOAPO 算法对 Pareto 真实解集的逼近程度要好于 SMOAPO 算法, SPEA 和 NSGA, 比 MOPSO 算法得到的解集要差, 而 RMOAPO 算法得到的非劣最优解集的分布性要好于其他四种比较算法。图 5.2 中利用 RMOAPO 算法得到测试函数 ZDT2 的 Pareto 前沿基本落在真实解上, 数据表明, 其对 Pareto 真实解集的逼近程度要好于 MOPSO, SPEA 和 SMOAPO 算法, 远远好于 NSGA 算法。在非劣最优解集的分布性保持方面, RMOAPO 算法要稍好于 SMOAPO 算法, 好于 SPEA 和 MOPSO 算法, 远远好于 NSGA 算法。RMOAPO 算法得到测试函数 ZDT3 的 Pareto 前沿的个别点不在真实解上, 对 Pareto 真实解集的逼近程度比 MOPSO 略差, 要好于其他比较算法, 对 Pareto 解集分布性而言, RMOAPO 算法与 SMOAPO 算法、SPEA 和 MOPSO 算法相差不多, 均要好于 NSGA 算法得到的 Pareto 前沿。RMOAPO 算法得到测试函数 ZDT4 的 Pareto 前沿基本都在真实 Pareto 前沿上, 但其非支配解的数量与分布性要劣于通过 MOPSO 算法得到的解, 远远好于 NSGA 和 SPEA 得到的解, 而 SMOAPO 算法得到的解虽然要好于 NSGA 和 SPEA 得到的解, 但其非支配解的数量和分布性方面均比 RMOAPO 算法差。由于测试函数 ZDT6 的 Pareto 前沿是非均匀的, 而 RMOAPO 及其他比较算法在设计过程中都采用一定的策略保持所得到的解在 Pareto 前沿上均匀分布, 故而通过这些算法得到的 ZDT6 的 Pareto 前沿与真实 Pareto 前沿相差较远, 分布性也不好。

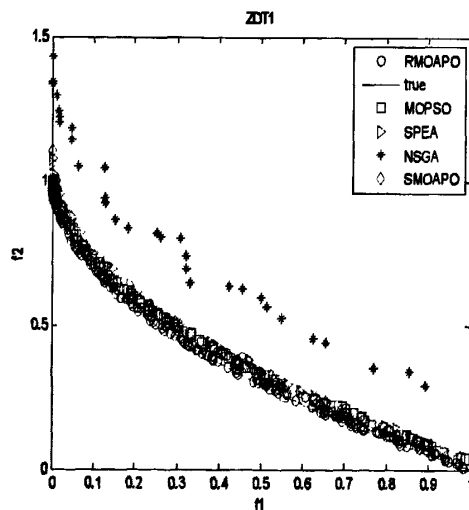


图 5.1 各算法对测试函数 ZDT1 的 Pareto 前沿

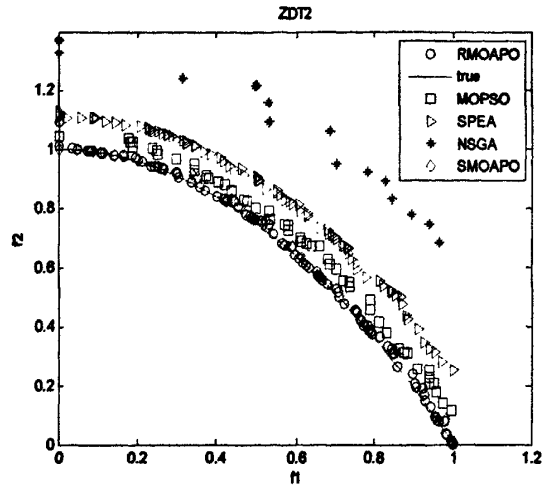


图 5.2 各算法对测试函数 ZDT2 的 Pareto 前沿

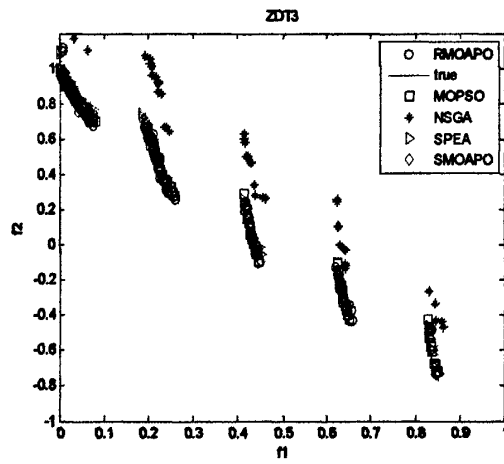


图 5.3 各算法对测试函数 ZDT3 的 Pareto 前沿

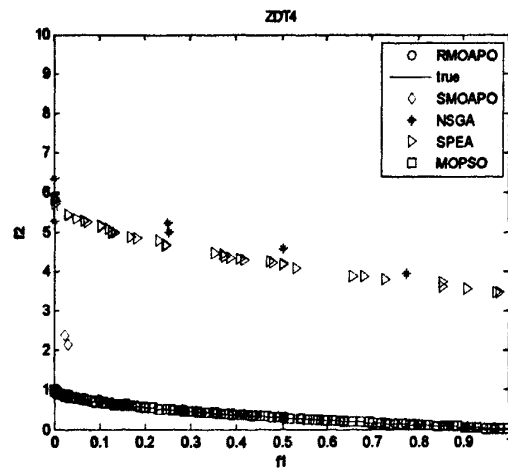


图 5.4 各算法对测试函数 ZDT4 的 Pareto 前沿

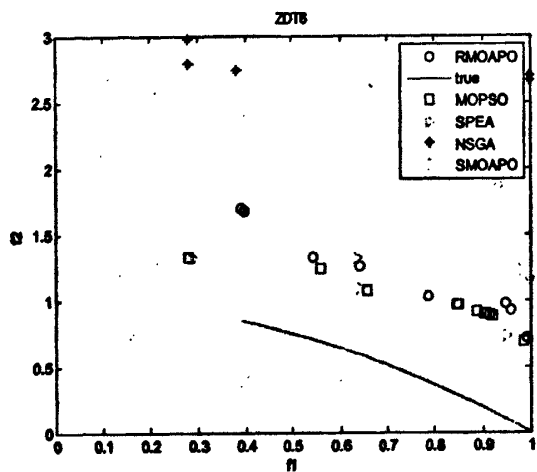


图 5.5 各算法对测试函数 ZDT6 的 Pareto 前沿

从实验结果我们可以看出，与经典多目标进化算法、MOPSO 算法及 SMOAPO 算法相比，RMOAPO 对典型的多目标优化算法的测试函数得到了较好的测试结果，并且该算法的进化代数要远远少于经典多目标进化算法及 MOPSO 算法，从而减小了算法计算量及时间花销。故 RMOAPO 算法在处理 Pareto front 为凸、非凸、不连续、多峰、局部前沿众多的多目标优化问题时，均能获得较好的效果。

表 5.2 RMOAPO 测试函数的 GD 值比较

Functions	GD	RMOAPO	SMOAPO	NSGA	SPEA	MOPSO
ZDT1	mean	0.0360	0.0524	0.1715	0.1034	0.0022
	best	0.0270	0.0365	0.1453	0.0893	0.0016
	worst	0.0463	0.0628	0.2140	0.1181	0.0048
	median	0.0433	0.0487	0.2140	0.1064	—
	std	0.0051	0.0081	0.0155	0.0066	—
ZDT2	mean	0.0176	0.0459	0.2668	0.1088	0.0509
	best	0.0083	0.0105	0.2159	0.0889	0.0276
	worst	0.0353	0.0562	0.3100	0.1340	0.0667
	median	0.0353	0.0105	0.3019	0.0961	—
	std	0.0058	0.0104	0.0232	0.0124	—
ZDT3	mean	0.0429	0.0833	0.1417	0.1537	0.0052
	best	0.0324	0.0451	0.1044	0.1294	0.0047
	worst	0.0815	0.1008	0.1620	0.2001	0.0060
	median	0.0348	0.0451	0.1263	0.1546	—
	std	0.0094	0.0151	0.0137	0.0163	—

ZDT4	mean	0.0730	0.0955	2.3950	0.7068	0.014
	best	0.0180	0.0667	0.8817	0.1856	0
	worst	0.1655	0.1305	5.1436	1.4667	0.0408
	median	0.0629	0.1305	2.4916	0.8156	—
	std	0.0241	0.0350	0.8727	0.3744	—
ZDT6	mean	0.2131	0.4256	1.0846	0.3939	0.5886
	best	0.1322	0.2311	0.7630	0.2323	0.2222
	worst	0.3033	0.6546	1.9086	0.6023	0.8558
	median	0.2059	0.6546	1.0412	0.5040	—
	std	0.0341	0.2107	0.2437	0.1028	—

表 5.3 RMOAPO 测试函数的 SP 值比较

Functions	SP	RMOAPO	SMOAPO	NSGA	SPEA	MOPSO
ZDT1	mean	0.0060	0.0102	0.0409	0.0218	0.0120
	best	0.0041	0.0063	0.0300	0.0155	0.0084
	worst	0.0118	0.0239	0.0603	0.0295	0.0203
	median	0.0047	0.0069	0.0512	0.0227	—
	std	0.0016	0.0051	0.0071	0.0031	—
ZDT2	mean	0.0049	0.0069	0.0725	0.0374	0.0151
	best	0.0035	0.0038	0.0337	0.0217	0.0117
	worst	0.0111	0.0113	0.1525	0.0573	0.0218
	median	0.0053	0.0083	0.0830	0.0349	—
	std	0.0015	0.0024	0.0269	0.0085	—
ZDT3	mean	0.0235	0.0233	0.0425	0.0222	0.0305
	best	1.005e-4	0.0057	0.0276	0.0166	0.0276
	worst	0.1611	0.0971	0.0770	0.0295	0.0341
	median	5.4351e-4	0.0285	0.0379	0.0211	—
	std	0.0448	0.0266	0.0112	0.0033	—
ZDT4	mean	0.1817	0.2472	0.2820	0.7068	0.0114
	best	0.0173	0.0294	0.0950	0.1856	0.0055
	worst	1.0517	1.3377	0.5634	1.4667	0.0410
	median	0.2103	0.0239	0.1651	0.8156	—
	std	0.2949	0.3857	0.1176	0.3744	—

ZDT6	mean	0.4823	0.7106	0.4156	0.3066	0.4571
	best	0.3632	0.3013	0.3001	0.0239	0.2848
	worst	0.8637	0.8426	0.7441	0.6417	0.5994
	median	0.8637	0.3924	0.4041	0.4898	——
	std	0.0701	0.0754	0.0935	0.0712	——

5.5 本章小结

本章通过分析 MOAPO 算法中个体质量与个体适应值的关系，提出了一种 RMOAPO 算法，用于解决多目标优化问题。该算法根据 Pareto 支配概念给个体分配序值，用于指示个体的优劣程度。对于具有相同序值的个体，采用邻域内的拥挤程度作为个体好坏的评价指标；对于具有相同序值和相同拥挤程度的个体，通过调整个体邻域半径来评价个体的优劣程度。根据以上指标，给每个个体分配不同的序号，完成对搜索空间中进化个体的评价。通过对 ZDT 系列函数的测试结果表明，与经典的多目标进化算法及 SMOAPO 算法相比，本章提出的 RMOAPO 算法在对真实 Pareto 前沿的逼近程度及解集分布性方面均有较好的性能，并且该算法的进化代数较小，从而有效提高了算法效率。

第 6 章 约束多目标拟态物理学优化算法研究

6.1 引言

目前多目标进化算法中主要研究无约束问题,然而现实世界中的多目标优化问题往往都带有约束条件。对于约束多目标优化问题,由于约束条件的存在,导致搜索空间分解成可行和不可行两个区域,特别是对于有些约束多目标优化问题,其最优解就位于可行域与不可行域的边界上或在其附近,使得多目标优化算法在收敛到真正的 Pareto 最优区域并且保持解的多样性存在巨大困难,其求解难度要大于单目标约束优化和无约束多目标优化。

一般地,以最小化问题为例,约束多目标优化问题 (Constrained Multi-objective Optimization Problem, CMOP) 可以定义如下:

$$\begin{aligned}
 \min y &= f(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\
 \text{s.t.} \quad &g_i(x) \geq 0, (i=1, 2, \dots, k) \\
 &h_j(x) = 0, (j=1, 2, \dots, l) \\
 \text{where } &x = (x_1, x_2, \dots, x_n) \in X, \\
 &X = \{(x_1, x_2, \dots, x_n) | l_i \leq x_i \leq u_i, i=1, 2, \dots, n\}, \\
 &L = (l_1, l_2, \dots, l_n), \\
 &U = (u_1, u_2, \dots, u_n), \\
 &y = (y_1, y_2, \dots, y_m) \in Y
 \end{aligned} \tag{6.1}$$

其中, x 为决策变量, y 为目标函数, X 表示决策空间, L 和 U 分别为其下界和上界, Y 表示目标空间。

约束多目标优化问题的可行域及约束非劣解集定义如下:

定义 6.1 可行域是一个满足约束条件的决策变量的集合, 定义如下:

$$X_f = \{x \in X | g_i(x) \leq 0 \cap h_j(x) = 0, i=1, 2, \dots, k, j=1, 2, \dots, l\}$$

定义 6.2 在约束多目标优化问题中, 约束 Pareto 最优解集 (或称约束非劣解集) 定义如下:

$$P_\sigma = \{x^* | \neg x \in X_f : x \succ x^*\}$$

P_σ 对应于目标空间的集合称为约束 Pareto 前沿, 定义如下:

$$P_f = \{f(x) = (f_1(x), f_2(x), \dots, f_m(x))^T | x \in P_\sigma\}$$

目前在约束多目标优化领域常用的约束处理方法主要借鉴单目标约束优化领域的成果, 具有代表性的方法有以下几种:

1. 约束保持法

这种方法是最容易理解的一种约束处理方法,它的主要思想是通过确保进化过程中所有个体都在可行域范围之内,从而缩小搜索范围,加快搜索速度。但这种方法要求初始化时个体必须满足所有的约束条件,因此对于有些问题需要花费很长时间来产生可行初始解,有些甚至产生不了可行解。

2. 罚函数法

这种方法是传统优化方法以及智能优化算法求解约束优化问题时处理约束冲突最常用的方法。它的基本思想是根据约束的特点,构造某种惩罚函数,追加到目标函数中去,对优化目标按约束违反程度进行惩罚,从而把约束问题转换成无约束问题,然后利用无约束优化方法来求解该约束问题。这类方法简单易行,不需要约束的梯度信息,也不需要考虑产生的个体是否超出可行域,因此很适合于一些工程应用。但其主要缺点是难以选择合适的惩罚因子,以确定对不可行解进行合适的惩罚。若惩罚因子太小,则惩罚项在目标函数中所占比重比较小,从而很难产生可行解;若惩罚因子太大,则容易使算法较早地收敛到某个局部最优解。

3. 可行规则法。这种方法主要根据可行规则来寻找个体的最优位置,它不需要保证任意时刻个体必须在可行域内,也不需要设计额外的参数,该方法在处理约束时只需要计算约束违反程度。但由于可行个体始终优于不可行个体,所以较容易陷入早熟收敛。

目前对约束多目标优化算法研究工作主要集中在以下几个方面: Deb 等人通过对单目标约束优化的比较准则进行扩展,提出用可行规则法来处理多目标问题中的约束,将其应用于经典算法 NSGA-II 中^[38],成为当前多目标进化算法中的经典算法,同时 Deb 等人还设计了约束多目标优化的测试函数集; Sarker 等人提出了一种对 Pareto 前沿差分进化算法 (PDE) 改进的约束多目标进化算法^[108]; Jiménez 及 Deb 等人提出的将约束处理及多样性处理结合的约束多目标进化算法^[109]; Aguirre 及 Coello 等人提出了逆向可收缩 PAES 算法来解决约束多目标优化问题等^[110]。邹秀芬、康立山等人提出了将传统意义下的 Pareto 占优关系转化为约束 Pareto 占优,然后在约束占优的概念下求解多目标优化问题^[105]; 王跃宣等人通过计算不可行度及不可行度阈值来进行选择操作,处理带约束的多目标优化问题^[111]; 孟红云、刘三阳等人用构造双群体分别保持搜索过程中找到的可行解与部分有优良特性的不可行解的方法来求解约束多目标优化问题^[112]; 尚荣华、焦李成等人将约束条件作为一个优化目标,然后用免疫记忆克隆算法进行求解^[113]。

近年来研究者们提出了许多新的基于群体智能及模拟物理化学原理的多目标优化算法。但这些算法主要集中于无约束多目标优化领域,且大多数缺乏必要的理论分析与研究。同时,现有的约束多目标优化理论及算法,大部分是采用有限齐次 Markov

链作为分析工具,对约束多目标优化问题的收敛性分析进行研究^[105,106]。但是许多基于群体智能和模拟物理化学的多目标优化算法,并不满足 Markov 性,无法用有限齐次 Markov 链作为算法收敛性的分析工具。

针对目前约束多目标优化算法中存在的不足,本章将拟态物理学方法应用于约束多目标优化领域中,首先结合约束多目标优化问题的特点,利用可行规则法作为约束处理技术,提出了一种基于虚拟力减小的约束多目标拟态物理学优化(VDCMOAPO)算法,然后结合该算法特点,利用概率论中的基本知识,对该算法的收敛性进行了理论分析及证明,并通过仿真实验说明了所提出算法的有效性。最后采用约束保持法作为约束处理机制,结合 RMOAPO 算法思想,提出了一种基于序值的约束多目标拟态物理学优化(CRMOAPO)算法,并通过仿真实验测试了算法性能。

6.2 基于可行规则法的约束多目标拟态物理学优化算法研究

约束多目标优化是多目标优化问题的一个重要研究方向。约束多目标优化的约束处理技术、算法对真正约束 Pareto 最优解的逼近程度及解集分布性保持是其研究的核心内容。多目标优化问题的搜索空间被约束条件分为可行与不可行两个区域,求解约束多目标优化问题的本质就是搜索可行优化解,使其收敛到真正的约束 Pareto 最优区域并且保持解集的多样性。

可行规则法作为一种约束处理技术,不需要任何时刻都要保证个体必须在可行域内,也不需要设计额外的参数,是约束多目标优化领域处理约束的一种较为常用的方法^[114]。本节利用可行规则法,结合约束多目标拟态物理学优化(CMOAPO)算法自身特点,通过调整个体所受虚拟力的大小,使个体朝 Pareto 最优区域运动,从而得到约束 Pareto 非劣解集。

6.2.1 虚拟力减小的约束多目标拟态物理学优化算法

可行规则法在处理约束时需要计算个体的约束违反程度,其定义如下:

定义 6.3 对于约束多目标优化问题(6.1),每个解的约束冲突程度(或称为不可行度)定义如下:

$$\phi(x_i) = \sum_{j=1}^k [\min\{0, g_j(x_i)\}]^2 + \sum_{p=1}^l [h_p(x_i)]^2 \quad (6.2)$$

不可行度也可以认为是解 x_i 到可行域的距离, x_i 离可行域越远,不可行度越大,反之 x_i 离可行域越近,不可行度越小,当 x_i 是可行解时,不可行度为零。

与解决无约束多目标优化问题相似,约束多目标优化算法的目标是:找到可行的非支配解集,使其充分接近且均匀分布于约束 Pareto 前沿。根据约束多目标优化中的

约束支配关系可知,对于可行域内的个体,其优化目标是朝 Pareto 前沿运动,而对于不可行域内的个体,其优化目标是朝可行域运动。所以在算法的设计过程中应该采用不同的方法处理不同区域内的个体,使其能够朝可行优化的方向运动。在本算法中,我们采用随机加权和方法将多个子目标函数转换成单个总目标函数。

如果直接将 APO 应用于约束多目标优化问题中,那么会出现这样的情况:在 t 时刻,个体 i 位于可行域内,而在其他个体虚拟力作用下,在 $t+1$ 时刻,个体 i 运动至不可行域内,甚至远离可行域,特别是约束 Pareto 前沿处于可行域和不可行域的边界时,这种情况是很容易出现的。所以在 VDCMOAPO 算法中,我们提出一种虚拟力减小的方法来尽量避免或减少上述情况的出现。即如果个体 i 在其他个体的合力作用下由可行域运动至不可行域,那么将该个体受到的虚拟力减小,然后再重新计算该个体的速度及位置,作为这次迭代的结果。这样就会尽可能地避免上述情况的发生,即使个体由可行域进入到不可行域,那么该个体位置较之虚拟力减小规则引入前得到的位置更接近于可行域。

VDCMOAPO 算法框架包括四部分:初始化,质量的计算,虚拟力的计算,位置和速度的更新。其中,虚拟力减小规则体现于更新部分。VDCMOAPO 算法框架如图 6.1 所示。

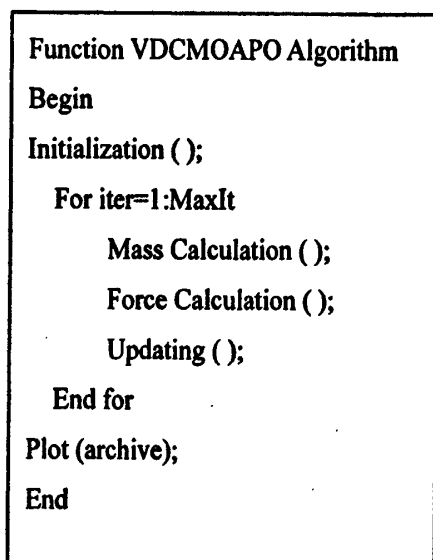


图 6.1 VDCMOAPO 算法框架

1. 初始化

设 P 为进化群体, $archive$ 为约束非劣解集, $feasible$ 为可行解集, $infeasible$ 为不可行解集。初始时随机产生 n 个个体组成 P , $archive$ 、 $feasible$ 和 $infeasible$ 均为空,大小均为 n 。首先根据约束条件将 P 中的可行个体放入 $feasible$ 集合中,不可行个体

放入到 *infeasible* 集合中, 计算各个目标下每个个体的目标函数的适应值, 挑选出 *feasible* 中的非支配个体放入到 *archive* 集中, 由于初始状态下个体所受到其他个体的作用力都为 0, 所以从 *archive* 集中随机选择一个个体作为全局最优个体。根据定义 6.3 计算 *infeasible* 中个体的不可行度。根据不可行度对 *infeasible* 集合按照降序进行快速排序, 如果两个解的不可行度相同, 对其进行随机排列即可。选择 *infeasible* 中第一个个体作为全局最差个体。

2. 个体质量计算

APO 算法中, 质量是用户定义的有关优化问题的适应值的函数, 即 $m_i = g(f(X_i))$ 。故而, 质量是个体好坏的一个重要标志。好的个体具有较大质量, 差的个体具有较小质量。在约束多目标拟态物理学优化算法中, 质量同样是评价个体好坏的重要指标。在约束多目标优化问题中, 对于可行域内的个体, 用户根据优化问题所定义的适应值函数 $f(X_i)$ 可以反映出个体的优劣。但对于不可行域内的个体, 违反约束程度的大小 $\varphi(X_i)$ 成为评价个体好坏的重要标志。所以在约束多目标优化问题中, 质量函数要体现出以下特点:

(1) 可行域内个体的质量要大于不可行域内个体的质量;

(2) 对于可行域内的个体, 具有较好个体适应值的个体质量较大, 具有较差适应值的个体质量较小;

(3) 对于不可行域内个体, 具有违反约束程度较小的个体质量较大, 具有较大约束违反程度的个体质量较小。

若要同时满足上述特点, APO 算法中的质量函数就不太适应了, 因此需要对其进行修改。APO 算法中, 质量函数满足条件: $m_i \in (0, 1]$ 。在约束多目标拟态物理学优化算法中, 为了充分体现可行域个体与不可行域内个体的差别, 我们将可行域内个体质量限定在 $(1, 2]$ 范围内, 将不可行域内个体质量限定在 $(0, 1]$ 范围内, 所以在约束多目标拟态物理学优化算法中, 质量函数 $m_i \in (0, 2]$ 。根据以上所述, 我们将质量函数修改如下:

$$\begin{aligned} m_i &= 1 + e^{\frac{f(x_{best}) - f(x_i)}{f(x_{worst}) - f(x_{best})}} \quad (x_i \in \text{feasible}) \\ m_i &= e^{\frac{-\varphi(x_i)}{\varphi(x)}} \quad (x_i \in \text{infeasible}) \end{aligned} \quad (6.3)$$

3. 虚拟力的计算

对于约束多目标优化问题而言, 可行域内个体与不可行域内个体有着本质区别, 所以, 不能将 APO 算法中的虚拟力作用规则直接应用于约束多目标拟态物理学优化算法中, 根据约束多目标优化问题的特点, 我们将原始 APO 算法中虚拟力的作用规则, 修改为以下四条规则:

规则 1: 如果个体 i 和个体 j 都处于可行域内, 则个体 i 受到个体 j 在第 k 维上的

作用力为

$$F_{y,k} = \begin{cases} Gm_i m_j (x_{j,k} - x_{i,k}), & (f(x_i) > f(x_j)) \\ -Gm_i m_j (x_{j,k} - x_{i,k}), & (f(x_i) \leq f(x_j)) \end{cases} \quad \forall i \neq j \text{ and } i \neq \text{best} \quad (6.4)$$

规则 2: 如果个体 i 在可行域内, 个体 j 在不可行域内, 则个体 i 受到个体 j 在第 k 维上的作用力为

$$F_{y,k} = 0 \quad (6.5)$$

规则 3: 如果个体 i 在不可行域内, 个体 j 在可行域内, 则个体 i 受到个体 j 在第 k 维上的作用力为

$$F_{y,k} = Gm_i m_j (x_{j,k} - x_{i,k}) \quad (6.6)$$

规则 4: 如果个体 i 和个体 j 都处于不可行域内, 则个体 i 受到个体 j 在第 k 维上的作用力为

$$F_{y,k} = \begin{cases} Gm_i m_j (x_{j,k} - x_{i,k}), & \text{if } (\varphi(x_i) > \varphi(x_j)) \\ -Gm_i m_j (x_{j,k} - x_{i,k}), & \text{if } (\varphi(x_i) \leq \varphi(x_j)) \end{cases} \quad \forall i \neq j \quad (6.7)$$

根据公式 (2.8) 计算出个体 i 所受到的其他所有个体在第 k 维上的合力。

4. 基于虚拟力减小的位置与速度的更新

在 APO 算法中, 计算出个体在第 k 维上受到的合力之和, 根据公式 (2.9) 和 (2.10) 更新个体速度和位置。有时在个体更新了速度和位置之后会出现这种情况: 在更新之前, 个体位于可行域内, 但当更新之后, 个体则位于不可行域内, 即个体从可行域运动至不可行域。这种情况在 Pareto 最优解位于可行域和不可行域的交界处时更容易出现。这是由于个体在所有其他个体的虚拟力作用下, 朝 Pareto 最优解方向运动, 但由于作用力过大, 使得个体的运动步长较大, 超越了可行域边界。为了尽量避免这种情况出现, 我们将其本次迭代中该个体受到其他个体的虚拟力的合力减半, 再根据公式 (2.9) 和 (2.10) 重新计算该个体的速度和位置, 并作为此次迭代的结果。这样新得到的个体位置有两种情况: 一种是新位置在可行域中, 另一种是新位置虽然在不可行域中, 但较原有位置更接近可行域。

6.2.2 算法收敛性证明

设 $PF(t)$ 是 VDCMOAPO 算法产生的目标空间群体序列, 即第 t 次迭代后 *archive* 集中的元素序列在目标空间的映射。为了方便表述, 将目标空间网格化为若干个 m 维超长方体, 每一个超长方体内的 Pareto 最优解的个数不能超过 1, 可以通过调整超长方体的个数 s , 使得上述情况成立。超长方体 i 内所包含的约束多目标优化问题的约束 Pareto 前沿为 PF_i , 设 $N_\varepsilon(F_i)$ 为 PF_i 的 ε 邻域。

定义 6.4: 若 $\forall \varepsilon > 0$, $\forall t' \geq t$, $\lim_{t \rightarrow \infty} P\{PF(t') \cap N_\varepsilon(PF_i) \neq \emptyset\} = 1, 1 \leq i \leq s$, 则称

VDCMOAPO 产生的目标空间群体序列 $PF(t)$ 强收敛于 PF_i ($1 \leq i \leq s$), 简称 VDCMOAPO 强收敛于其约束 Pareto 前沿。

定义 6.5: 如果 $\forall \varepsilon > 0$, $\lim_{t \rightarrow \infty} P\{PF(t) \cap N_\varepsilon(PF_i) \neq \emptyset\} = 1, 1 \leq i \leq s$, 则称 VDCMOAPO 弱收敛于其约束 Pareto 前沿。

为了方便描述, 将迭代过程中产生的目标空间的非劣解与约束 Pareto 前沿间的几种关系记为:

$$A_t = P\{PF(t+1) \cap N_\varepsilon(PF_i) = \emptyset \mid PF(t) \cap N_\varepsilon(PF_i) \neq \emptyset\} \quad (6.8)$$

$$B_t = P\{PF(t+1) \cap N_\varepsilon(PF_i) = \emptyset \mid PF(t) \cap N_\varepsilon(PF_i) = \emptyset\} \quad (6.9)$$

$$B'_t = B_0 * B_1 * \dots * B_t \quad (6.10)$$

其中 A_t 表示在第 t 次迭代产生的 *archive* 集收敛的条件下第 $t+1$ 次迭代产生的 *archive* 集不收敛的情况; B_t 表示第 t 次迭代产生的 *archive* 集不收敛的条件下第 $t+1$ 次迭代产生的 *archive* 也不收敛的情况。

定理 6.1: $\forall \varepsilon > 0$, 若 $\lim_{t \rightarrow \infty} B'_t = 0$, 则 $\lim_{t \rightarrow \infty} P\{PF(t) \cap N_\varepsilon(PF_i) \neq \emptyset\} = 1$, 即 VDCMOAPO

弱收敛于其约束 Pareto 前沿; 若 $\sum_{t=1}^{\infty} B'_t < \infty$, 则 $P\{\bigcap_{t=1}^{\infty} PF(t) \cap N_\varepsilon(PF_i) = \emptyset\} = 0$, 即

VDCMOAPO 强收敛于其约束 Pareto 前沿。

证明:

$$\begin{aligned} & P\{PF(t+1) \cap N_\varepsilon(PF_i) = \emptyset\} \\ &= P\{PF(t+1) \cap N_\varepsilon(PF_i) = \emptyset \mid PF(t) \cap N_\varepsilon(PF_i) \neq \emptyset\} * P\{PF(t) \cap N_\varepsilon(PF_i) \neq \emptyset\} \\ &+ P\{PF(t+1) \cap N_\varepsilon(PF_i) = \emptyset \mid PF(t) \cap N_\varepsilon(PF_i) = \emptyset\} * P\{PF(t) \cap N_\varepsilon(PF_i) = \emptyset\} \\ &= A_t * P\{PF(t) \cap N_\varepsilon(PF_i) \neq \emptyset\} + B_t * P\{PF(t) \cap N_\varepsilon(PF_i) = \emptyset\} \end{aligned} \quad (6.11)$$

VDCMOAPO 采用 *archive* 集保留可行域内的非劣最优解, 故不可能存在第 t 代时算法收敛于其 Pareto 前沿, 而在第 $t+1$ 代时不收敛的情况, 所以显然 $A_t = 0$ 。

$$\begin{aligned} P\{PF(t+1) \cap N_\varepsilon(PF_i) = \emptyset\} &= B_t * P\{PF(t) \cap N_\varepsilon(PF_i) = \emptyset\} \\ &= B_t * B_{t-1} * P\{PF(t-1) \cap N_\varepsilon(PF_i) = \emptyset\} \\ &= B_t * B_{t-1} * \dots * B_1 * B_0 * P\{PF(0) \cap N_\varepsilon(PF_i) = \emptyset\} \\ &= B'_t * P\{PF(0) \cap N_\varepsilon(PF_i) = \emptyset\} \end{aligned} \quad (6.12)$$

$$\because \lim_{t \rightarrow \infty} B'_t = 0$$

$$\therefore \lim_{t \rightarrow \infty} P\{PF(t+1) \cap N_\varepsilon(PF_i) = \emptyset\}_n = 0$$

$$\text{即 } \lim_{t \rightarrow \infty} P\{PF(t+1) \cap N_\varepsilon(PF_i) \neq \emptyset\} = 1,$$

所以 $\forall \varepsilon > 0$, 若 $\lim_{i \rightarrow \infty} B_i' = 0$, 则 $\lim_{i \rightarrow \infty} P\{PF(t) \cap N_\varepsilon(PF_i) \neq \emptyset\} = 1$,

故 VDCMOAPO 弱收敛于其约束 Pareto 前沿。

$$\because P\{PF(t+1) \cap N_\varepsilon(PF_i) = \emptyset\} = B_i' * P\{PF(0) \cap N_\varepsilon(PF_i) = \emptyset\}$$

由于 VDCMOAPO 算法的群体在初始化时是均匀产生的, 并且产生的群体在目标空间通常很少会包含在其约束 Pareto 前沿上, 所以 $P\{PF(0) \cap N_\varepsilon(PF_i) = \emptyset\} = \text{常数}$ 。

$$\therefore \sum_{i=1}^{\infty} (B_i' * P\{PF(0) \cap N_\varepsilon(PF_i) = \emptyset\}) = \sum_{i=1}^{\infty} B_i' * P\{PF(0) \cap N_\varepsilon(PF_i) = \emptyset\}$$

且根据假设 $\sum_{i=1}^{\infty} B_i' < \infty$,

由于 $P\{PF(0) \cap N_\varepsilon(PF_i) = \emptyset\} \in [0, 1]$

$$\text{所以 } \sum_{i=1}^{\infty} (B_i' * P\{PF(0) \cap N_\varepsilon(PF_i) = \emptyset\}) = \sum_{i=1}^{\infty} B_i' * P\{PF(0) \cap N_\varepsilon(PF_i) = \emptyset\} < \infty$$

$$\text{即 } \sum_{i=1}^{\infty} P\{PF(t+1) \cap N_\varepsilon(PF_i) = \emptyset\} < \infty$$

根据引理 4.1 可知 $P\{\bigcap_{i=1}^{\infty} \bigcup_{t \geq i} PF(t) \cap N_\varepsilon(PF_i) = \emptyset\} = 0$, 即 VDCMOAPO 强收敛于其约束 Pareto 前沿。

证毕。

定理 6.2: 若 VDCMOAPO 中 $\lambda \sim N(0, 1)$, 约束多目标优化问题的决策空间 Ω 是 R^n 中的紧集, 目标函数 $f(X)$ 是目标空间 R^m 上的连续函数, 则 VDCMOAPO 强收敛于其约束 Pareto 前沿。

证明:

由 VDCMOAPO 算法中个体的速度和位置进化方程可得,

$$\Delta x = x(t+1) - x(t) = wV + \lambda F / m \quad (6.13)$$

令 $wV = \mu$, $F/m = \sigma$, 则 $\Delta x = \mu + \lambda \sigma$ 。其中, μ, σ 均为常数项。由于 λ 是服从正态分布的随机变量, 那么正态分布的线性组合也服从正态分布。因此, $\Delta x \sim N(\mu, \sigma^2) = (N(\mu_1, \sigma_1^2), N(\mu_2, \sigma_2^2), \dots, N(\mu_n, \sigma_n^2))^T$, 且 Δx 的 n 个分量 $\Delta x_1, \Delta x_2, \dots, \Delta x_n$ 之间相互独立。

设 $y_0 = f(X_0)$ 为约束 Pareto 前沿上的点, 其中 $X_0 = (x_0^1, x_0^2, \dots, x_0^n) \in \Omega$

$\because f(X)$ 是连续函数

$\therefore \exists r > 0$, 当 $\|X - X_0\|_\infty \leq r$ 时有 $\|f(X) - f(X_0)\|_\infty \leq \varepsilon$

$\therefore f(X) \in N_\varepsilon(PF_i)$

记 $D_0 = \{x \in \Omega \mid \|f(X) - f(X_0)\| < \varepsilon\}$, 即 D_0 是决策空间 Ω 中可行非劣最优解的 ε

邻域;

$D_1 = \Omega \setminus D_0$, 即 D_1 是决策空间 Ω 中除去非劣最优解 ε 邻域的剩余部分。

VDCMOAPO 算法中产生的群体可以分为两种状态:

S1: 在决策空间 Ω 中至少存在一个个体属于 D_0 ;

S2: 决策空间中所有个体都不属于 D_0 , 都属于 D_1 ;

则 VDCMOAPO 算法在 $t+1$ 代产生的群体与在第 t 代产生的群体的关系有如下四种:

- (1) t 代时状态为 S1 的点集在 $t+1$ 代状态为 S1;
- (2) t 代时状态为 S1 的点集在 $t+1$ 代状态为 S2;
- (3) t 代时状态为 S2 的点集在 $t+1$ 代状态为 S1;
- (4) t 代时状态为 S2 的点集在 $t+1$ 代状态为 S2.

只要证明在 t 趋向 ∞ 时 (2) 和 (4) 的概率趋于一个很小的常数, 即可证明 VDCMOAPO 强收敛于其约束 Pareto 前沿。(2) 表示 VDCMOAPO 算法产生的群体在 t 代时收敛, 而在 $t+1$ 代时不收敛, 根据 VDCMOAPO 算法可知, (2) 的概率为 0.

记 $D_{x_0, r} = \{X \in \Omega \mid \|X - X_0\|_{\infty} \leq r\}$, 对于 $X = (x_1, x_2, \dots, x_n) \in \Omega$,

$$\begin{aligned}
 & P\{(X + \Delta X) \in D_{x_0, r}\} \\
 &= \prod_{i=1}^n P\{|x_i' + \Delta x_i' - x_0^i| \leq r\} \\
 &= \prod_{i=1}^n P\{x_0^i - x_i' - r \leq \Delta x_i' \leq x_0^i - x_i' + r\} \\
 &= \prod_{i=1}^n \int_{x_0^i - x_i' - r}^{x_0^i - x_i' + r} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(y - \mu_i)^2}{2\sigma_i^2}} dy
 \end{aligned}$$

所以 $0 < P\{(X + \Delta X) \in D_{x_0, r}\} < 1$.

$\therefore P\{(X + \Delta X) \in D_{x_0, r}\}$ 是紧集 Ω 上的连续函数

$\therefore \exists X_1', X_0' \in \Omega$,

使得 $P\{(X_1' + \Delta X') \in D_{x_0', r}\} = \min P\{(X + \Delta X) \in D_{x_0, r}\}$ 且 $0 < P\{(X_1' + \Delta X') \in D_{x_0', r}\} < 1$

所以 $P\{(X + \Delta X) \in D_{x_0, r}\} \geq P\{(X_1' + \Delta X') \in D_{x_0', r}\} = \min P\{(X + \Delta X) \in D_{x_0, r}\}$

$\therefore P\{PF(t+1) \cap N_{\varepsilon}(PF_t) \neq \emptyset \mid PF(t) \cap N_{\varepsilon}(PF_t) = \emptyset\} \geq P\{(X_1' + \Delta X') \in D_{x_0', r}\}$

$B_t = P\{PF(t+1) \cap N_{\varepsilon}(PF_t) = \emptyset \mid PF(t) \cap N_{\varepsilon}(PF_t) = \emptyset\} \leq 1 - P\{(X_1' + \Delta X') \in D_{x_0', r}\} = c$

$$c \in (0,1)$$

$$\sum_{i=1}^{\infty} B_i' = B_0 + B_0 B_1 + \dots + B_0 B_1 \dots B_n \leq c + c^2 + \dots + c^n = \frac{c}{1-c} < \infty$$

对于 $\forall \varepsilon > 0$ ，由定理 6.1 可知，VDCMOAPO 算法强收敛于其约束 Pareto 前沿。
证毕。

6.2.3 实验及结果分析

为了测试 VDCMOAPO 算法的性能，我们利用该算法对三个典型的约束多目标测试函数（Binh2 测试函数、SRN 测试函数和 DEB 测试函数，如表 6.1 所示）进行了仿真实验。实验中群体 P 和 $archive$ 集的规模设置均为 100，进化代数 为 30 代（现有算法一般进化代数为 2000 代以上）。其中第一个测试函数的约束 Pareto 最优区域是目标空间上的一段连续区域，第二个测试函数的 Pareto 最优解集是无约束 Pareto 最优解集的一个子集，第三个测试函数的一部分非约束 Pareto 最优区域是不可行的。图 6.2、6.3 和 6.4 是利用 VDCMOAPO 算法对每个测试函数进行测试的约束 Pareto 前沿。

从图中可以看出，利用 VDCMOAPO 对这三个测试函数进行多目标优化得到的解基本上都在真实的约束 Pareto 前沿上，并且分布比较均匀。所以本节提出的 VDCMOAPO 算法在处理某些约束多目标优化问题时具有较好的性能。

表 6.1 VDCMOAPO 算法测试函数

Problem	Variable bounds	Objective functions	Constraints
Binh(2)	$x_1 \in [0, 5]$	$\min f_1(x_1, x_2) = 4x_1^2 + 4x_2^2$	$(x_1 - 5)^2 + x_2^2 - 25 \leq 0$
	$x_2 \in [0, 3]$	$\min f_2(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 5)^2$	$-(x_1 - 8)^2 - (x_2 + 3)^2 + 7.7 \leq 0$
SRN	$x_i \in [-20, 20]$	$\min f_1(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$	$x_1^2 + x_2^2 - 225 \leq 0$
	$i = 1, 2$	$\min f_2(x_1, x_2) = 9x_1 - (x_2 - 1)^2$	$x_1 - 3x_2 + 10 \leq 0$
DEB	$x_1 \in [0.1, 1.0]$	$\min f_1(X) = x_1$	$-9x_1 - x_2 + 6 \leq 0$
	$x_2 \in [0, 5]$	$\min f_2(X) = (1 + x_2) / x_1$	$-9x_1 + x_2 + 1 \leq 0$

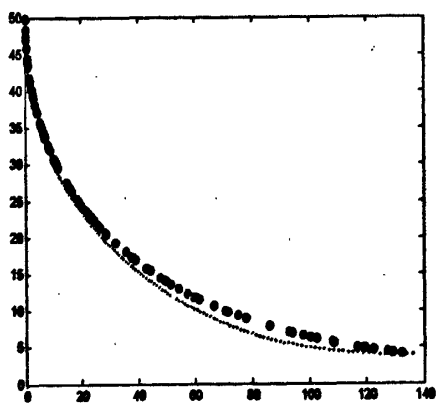


图 6.2 VDCMOAPO 对测试函数 Binh(2) 的约束 Pareto 前沿

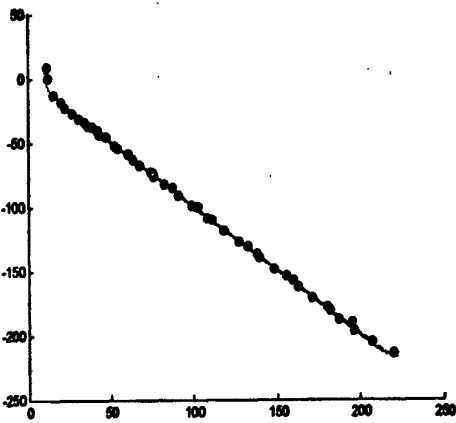


图 6.3 VDCMOAPO 对测试函数 SRN 的约束 Pareto 前沿

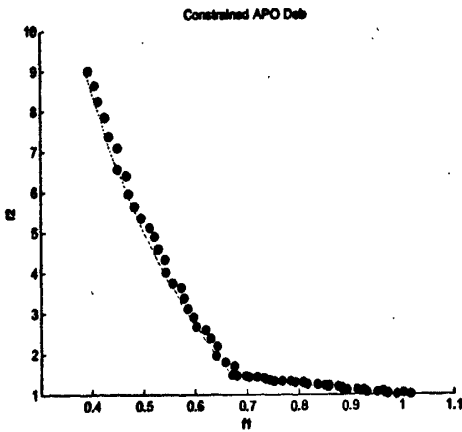


图 6.4 VDCMOAPO 对测试函数 DEB 的约束 Pareto 前沿

6.3 基于约束保持法的 CMOAPO 算法研究

约束保持法是一种最容易理解的约束处理方法,它通过确保进化过程中所有个体都在可行域范围之内来缩小搜索范围,从而加快搜索速度。这种方法要求初始化时个体必须满足所有的约束条件,因此对于有些问题需要花费很长时间来产生可行初始解,甚至有些问题很难产生可行解。但是对于有些问题,特别是一些工程应用中较为简单的约束多目标优化问题,该方法具有较快的搜索效率。

本节采用约束保持法作为约束处理技术,结合第五章基于序值的无约束 MOAPO 算法的基本思想,提出一种基于序值的约束多目标拟态物理学优化 (CRMOAPO) 算法,通过数值实验说明了该算法的有效性。

6.3.1 基于序值的约束多目标拟态物理学优化算法

求解约束多目标优化问题与求解无约束多目标优化问题的目标相同,都是找到可行的非支配解集,使其充分接近且均匀分布于可行的 Pareto 最优前沿,只是约束多目标优化算法需要对约束进行处理。如果采用一种方法使得算法进化的过程在可行域内进行,那么约束多目标优化问题就可以转化成无约束多目标优化问题。

通过第五章的数值实验及分析可知, RMOAPO 算法充分体现了多目标优化问题的特征,具有较好的性能,所以将约束保持法与 RMOAPO 算法相结合,提出一种 CRMOAPO 算法,其基本思想如下:

首先在可行域内随机生成 $PopSize$ 个个体位置,在速度约束范围内产生每个个体的初始速度。然后用 RMOAPO 算法对初始种群进行优化,即计算邻域半径、计算个体质量及所受其他个体的虚拟力、个体的运动及速度更新。有关 RMOAPO 算法在第五章已经进行了详细描述,在此不再赘述。接下来考查更新后的个体位置是否在可行域范围内,如果不在可行域内,则将其拉至到可行域内。具体方法就是如果个体 i 在第 k 维上超出了上界,则将其拉至到该维的上界,即

$$x_k(i) = x_k(\max) \quad (6.14)$$

其中 $x_k(i)$ 表示个体 i 在第 k 维上的位置分量, $x_k(\max)$ 表示可行域的第 k 维的上界;同理,如果个体 i 在第 k 维上超出了下界,则将其拉至到该维的下界,即

$$x_k(i) = x_k(\min) \quad (6.15)$$

其中 $x_k(\min)$ 表示可行域的第 k 维的下界。在确保更新后个体均位于可行域内的前提下,计算约束非劣解,更新 *archive* 集合。

CRMOAPO 算法步骤如下:

Step1 初始化参数: 种群规模 $PopSize$, *archive* 集大小 $archivesize$, 引力常数 G , 惯性权重 w , 最大迭代次数 $Maxit$ 等。在可行域内随机产生每个个体每一维初始

位置 x_i 和初始速度 v_i , 根据公式 (5.1), 计算邻域半径 r 。

Step2 根据 RMOAPO 算法中质量函数的计算方法得到个体质量 m_i , 根据公式 (2.7) 及公式 (2.8) 计算个体受到其他个体的虚拟力及所受合力。

Step3 根据公式 (2.9) 及 (2.10) 更新个体 i 速度和位置。考查个体 i 每一维的位置是否超出可行域范围, 如果个体 i 在第 k 维上超出了上界, 则根据公式 (6.14) 将其拉至可行域内; 如果超出了下界, 则根据公式 (6.15) 将其拉至可行域内。

Step4 计算每个个体在每个目标下的函数值, 保留非支配个体, 修剪 *archive* 集。

Step5 若达到最大迭代次数则退出, 否则返回 Step2。

6.3.2 实验及结果分析

为了测试 CRMOAPO 算法的性能, 我们利用该算法对 6.2.3 节中的三个典型的约束多目标测试函数中的 Binh2 测试函数和 SRN 测试函数 (如表 6.1 所示) 进行了实验。种群规模为 30, *archive* 集大小为 50, 惯性权重 w 从 0.9 线性减小到 0.4, 引力常数 G 取经验值 $G=10$, 最大进化代数 100。得到的结果如图 6.5 和图 6.6 所示。

从图中我们可以看出, 通过 CRMOAPO 算法得到的 Pareto 前沿基本都在真实 Pareto 前沿上, 且分布较为均匀, 所以 CRMOAPO 算法能够有效处理某些约束多目标优化问题。

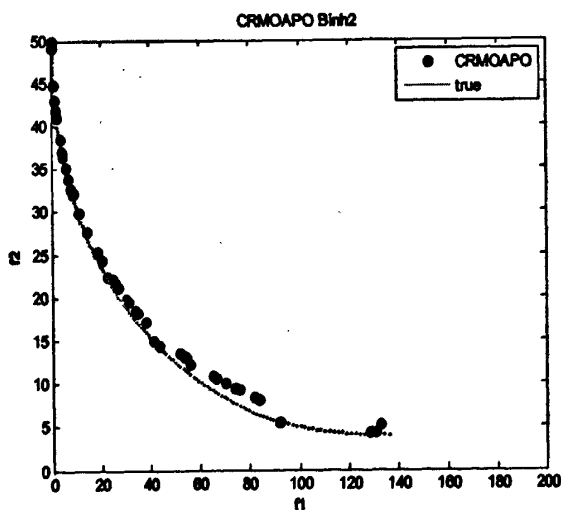


图 6.5 CRMOAPO 对测试函数 Binh(2) 的约束 Pareto 前沿

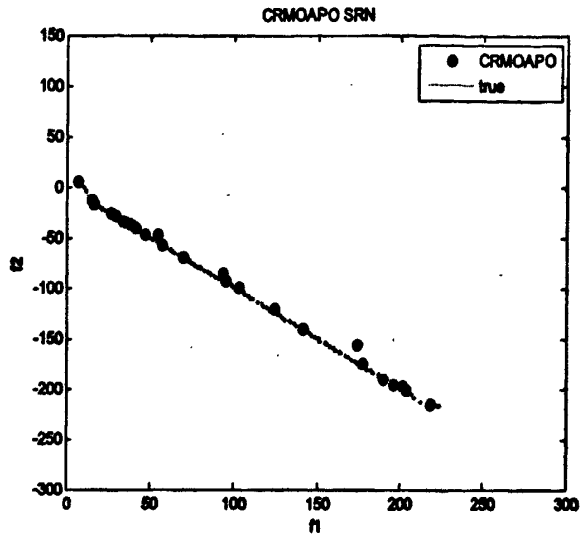


图 6.6 CRMOAPO 对测试函数 SRN 的约束 Pareto 前沿

6.4 本章小结

本章首先介绍了目前约束多目标进化算法的发展状况及存在的问题, 然后结合约束多目标优化问题的特点及 APO 算法自身特性, 将可行规则法作为约束处理技术, 提出了一种 VDCMOAPO 算法。该算法针对可行解在更新后成为不可行解的情况, 减小了其他个体作用在该个体上的虚拟力, 从而有效避免上述状况的发生。然后利用概率论中的基本知识, 对该算法的收敛性进行了理论分析及证明, 同时通过实验说明了所提出算法的有效性。最后采用约束保持法作为约束处理机制, 结合 RMOAPO 算法思想, 提出了一种 CRMOAPO 算法, 并通过仿真实验测试了该算法性能。

第7章 基于多目标拟态物理学优化算法的移动机器人 路径规划

7.1 引言

移动机器人是机器人技术研究的一个重要分支,是近年来发展起来的一门综合学科,是一个集环境感知、动态决策与规划、行为控制与执行等多种功能于一体的综合系统,集中了机械、电子、计算机、自动控制以及人工智能等多个学科的最新研究成果,代表了机电一体化的最高成就^[115],已成为一个跨学科、跨领域的研究热点。移动机器人具有广阔的应用前景,其应用范围和功能由于机器人技术的不断发展而不断拓展和提高,不仅在工业、国防、服务等行业中得到广泛的应用,而且在野外作业以及在有害、危险环境作业等领域中以人类无法比拟的巨大优势而崭露头角^[116]。路径规划是移动机器人领域的一项重要研究内容,是反映移动机器人实现智能化及自主工作的关键技术,其任务就是在具有障碍物的环境内,按照一定的评价标准,寻找一条从起始状态(包括位置和姿态)到目标状态的无碰路径^[117]。移动机器人路径规划的研究始于20世纪70年代,目前国内外许多学者仍在从事着大量的这方面的研究工作。

根据环境建模方式的不同,移动机器人路径规划方法可以分为三种类型^[118]:基于环境先验完全信息的路径规划(有时也称为离线规划或全局规划);基于传感器信息的不确定环境的路径规划(有时称为在线规划或局部规划);基于行为的路径规划。实际上,全局路径规划与局部路径规划并没有本质区别,大多数适用于全局路径规划的方法经过改进都可以用于局部路径规划,同时,适用于局部路径规划的方法一般也都适用于全局路径规划。根据搜索策略的不同,移动机器人路径规划方法可以分为传统方法和智能方法,常见的传统路径规划方法有:自由空间法^[119,120]、可视图法^[121]、栅格法^[122,123]和人工势场法^[124,125]。常见的智能路径规划方法有:基于神经网络的移动机器人路径规划算法^[126,127]、基于遗传算法的移动机器人路径规划^[128-134]、基于微粒群的移动机器人路径规划算法^[135-139]、基于模糊逻辑的移动机器人路径规划算法^[140]、基于蚁群算法的移动机器人路径规划算法^[141]及其他智能路径规划方法^[142]。

具有代表性的传统的移动机器人路径规划方法有以下几种:

1. 可视图法

可视图法是 Nilsson 于 1968 年提出的。这类方法把移动机器人看作一个点,以多

边形障碍物模型为基础,将任意形状的障碍物用近似于该形状的多边形代替,将机器人、目标点和多边形障碍物的各个顶点进行连接,并且保证机器人和障碍物各顶点之间、目标点和障碍物各顶点之间以及各障碍物顶点与顶点之间的连线,均不能穿越障碍物,从而形成了一张图,称之为可视图。由于任意两直线的顶点都是可视的,因而移动机器人从起点沿着这些连线到达目标点的所有路径均是无碰路径。由于不是所有的边都能成为最终路径,所以在一些简化的可视图法中不必要的连线往往被删除,以缩短搜索时间,提高算法效率。这类方法能够求得最短路径,但由于机器人被看作是一个点,其尺寸被忽略不计,使得机器人有时会离障碍物太近,甚至与障碍物接触,并且这类方法的不足就是搜索时间较长,缺乏灵活性,存在组合爆炸问题。

2. 自由空间法

自由空间法的基本思想是首先采用预先定义的基本图形构造自由空间,将这些基本单元表示成一个拓扑意义上的连通图,然后运用图搜索方法来对路径进行搜索。为了使问题简化,自由空间法通常采用“结构空间”来描述机器人及其周围的环境,其中最具代表性的是 Voronoi 图法和四叉树(Quadtree)及其扩展算法。Voronoi 图法的基本思想是:首先产生到障碍物的所有边界点等距离的 Voronoi 边,称这些边之间的交点为 Voronoi 顶点。然后,移动机器人只能行走在这类 Voronoi 边上,这样,机器人不仅不会与障碍物相碰撞,而且一定在任意两个障碍物的中间。四叉树是一种递归网格,首先建立移动机器人所处环境的二维直角坐标网格,将机器人所处环境划分为若干较大的网格单元。如果障碍物占用了网格单元的元素,就把这部分分成四个小格子(四叉树)。如果这四个小格子中还有被占据的单元则递归地对该单元再分割成更小的四个子网格。

自由空间法将机器人缩小成点,将其周围的障碍物及边界按比例相应地扩大,使机器人点能够在障碍物空间中移动到任意一点,而不与障碍物及边界发生碰撞。这类路径规划方法比较灵活,起始点和目标点的改变不会造成连通图的重构,但其复杂程度与障碍物的个数成正比,并且有时无法获得最短路径。

3. 栅格法

栅格法是将移动机器人的工作环境分解为一序列,即具有二值信息的网格单元,不含有任何障碍物的栅格是自由栅格,否则为障碍栅格,自由空间和障碍物均可表示为栅格块的集成。这些栅格组成一个连通图,在这个连通图上搜索一条从起始栅格到目标栅格的路径,这条路径可以用栅格序号表示。这类方法将栅格作为基本单位记录环境信息,栅格的粒度直接影响着环境信息存储量的大小以及路径搜索的时间,所以在实用上具有一定的局限性。

4. 拓扑法

拓扑法根据环境信息和移动机器人的几何特点,将规划空间分割成具有拓扑特征

一致的子空间,根据彼此连通性建立拓扑网络,从该网络中寻找一条从起始点到目标点的拓扑路径,最终由拓扑路径求出几何路径。该类方法的基本思想就是降维,将高维几何空间中求路径的问题转化为在低维拓扑空间中辨别连通性的问题。该类方法的优点是充分利用拓扑特征而缩小了搜索空间,算法的复杂度与障碍物的数目相关,理论上是完备的;而且拓扑法通常不需要机器人的准确位置,对于位置误差具有较好的鲁棒性。该类方法的缺点主要是建立拓扑网络的过程非常复杂,特别是当障碍物发生变化时,如何有效地修正已经存在的拓扑网络是需要解决的问题。

5. 人工势场法 (Artificial Potential Field)

人工势场法最初由 Khatib 提出,这种方法由于它的简单性而被广泛采用。其基本思想是把移动机器人在已知全局环境中的运动看作一种虚拟的人工受力场中的运动。目标点对机器人产生引力作用,障碍物对机器人产生斥力作用,引力和斥力的合力控制机器人的运动。该方法结构简单,易于实现,但是这种方法也存在着一些缺点,如:存在陷阱区、在相近的障碍物前不能发现路径、在障碍物前产生振荡以及在狭窄通道中摆动等缺点。

移动机器人路径规划存在以下特点:

- (1) 复杂性:在复杂环境中,机器人路径规划非常复杂,而且计算量很大。
- (2) 多约束性:机器人的形状、速度、加速度及障碍物等对机器人的运动存在约束。
- (3) 多目标性:机器人运动过程中路径性能要求存在多种目标,如路径最短,最光滑,安全性能最好等,但它们之间往往存在冲突。

本质上移动机器人的路径规划问题具有多个优化目标,然而目前移动机器人中的路径规划算法大多采用单目标约束优化方法进行处理,或采用加权求和法将问题的多个性能指标组合成一个标量函数,使之转化为单目标优化问题进行求解。加权求和法简单直观,但该方法运行一次只能得到一个解,并且解的值依赖于权重的选取。为解决该问题,本章首先设计一种基于障碍物信息的环境模型,同时,根据移动机器人的路径规划需求构造适应值函数。然后根据环境模型中已知的障碍物信息给出约束条件,从而构成问题的状态空间,将移动机器人路径规划问题转换成连续空间中的约束多目标优化问题,利用约束多目标拟态物理学优化算法进行搜索,算法运行一次便可以得到问题的一组 Pareto 最优解。决策者可以依据当前需要,从中挑选出一个或一些“足够满意”的解作为问题的最终解,进而找到从起始点到终止点的满意路径。

7.2 环境描述

本文研究的移动机器人路径规划问题是指移动机器人在一个具有障碍物的环境

中从起始点无碰撞地移动到目标点,不研究机器人的姿态变化。为了方便研究,我们做如下假定:①移动机器人在二维有限空间中运动,不考虑其高度信息;②在机器人移动过程中,障碍物的几何形状、位置及大小均不发生变化;③把障碍物边界向外扩展机器人本体在长、宽方向上最大尺寸的 $\frac{1}{2}$,机器人可看作质点。

路径规划的第一步就是要建立合适的环境模型,本章建立一种基于障碍物顶点的路径规划环境模型。该模型经每一个障碍物顶点作 Y 轴的平行线,如图7.1所示,这样每个障碍物的横坐标已知,只要控制路径与障碍物交点的纵坐标以及相邻节点的连线不经过障碍物即可。

设移动机器人出发点为 S ,目标点为 G ,图7.1中不规则的多边形表示障碍物。移动机器人的路径规划就是寻找在该环境中由 S 运动至 G 的过程中所经过的无碰撞点序列的集合,每一个序列代表一条路径,所以每一条路径可以表示为:

$$P = \{p_0, p_1, \dots, p_n, p_{n+1}\}, p_0 = S, p_{n+1} = G \quad (7.1)$$

式(7.1)中点的序列 $\{p_1, \dots, p_n\}$ 为规划目标,对于该序列中任意一点 p_i 的要求是: p_i 为非障碍点,且 p_i 与其邻点 p_{i-1} 和 p_{i+1} 的连线不经过障碍物。

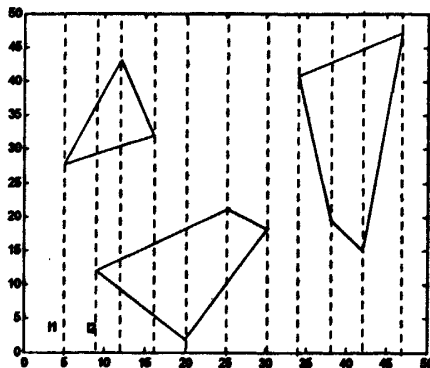


图 7.1 移动机器人工作环境模型

图7.1中有3个障碍物,每个障碍物在二维空间的顶点已知,如果以每个障碍物顶点横坐标为定值,纵坐标为可变量,则可产生图7.1中一簇平行于纵轴的竖直虚线 $\{l_1, \dots, l_n\}$,如果有若干个障碍物顶点的横坐标相同,则它们具有同一条竖直虚线,移动机器人从出发点 S 到目标点 G 的无碰路径 P 与该平行直线族的交点即为目标点序列 $\{p_1, \dots, p_n\}$,显然,每个交点的横坐标已知,如果把纵坐标作为竖直虚线上可上下移动的未知可变量,不规则多边形障碍物为约束限制,这样可得一个移动机器人从起始点到终止点的无碰路径长度:

$$f_1(P) = L_P = L_{S, p_1} + \sum_{j=1}^{n-1} |p_j p_{j+1}| + L_{p_n, G} = \sum_{j=0}^n |p_j p_{j+1}| = \sum_{j=0}^n \sqrt{(X_{j+1} - X_j)^2 + (Y_{j+1} - Y_j)^2} \quad (7.2)$$

考虑图 7.1 中的平行线族 $\{l_1, \dots, l_n\}$, 如果将个体的每一维位置分量对应于每条直线上的一点, 这样不仅可以方便地找到问题解的表达, 而且个体的每一维位置分量均可具有明确的物理意义, 很容易将移动机器人路径规划问题与连续空间的约束多目标优化问题进行映射: 移动机器人路径规划问题中的每条可行路径映射为多目标优化问题搜索空间中的一个个体; 移动机器人路径规划问题中的每条路径与每条竖直虚线的交点映射为多目标优化算法中每个个体的一维位置分量; 移动机器人路径规划问题中的规划目标映射为多目标优化算法中的目标函数。

7.3 移动机器人路径规划建模

现存的大多数移动机器人路径规划算法往往仅把路径的长度作为最终优化目标, 而未考虑路径的平滑性。通常机器人的转动与机器人的运动时间有直接关系, 因为每次角度较大的转动, 机器人必须减慢速度甚至停止, 这样走完从起点到终点的路程所用的时间就会大大增加。并且如果转动的角度等于 $\pi/2$ 时, 说明机器人前进的方向与指向终点的方向垂直, 若大于 $\pi/2$, 则说明机器人前进的方向背离了指向终点的方向, 所以在设计移动机器人路径规划算法时, 这些情况应尽量避免, 因此路径的平滑性对于移动机器人路径规划也是非常重要的。基于上述原因, 现将路径平滑度作为路径规划问题中的另一个优化目标, 其定义如等式 (7.3) 所示:

$$f_2(P) = \frac{\sum_{j=2}^n a_j + k \cdot \frac{\pi}{2}}{n-1} \quad (7.3)$$

其中 a_j 表示两个向量 $p_{j-1}p_j$ 和 p_jp_{j+1} 间的夹角 ($0 \leq a_j \leq \pi$); k 为 a_j 中大于或者等于 $\pi/2$ 的个数, 即当某一夹角大于或等于 $\pi/2$ 时, 对目标值进行惩罚。

综合以上所述, 设任意一条路径为 $P = [p_0, p_1, \dots, p_n, p_{n+1}]$, 其中 p_0 为起始点 S , p_{n+1} 为目标点 G , 存在若干个障碍物 $O_i (i=1, 2, \dots, h)$, 则移动机器人路径规划多目标优化模型可以表示为:

$$\begin{aligned} F(P) &= \min\{f_1(P), f_2(P)\} \\ P &= \{p_0, p_1, \dots, p_{n+1}\}, \quad p_0 = S, p_{n+1} = G \\ P \cap O &= \Phi, \quad O = \{O_1, O_2, \dots, O_h\} \end{aligned} \quad (7.4)$$

其中 $p_i (i=0, 1, \dots, n+1)$ 是机器人经过的点的序列, 其连线构成机器人的路径。 $f_1(P)$ 是路径总长度, $f_2(P)$ 是路径的平滑度。

7.4 基于 CRMOAPO 的移动机器人路径规划算法

7.4.1 算法思想及流程

应用 MOAPO 算法解决移动机器人路径规划问题有两个关键步骤: 问题解的编码和适应值函数的选择。在 7.2 节中已讨论移动机器人路径规划问题与多目标优化问题的映射, 将问题解的编码进行了详细介绍, 在此不再赘述。对于适应值函数的选择, 本节在等式 (7.4) 所示的移动机器人路径规划的多目标优化模型上, 采用基于 Pareto 的序值概念和邻域共享机制来定义搜索空间中个体的适应值函数。应用 6.3 节提出的 CRMOAPO 算法对移动机器人路径规划问题进行求解。首先根据障碍物信息建立约束条件, 由于 CRMOAPO 算法采用约束保持法作为约束处理机制, 所以个体的初始位置是在可行域内随机均匀产生的。然后根据移动机器人路径规划的多目标优化的数学模型计算每个个体在每个子目标下的函数值, 将非支配个体保存在外部存储体中。采用基于序值的多目标拟态物理学优化算法的适应值函数的定义方式计算每个个体的质量, 从而计算个体受到其他个体的作用力及个体所受合力, 对个体进行更新。在个体更新之后考查每个个体是否在可行域范围内, 如果不在可行域内, 则将其拉至可行域, 再计算每个个体在每个子目标下的函数值, 选取非支配个体, 并将其与外部存储体中的个体比较, 更新和修剪外部存储体, 进入下一次迭代。

基于 CRMOAPO 的移动机器人路径规划算法步骤如下:

- Step1** 初始化参数: 种群大小 $PopSize$, 引力常数 G , 惯性权重 w , 最大循环次数 $Maxit$, 计数器 $t=0$ 。根据障碍物信息将障碍物顶点按照横坐标由小到大的顺序排列, 计算搜索空间的可行域, 在可行域内随机产生每个个体每一维初始位置 x_i (这里的 x_i 对应于路径节点上的纵坐标 Y_i) 和初始速度 v_i , 根据等式 (5.1), 计算邻域半径 r 。
- Step2** 根据公式 (7.2) 和公式 (7.3) 计算每个个体 $i(i=1, \dots, PopSize)$ 在每个目标下的函数值, 保留支配个体 i 的所有个体的数量 n_i , 找出非支配个体, 将其存储在 $archive$ 集中, 并给每个非支配个体分配序值为 1, 为其余个体分配序值 n_i+1 。
- Step3** 根据序值将群体中所有个体按升序排列。计算具有相同序值的个体 r 邻域内的个体数量, 按照该数量的升序将同序值个体排序, 对具有相同数量的同序值个体 i 进行标记 $flag(i)=1$ 。
- Step4** 如果 $flag(i)=1$, 则根据上述方法调整邻域半径, 如果调整后具有相同序值的个体的 r 邻域内的个体数量仍相同, 对这些个体的顺序随机排列。
- Step5** 根据上述排序结果将每个个体赋一个序号, 即为每个个体分配一个 1 到 N 间的自然数作为 $rank_i(t)$ 。

- Step6 根据公式 (5.2) 计算个体质量, 根据公式 (2.7)、(2.8) 计算个体受到的合力, 根据公式 (2.9)、(2.10), 更新个体速度和位置。
- Step7 考查更新后的每个个体每一维位置是否在可行域内, 若不在可行域内, 如果超出上界, 则根据公式 (6.14) 调整其更新位置; 如果超出下界, 则根据公式 (6.15) 调整其更新位置。
- Step8 计算每个个体在每个目标下的函数值, 保留支配每个个体的个体数量 n_i ($i=1,...,PopSize$), 找出非支配个体, 并给每个非支配个体分配序值为 1, 为其余个体分配序值 n_i+1 。将每一个非支配个体与存储在 *archive* 集中的个体比较, 若该个体既不被 *archive* 集中个体支配, 也不支配 *archive* 集中的个体, 则将该个体存入 *archive* 集中; 若该个体支配 *archive* 集中的个体, 则将被支配个体从 *archive* 集中删除, 将非支配个体放入 *archive* 集中。若需要存储的个体超过 *archive* 集的规模, 则将支配其他个体数量最少的个体删除。
- Step9 若达到最大迭代次数则退出, 否则返回 Step3。

7.4.2 仿真试验及结果分析

为了验证 CRMOAPO 算法的有效性, 我们在 PC 机上对基于多目标拟态物理学优化的移动机器人路径规划算法进行了仿真实验, 实验的主要参数设置如下: 移动机器人工作环境为 50×50 , 群体规模为 50, *archive* 集大小为 20, 惯性权重 w 从 0.9 线性减小到 0.4, 引力常数 G 从 50 线性减小到 1, 最大进化代数 为 100。我们分别对具有 3 个不规则障碍物 (记作环境 1) 和具有五个不规则障碍物 (记作环境 2) 的机器人工作环境中的路径规划进行了 20 次独立实验, 得到的结果如表 7.1 所示, 并将两个环境下第 10 次路径规划的结果以图形形式输出。

表 7.1 不同环境下 CRMOPAO 路径规划算法指标

CRMOAPO	环境 1	环境 2
起点	0, 25	0, 25
终点	50, 17	50, 25
平均找到的路径数量	3.5	2.5
路径的平均长度	73.12	60.81
路径的平均平滑度	0.6673	0.4716
平均运行时间	9.7518	17.0445

图 7.2 所示三个不规则形状障碍物, 有横坐标不同的障碍物顶点 11 个, 所以每个个体有 11 维位置分量。从图中我们可以看到通过本文所提算法找到了三条不同的

路径, 决策者可以根据需要从三条路径中选择合适的作为移动机器人的行走路径。图 7.3 所示五个不规则形状的障碍物, 使得机器人的工作空间的变得更加复杂, 通过障碍物信息可知, 该环境模型中有横坐标不同的障碍物顶点 19 个, 所以每个个体有 19 维位置分量。从图中我们可以看到本文算法找到了两条不同的路径。

从图 7.2 和图 7.3 及其相应的实验结果可以看出, 本文提出的算法针对障碍物较多的情况效果较好, 这是因为障碍物较多往往使得具有不同横坐标的障碍物顶点较多, 这样可产生较多的平行于纵轴的竖直虚线 $\{l_1, \dots, l_n\}$, 即个体的维数越大, 规划出的路径性能越好, 但同时计算量也随着维数的增加而增大, 所花费时间也就越多。如果需要获得性能较好的路径, 可以在根据障碍物顶点产生个体维数的基础上再增加个体维数, 这样就可以较为容易地得到一组性能较好的 Pareto 最优路径。

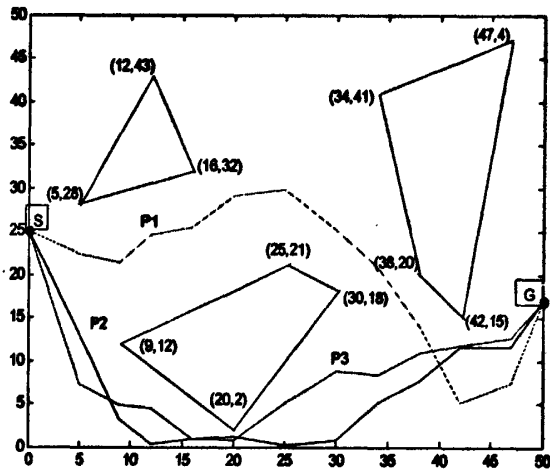


图 7.2 较少障碍物的 CRMOAPO 路径规划结果

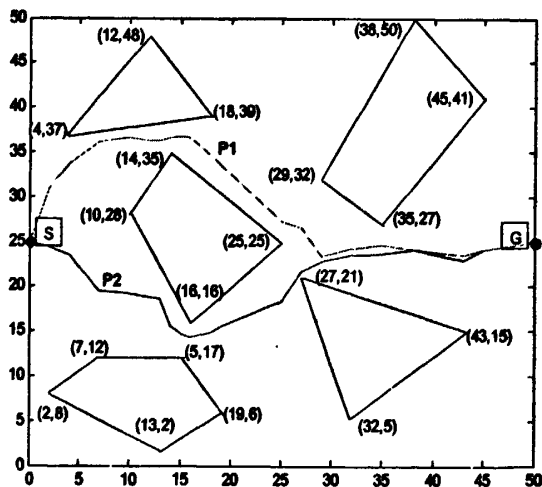


图 7.3 较多障碍物的 CRMOAPO 路径规划结果

为了充分说明 CRMOAPO 算法在解决移动机器人路径规划问题的有效性,我们在同一台 PC 机上,利用 NSGA-II 算法在环境 1 与环境 2 下针对公式 (7.4) 所示的移动机器人路径规划的多目标优化模型进行了求解,所得结果如表 7.2 所示。

表 7.2 不同环境下 NSGA-II 路径规划算法指标

NSGA-II	环境 1	环境 2
起点	0, 25	0, 25
终点	50, 17	50, 25
平均找到的路径数量	3.5	3
路径的平均长度	72.97	61.14
路径的平均平滑度	0.6438	0.4625
平均运行时间	11.2397	18.6546

对比表 7.1 与表 7.2 可以发现,通过 CRMOAPO 算法产生的路径与 NSGA-II 产生的路径在平均长度和平均平滑度方面大体相似,对于障碍物较少的环境 1, NSGA-II 产生的路径在上述两个方面略有优势,而对于障碍物较多的环境 2, CRMOAPO 算法产生的路径稍好一些。但在两个环境下, CRMOAPO 算法的运行时间较之 NSGA-II 算法要短。

7.5 基于 VDCMOAPO 的移动机器人路径规划算法

7.5.1 算法思想及流程

基于 VDCMOAPO 算法的移动机器人路径规划的问题编码与 7.4 节基于 CRMOAPO 算法的相同,都是采用 7.2 节中的问题编码方式,数学模型为公式 (7.4) 所示的移动机器人路径规划的多目标优化模型,适应值函数基于聚集函数的方法进行定义。

在 7.2 节中讨论的移动机器人路径规划问题的编码方法是基于障碍物信息的,如果在障碍物非常多的工作环境下,则问题的可行域就会相对狭小,在这种情况下采用可行规则法作为约束处理机制,可能会找不到可行解(虽然这种情况很少)。为了保证能够产生满意的可行解,在算法初始化过程中,首先在问题可行域内产生一条初始路径,其余路径在问题空间内随机均匀产生,同时在速度的限定范围内随机均匀产生每个个体的初始速度。然后根据移动机器人路径规划的多目标优化的数学模型计算每

个个体在每个子目标下的函数值, 将非支配个体保存在外部存储体中。采用 6.2 节中 VDCMOAPO 算法的适应值函数的定义方式计算每个个体的质量, 同样采用 VDCMOAPO 算法中的作用力规则计算个体受到其他个体的作用力及个体所受合力, 对个体进行更新, 计算每个个体在每个子目标下的函数值, 选出非支配个体, 对外部存储体进行更新与修剪。

基于 VDCMOAPO 算法的移动机器人路径规划算法步骤如下:

- Step1** 初始化参数: 种群大小 $PopSize$, 引力常数 G , 惯性权重 w , 最大循环次数 $Maxit$, 计数器 $t=0$ 。根据障碍物信息将障碍物顶点按照横坐标由小到大的顺序排列, 计算搜索空间的可行域, 在可行域内随机产生一个个体, 同时在问题域内随机均匀产生剩余个体每一维初始位置 x_i (这里的 x_i 对应于路径节点上的纵坐标 Y_i), 在速度限制范围内随机均匀产生每个个体的初始速度 v_i 。
- Step2** 根据约束条件对个体进行判断: 可行个体存放于 *feasible* 集合, 不可行个体存放在 *infeasible* 集合。对于可行个体, 根据公式 (7.2) 和 (7.3) 计算每个个体 $i(i=1, \dots, N)$ 在每个目标下的函数值, 找出非支配个体, 将其与存储在 *archive* 集中的个体比较, 删除被支配个体, 如果非支配个体数量超出 *archive* 集合, 则对其进行修剪; 对于不可行个体, 根据公式 (6.2) 计算其不可行度。
- Step3** 根据公式 (6.3) 计算种群中每个个体质量, 根据公式 (6.4-6.7) 计算每个个体在每一维上受到其他个体对它的作用力, 根据公式 (2.8) 计算个体所受到的合力。
- Step4** 根据公式 (2.9) 及公式 (2.10) 更新个体的速度和位置。判断更新后的个体是否由可行域进入到不可行域, 如果存在这种状况, 将该个体受到的合力减小至原来一半, 然后再根据 (2.9) 及公式 (2.10) 重新更新个体的速度和位置。
- Step5** 若满足终止条件则输出 *archive* 集合中个体并退出, 否则转 Step2。

7.5.2 仿真实验与结果分析

为了验证 VDCMOAPO 算法的有效性, 我们在同一台 PC 机上对其进行了仿真实验, 实验的主要参数设置如下: 机器人环境为 50×50 , 群体规模为 50, *archive* 集大小为 20, 惯性权重 w 从 0.9 线性减小到 0.4, 引力常数 G 从 50 线性减小到 1, 最大进化代数 100。我们分别对 7.4 节所述的环境 1 和环境 2 利用基于 VDCMOAPO 的路径规划进行了 20 次独立实验, 得到的结果如表 7.3 所示, 并将两个环境下第 10 次路径规划的结果以图形形式输出, 如图 7.4、7.5 所示。

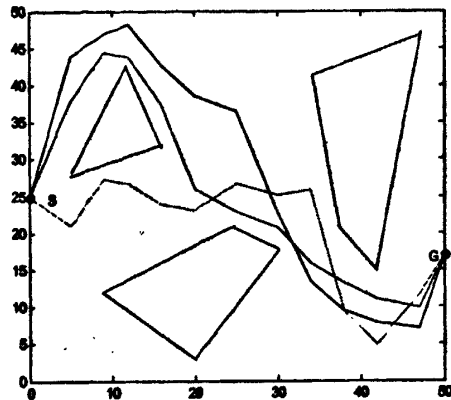


图 7.4 较少障碍物的 VDCMOAPO 路径规划结果

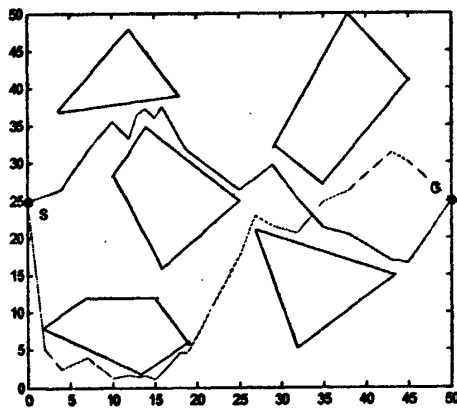


图 7.5 较多障碍物的 VDCMOAPO 路径规划结果

从图中可以看出, 利用 VDCMOAPO 算法得到的路径在光滑度方面要比用 CRMOAPO 算法得到的路径较差, 在路径中存在一些折线, 这样路径长度将增加。这种情况的产生主要因为 VDCMOAPO 算法采用可行规则法作为约束处理机制, 而这种约束处理机制比较容易陷入局部最优

表 7.3 不同环境下 VDCMOAPO 路径规划算法指标

VDCMOAPO	环境 1	环境 2
起点	0, 25	0, 25
终点	50, 17	50, 25
平均找到的路径数量	4	3
路径的平均长度	106.76	73.94

路径的平均平滑度	1.4058	1.3288
平均运行时间	8.9512	15.7424

通过表 7.1 与表 7.3 的比较可以看出, CRMOAPO 算法得到的路径在平均长度与平均平滑度方面性能较好, 而基于 VDCMOAPO 的移动机器人路径规划算法在平均找到的路径数量及平均运行时间方面具有较好的性能。因为基于 CRMOAPO 的路径规划算法的约束处理机制是约束保持法, 每次只在可行域内产生路径, 如果个体越过了可行域的限制, 则将其拉至可行域。在每一次个体更新之后, 算法都要判断更新后的个体是否在可行域内, 这样算法的运行时间就会增加。

通过仿真实验结果我们可以得到结论: 对于工程应用中的移动机器人路径规划问题, 在对路径长度及平滑度性能要求较高的情况下, 采用基于 CRMOAPO 的移动机器人路径规划算法较为合适; 如果对算法运行时间或算法健壮性要求较高的移动机器人路径规划问题中, 则较适合采用基于 VDCMOAPO 的移动机器人路径规划算法。

7.6 本章小结

本章利用多目标拟态物理学优化算法对移动机器人路径规划问题进行了建模。首先介绍了目前移动机器人路径规划发展的状况, 分析了该领域存在的一些问题。然后建立了基于障碍物顶点信息的移动机器人工作环境, 构建了移动机器人路径规划问题的多目标优化数学模型, 从而将路径规划问题转换成连续空间的约束多目标优化问题。接下来采用约束保持法作为约束处理机制, 通过 CRMOAPO 算法对移动机器人路径进行了优化求解。通过仿真实验及与基于经典多目标进化算法的移动机器人路径规划算法在相同环境下的比较可知, 利用约束多目标拟态物理学优化算法对移动机器人路径规划问题进行优化求解是有效的, 算法运行一次可以得到一组 Pareto 最优路径解, 从而可以保证移动机器人路径规划的健壮性。然后采用可行规则法作为约束处理机制, 利用 VDCMOAPO 算法对移动机器人路径进行了规划, 最后通过两种基于不同约束机制的多目标拟态物理学优化算法规划的路径的比较可知, 对路径长度和平滑度要求较高的工程应用中适合采用基于 CRMOAPO 的路径规划算法; 对路径长度及平滑度要求不是很高但对算法运行时间要求较高的路径规划, 较为适合采用基于 VDCMOAPO 的移动机器人路径规划算法。

结论与展望

本文将拟态物理学优化算法引入到多目标优化领域, 构建了多目标拟态物理学优化算法, 从算法框架建立、算法有效性分析、种群多样性保持、适应值函数构建、约束多目标拟态物理学优化算法以及基于移动机器人路径规划的应用等方面进行了理论与应用研究。论文所做的主要工作归纳如下:

1. 在剖析了具有代表性的基于种群的多目标优化技术的基础上, 通过比较拟态物理学优化算法与典型的基于种群的优化算法的异同, 分析了将拟态物理学优化算法引入到多目标优化领域需要解决的关键问题: 能够体现多目标优化问题中个体适应值优劣的质量函数的建立、融合多目标优化问题及拟态物理学优化特点的种群多样性保持机制构建、最优个体保留机制及外部存储体的修剪技术, 从而阐述了构建多目标拟态物理学优化算法的可行性。在此基础上建立了多目标拟态物理学优化算法的基本框架。
2. 借鉴传统多目标优化算法中聚集函数的基本思想, 提出了一种基于聚集函数的无约束多目标拟态物理学优化算法。针对算法前期需要较强的全局搜索能力, 而后期需要精细搜索的特点, 给出了对算法具有重要影响的两个参数——惯性权重及引力常数随进化过程而线性变化的一种策略。通过仿真实验说明了多目标拟态物理学优化算法的有效性。
3. 结合多目标优化问题的特点, 从拟态物理学优化机制出发, 分析了多目标拟态物理学优化算法对 Pareto 非劣解集多样性保持技术, 提出了一种基于虚拟力排序的无约束多目标拟态物理学优化算法, 利用概率论的基本知识, 对该算法的收敛性进行了理论分析与证明。通过仿真实验及与经典多目标进化算法的比较, 说明了所提算法在解集多样性保持方面具有较好性能。
4. 针对多目标拟态物理学优化算法的适应值函数的构建问题, 分析了基于 Pareto 概念的个体优劣与个体质量间的关系, 将序值及个体邻域拥挤度作为个体适应值评价标准, 通过质量函数来体现个体适应值, 构建了一种充分体现多目标优化问题自身特点的质量函数。在此基础上, 提出了一种基于序值的无约束多目标拟态物理学优化算法。选用一组复杂的测试函数, 对所提算法进行了仿真测试, 通过与经典多目标进化算法及基于聚集函数的多目标拟态物理学优化算法比较, 说明了该算法具有较好性能。
5. 针对目前约束多目标优化问题的研究相对较少的情况, 将拟态物理学优化算法引入到约束多目标优化领域。首先采用可行规则法作为多目标约束处理机制, 融合

拟态物理学优化特点,提出了一种基于虚拟力减小的约束多目标拟态物理学优化算法,利用概率论的基本知识,对算法的收敛性进行了理论证明,通过仿真实验说明了该算法的有效性。然后将约束保持法作为多目标约束处理机制,提出了一种基于序值的约束多目标拟态物理学优化算法,对其进行了仿真测试。

6. 基于移动机器人路径规划问题,对多目标拟态物理学优化算法进行了应用研究。在依据障碍物信息的基础上,建立了移动机器人的工作环境,从而将路径规划问题转换成连续空间的多目标优化问题。分析了移动机器人路径规划的特点,将路径长度与路径平滑度作为优化目标,构建了移动机器人路径规划的多目标优化模型,分别利用约束保持法和可行规则法作为约束处理机制,提出了两种约束多目标拟态物理学优化算法,对移动机器人路径规划问题进行求解优化。仿真实验表明,利用多目标拟态物理学优化算法可以有效解决移动机器人路径规划问题。同时,通过对两种路径规划算法的性能比较,给出了这两种路径规划算法的工程应用领域。

本文设计的多目标拟态物理学优化算法为解决多目标优化问题提供了一条新的途径,对多目标优化领域的发展具有一定的理论意义和应用价值。诚然,作为一种新型的多目标优化工具,多目标拟态物理学优化算法有很大的发展空间,今后拟在以下几个方面进行更深入的研究:

1. 提高多目标拟态物理学优化算法对大范围决策域、高维目标空间以及大规模优化问题的处理能力。由于高维目标空间的多目标优化问题中个体间的关系更多地趋向于“等同”,所以该问题的研究对传统 Pareto 支配概念提出挑战,虽然目前已有一些支配关系的提成,但均未形成主流。因此下一步工作将寻找一种适合多目标拟态物理学优化的个体支配机制,有效解决复杂的高维多目标优化问题。
2. 多目标拟态物理学优化算法参数设计,特别是自适应参数调整。算法的参数对算法的性能有非常重要的影响,从理论上对算法参数进行分析,对进一步提高算法性能有着重要意义。
3. 充分融合拟态物理学优化算法及多目标优化问题特点,改进现有算法,提出新的高效的多目标拟态物理学优化算法,分析其所适用的问题域求解对象范围,以及工程应用的领域面向。
4. 分析现有的基于种群的多目标优化算法中的成功算法,将其成果与多目标拟态物理学优化算法有效融合,提出高效的混合算法,从而进一步提高算法的求解效率及性能。
5. 能够充分体现多目标优化问题特点的约束处理机制的研究。现有的约束多目标优化算法大都是沿袭了单目标约束优化算法的约束处理机制,而未真正体现出多目标的特点,所以对其进行理论分析与研究,将有重要的实际意义。

参考文献

- [1]. Zitzler Evolutionary Algorithms for Multi-objective Optimization: Methods and Applications:[Swiss Federal Institute of Technology Zurich for the degree of Doctor of Technical Sciences] Swiss: Swiss Federal Institute of Technology Zurich,1999
- [2]. 崔逊学.基于进化算法的多目标优化算法研究:[中国科技大学博士学位论文].合肥:中国科技大学, 2001
- [3]. Carlos A. Coello Coello. Evolutionary Multi-Objective Optimization: Some Current Research Trends and Topics that Remain to Be Explored. Front Computer Science China 2009, 3(1):18-30
- [4]. Carlos A. Coello Coello and Landa Becerra R. Evolutionary Multi-Objective Optimization in Materials Science and Engineering. Materials and Manufacturing Processes, 2009, 24(2):119-129
- [5]. Rangaiah Gade Pandu. Multi-Objective Evolutionary Algorithms: A Review of the State-of-the Art and Some of Their Applications in Chemical Engineering. Singapore: World Scientific, 2009, 61-90
- [6]. 张敏.约束优化及多目标优化研究:[中国科技大学博士学位论文].合肥:中国科技大学, 2008
- [7]. 汪祖柱.基于演化算法的多目标优化方法及其应用研究:[安徽大学博士论文].合肥:安徽大学, 2005
- [8]. 刘桂琴.基于微型遗传算法的多目标优化方法及应用研究:[湖南大学博士学位论文].长沙:湖南大学, 2008
- [9]. 郑金华.多目标进化算法及其应用.北京:科学出版社, 2007
- [10]. 公茂果,焦李成,杨咚咚等. 进化多目标优化算法研究.软件学报, 2009, 20(2): 271-289
- [11]. 崔逊学.多目标进化算法. 北京:国防工业出版社, 2008
- [12]. Farina M, Amato P. On the Optimal Solution Definition for Many-Criteria Optimization Problems. In : Proceedings of the NAFIPSFLINT International conference'2002, Piscataway: IEEE Service Center, 2002, 233-238
- [13]. Carlos A.Coello Coello An Empirical Study of Evolutionary Techniques for Multi-Objective Optimization in Engineering Design: [Tulane University for the

- degree of Doctor]. Department of Computer Science, Tulane University, New Orleans, LA, 1996: 3-13
- [14]. L.A.Zadeh. Optimality and Nonscalar-valued Performance Criteria. IEEE Transactions on Automatic Control, 1963, 8(1):59-60
- [15]. A.M.Geoffrion. Proper Efficiency and the Theory of Vector Optimization. Journal of Mathematical Analysis and Application, 1968, 41:491-502
- [16]. Y Y. Haimes, L.S.Lasdon, D.A.Wisner. On a Bicriterion Formulation of the Problems of Integrated Systems Identification and System Optimization. IEEE Transactions System, Man, Cybernetics, 1971, 1(3):296-297
- [17]. 杨保安,刘巍. 专家系统与目标规划相结合的多目标决策分析研究. 系统工程理论与实践, 1992, 3,18-22
- [18]. Saúl Zapotecas Martínez and Carlos A. Coello Coello. A Proposal to Hybridize Multi-Objective Evolutionary Algorithms with Non-Gradient Mathematical Programming Techniques. In: Parallel Problem Solving from Nature-PPSN X, Germany, 2008, 837-846
- [19]. Carlos A.Coello Coello. Evolutionary Multi-Objective Optimization: Past, Present and Future. In: Workshops and Tutorials of the 7th International Conference on Parallel Problem Solving from Nature. Granada Spain, Sep. 2002
- [20]. Deb K. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In: Proceeding of Evolutionary Algorithms in Engineering and Computer Science, 1999
- [21]. J.J. Durillo, A.J. Nebro, Carlos A. Coello Coello, et al. A Study of Multi-Objective Metaheuristics When Solving Parameter Scalable Problems. IEEE Transactions on Evolutionary Computation, August 2010, 14(4): 618-635
- [22]. Oliver Schütze, Carlos A. Coello Coello, Sanaz Mostaghim,et al. Hybridizing Evolutionary Strategies with Continuation Methods for Solving Multi-Objective Problems. Engineering Optimization, 2008, 40(5):383-402
- [23]. Datta D., Fonseca C. M. and Deb K. A Multi-Objective Evolutionary Algorithm to Exploit the Similarities of Resource Allocation Problems. Journal of Scheduling, 2008, 11(6):405-419
- [24]. Deb, K., Sundar, J., Reddy, Uday, B., and Chaudhuri, S. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. International Journal of Computational Intelligence Research (IJCIR), 2006, 2(6): 273-286
- [25]. Schaffer J D. Multiple Objective Optimization with Vector Evaluated Genetic

- Algorithms. In: Proceedings of the First International Conference on Genetic Algorithms. Lawrence Erlbaum, 1985: 93-100
- [26]. Goldberg DE. Genetic Algorithm for Search, Optimization and Machine Learning. Boston: Addison-Wesley Longman Publishing Co. Inc. 1989
- [27]. Fonseca C M, Fleming P J. Genetic Algorithms for Multi-Objective Optimization: Formulation, Discussion and Generalization. In: Presented at Proceedings of the 5th International Conference on Genetic Algorithms, San Mateo, California, 1993
- [28]. Srinivas N, Deb K. Multi-Objective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 1994, 2:221-248
- [29]. Horn J, Nafpliotis N, Goldberg D E. A Niche Pareto Genetic Algorithm for Multi-Objective Optimization. In: IEEE World Congress on Evolutionary Computation, Piscataway, NJ, 1994
- [30]. David A. Van Veldhuizen. Multi-Objective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, [Air Force Institute of Technology]. Wright-Patterson AFB, Ohio, 1999
- [31]. Zitzler E, Thiele L. Multi-Objective Evolutionary Algorithms: A Comparative Case Study at Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 1999, 3:257-271
- [32]. Zitzler E, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multi-Objective Optimization. In: 2001-Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Athens, Greece, 2002, 95-100
- [33]. Knowles J D, Corne D W. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multi-Objective Optimization. In: Congress on Evolutionary Computation (CEC99) Piscataway, NJ, 1999
- [34]. Corne D W, Knowles J D, and Oates M J. The Pareto Envelope-based Selection Algorithm for Multi-Objective Optimization. In: Proceedings of the Parallel Problem Solving from Nature VI Conference, Springer, 2000, 839-848
- [35]. Corne DW, Jerram NR, Knowles JD et al. PESA-II: Region-Based Selection in Evolutionary Multi-Objective Optimization. In: Genetic and Evolutionary Computation Conference. CECCO 2001, San Francisco: Morgan Kaufmann Publishers, 2001, 283-290
- [36]. Erickson M, Mayer A, Horn J. The Niche Pareto Genetic Algorithm2 Applied to the Design of Groundwater Remediation System. In: The 1st International

- Conference on Evolutionary Multi-Criterion Optimization. EMO2001. Berlin: Springer-Verlag, 2001, 681-695
- [37]. Carlos A. Coello Coello and G T Pulido. A Micro-Gentic Algorithm for Multi-Objective Optimization. In: Proc. of the Genetic and Evolutionary Computation Conf. GECCO 2001. San Francisco: Morgan Kaufmann Publishers, 2001, 274-282
- [38]. Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. In: Parallel Problem Solving from Nature (PPSN VI), Springer, 2000, 849-858
- [39]. Oliver Schuetze, Gustavo Sanchez and Carlos A. Coello Coello. A New Memetic Strategy for the Numerical Treatment of Multi-Objective Optimization Problems. In: 2008 Genetic and Evolutionary Computation Conference (GECCO'2008), Atlanta, USA, ACM Press, 2008, 705-712
- [40]. Juan J. Durillo, Antonio J. Nebro, Carlos A. Coello Coello, et al. A Comparative Study of the Effect of Parameter Scalability in Multi-Objective Metaheuristics. In: 2008 Congress on Evolutionary Computation (CEC'2008), Hong Kong, IEEE Service Center, 2008, 1893-1900
- [41]. 李密青, 郑金华. 基于邻域表的多目标进化算法. 小型微型计算机系统, 2009, 30(8):1645-1650
- [42]. Davarynejad M., Ahn C. W., Vrancken J. Evolutionary Hidden Information Detection by Granulation-Based Fitness Approximation. Applied Soft Computing, 2010, 10(3):719-729
- [43]. Fernández Eduardo, López Edy, Bernal Sergio, et al. Evolutionary Multi-Objective Optimization Using an Outranking-Based Dominance Generalization. Computers & Operations Research, 2010, 37(2):390-395
- [44]. M G Fabre, G T Pulido and. Carlos A Coello Coello. Alternative Fitness Assignment Methods for Many-Objective Optimization Problems. In: Artificial Evolution, 9th International Conference, Springer, Lecture Notes in Computer Science Vol. 5975, Strasbourg, France, 2010, 146-157
- [45]. M G Fabre, G T Pulido and. Carlos A Coello Coello. Ranking Methods for Many-Objective Problems. In: Advances in Artificial Intelligence. 8th Mexican International Conference on Artificial Intelligence, Guanajuato, México, 2009, (MICAI 2009) Springer, Lecture Notes in Artificial Intelligence Vol. 5845, 633-645

- [46]. A L Jaimes and Carlos A.Coello Coello. Study of Preference Relations in Many-Objective Optimization. In: 2009 Genetic and Evolutionary Computation Conference (GECCO'2009), ACM Press, Montreal, Canada, 2009, 611-618
- [47]. Laumanns M, Thiele L, Deb K et.al. Combining Convergence and Diversity in Evolutionary Multi-Objective Optimization. *Evolutionary Computation*, 2002, 10(3):263-282
- [48]. Brockhoff D, Zitzler E. Are All Objective Necessary on Dimensionality Reduction in Evolutionary Multi-Objective Optimization. In: *Parallel Problem Solving from Nature, PPSN IX, LNCS*, Berlin: Springer-Verlag, 2006, 533-542
- [49]. A. G. Hernández_Díaz, L. V. Santans-Quintero, Carlos A. Coello Coello, et al. Pareto-Adaptive ϵ -dominance. *Evolutionary Computation*, 2007, 15(4): 493-517
- [50]. Julián Molina, Luis V. Santana, Alfredo G, Hernández-Díaz et al. g-Dominance: Reference Point Based Dominance for Multi-Objective Metaheuristics. *European Journal of Operational Research*, 2009, 197(2): 685-692
- [51]. Eduardo Fernández, Edy López, Sergio Bernal, et al. Evolutionary Multi-Objective Optimization Using a Fuzzy-Based Dominance Concept. In: *International Conference on Engineering Optimization*, 2008, Brasil
- [52]. Korudu P, Das S, Welch shi. Multi-Objective Hybrid PSO Using μ -Fuzzy Dominance. In: *Proc. Of Genetic and Evolutionary Computation Conf., GECCO 2007*, New York: ACM Press, 2007, 853-860
- [53]. 刘鑒,李敏强,林丹. 基于 ϵ -支配的多目标进化算法及自适应 ϵ 调整策略. *计算机学报*, 2008, 31(7):3-12
- [54]. 李密青,郑金华,肖桂霞,等. 基于空间距离的多目标进化算法. *模式识别与人工智能*, 2009, 4(12):87-94
- [55]. 郑金华,史忠植,谢勇. 基于聚类的快速多目标遗传算法. *计算机研究与发展*, 2004, 41(7):44-50
- [56]. 李密青,郑金华,罗彪. 一种基于最小生成树的多目标进化算法. *计算机研究与发展*, 2009, 46(5): 101-111
- [57]. 石川,李清勇,史忠植. 一种快速的基于占优树的多目标进化算法. *软件学报*, 2007, 18(3):35-46
- [58]. 鲍培明,朱庆保.用于多目标进化的归一化排序非支配集构造方法. *电子学报*, 2009, 37(9):140-145
- [59]. 张敏,罗文坚,王煦法.一种基于正态分布交叉的 ϵ -MOEA. *软件学报*, 2009, 20(2):133-142

- [60]. 曾建潮,介婧,崔志华. 微粒群算法. 北京:科学出版社, 2004.
- [61]. Jacqueline Moore and Richard Chapman. Application of Particle Swarm to Multi-Objective Optimization. [Research Report] Department of Computer Science and Software Engineering, Auburn University, 1999
- [62]. Carlos A. Coello Coello and Maximino Salazar Lechuga. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In: Congress on Evolutionary Computation (CEC'2002), New Jersey, 2002, volume 2:1051-1056
- [63]. Tapabrata Ray, Tai Kang and Seow Kian Chye. Multi-Objective Design Optimization by an Evolutionary Algorithm. Engineering Optimization, 2001, 33(4):399-424
- [64]. Sierra M Reyes, Carlos A. Coello Coello. Improving PSO-Based Multi-Objective Optimization Using Crowding, Mutation and ϵ -Dominance. In: Proc. Of the 3rd Int'l Conf. Evolutionary Multi-Criterion Optimization, EMO 2005, Berlin: Springer-Verlag, 2005, 505-519
- [65]. Abido M A. Two Level of Nondominated Solutions Approach to Multi-Objective Particle Swarm Optimization. In: Proc. Of the Genetic and Evolutionary Computation Conf., GECCO 2007. New York: 2007. 726-733
- [66]. Juan J. Durillo, José García-Nieto, Antonio J. Nebro, et al. Multi-Objective Particle Swarm Optimizers: An Experimental Comparison. In: 5th International Conference on Evolutionary Multi-Criterion Optimization. Nantes, France, 2009, 495-509
- [67]. Antonio J. Nebro, Juan J. Durillo, Jose Garcia-Nieto, et al. SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization. In: 2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making, USA, 2009, 66-73
- [68]. Nadia Nedjah, Leandro dos Santos Coelho and Luiza de Macedo de Mourelle. Multi-Objective Swarm Intelligent Systems: Theory & Experiences. Berlin Heidelberg: Springer, 2010, 83-104
- [69]. Mostaghim S, Teich J. The Role of Dominance in Multi-Objective Particle Swarm Optimization Methods. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, 2003: 26-33
- [70]. 王辉,钱锋.基于拥挤度与变异的动态微粒群多目标优化算法. 控制与决策, 2008, 23(11):1238-1242
- [71]. 金欣磊,马龙华,刘波,等. 基于动态交换策略的快速多目标粒子群优化算法研究. 电路与系统学报, 2007, 12(2):78-82

- [72]. 李宁,邹彤,孙德宝,等. 基于粒子群的多目标优化算法. 计算机工程与应用, 2005, 23:43-46
- [73]. Salazar-Lechuga M, Rowe J E. Particle Swarm Optimization and Fitness Sharing to Solve Multi-Objective Optimization Problems. In: Proc. Of Congress on Evolutionary Computation, IEEE CEC 2005:1204-1211
- [74]. Leticia Cagnina, Susana Esquivel and Carlos A. Coello Coello. A Particle Swarm Optimizer for Multi-Objective Optimization. JCS&T, 2005, 5(4) :204-210
- [75]. 王小刚,李明杰,王福利,等. 一种新的多目标粒子群算法的研究与应用. 东北大学学报(自然科学版), 2008, 29(10):1377-1380
- [76]. 李中凯,谭建荣,冯毅雄,等. 基于拥挤距离排序的多目标粒子群优化算法及其应用. 计算机集成制造系统, 2008, 14(7):1329-1336
- [77]. 宋武,郑金华. 基于密度熵的多目标粒子群算法. 计算机工程与应用, 2007, 43(26):41-44
- [78]. 王丽萍,江波,邱飞岳. 基于决策偏好的多目标粒子群算法及其应用. 计算机集成制造系统, 2010, 16(1):140-148
- [79]. K.E. Parsopoulos and M.N. Vrahatis. Particle Swarm Optimization Method in Multi-Objective Problems. In: Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002), Madrid, Spain, 603-607
- [80]. U. Baumgartner, Ch. Magele and W. Renhart. Pareto Optimality and Particle Swarm Optimization. IEEE Transactions on Magnetics, 2004, 40(2):1172-1175
- [81]. 张利彪,周春光,马铭,等. 基于粒子群算法求解多目标优化问题. 计算机研究与发展, 2004, 41(7):1286-1291
- [82]. Xiaohui Hu and Russell C. Eberhart. Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization. In: IEEE CEC'2002, vol.2: 1677-1681
- [83]. Xiaohui Hu, Russell C. Eberhart and Yuhui Shi. Particle Swarm with Extended Memory for Multiobjective Optimization. In: Proc.2003 IEEE Swarm Intelligence Symp, Indianapolis, 2003, 193-197
- [84]. K.E. Parsopoulos , D. K. Tasoulis and M.N. Vrahatis. Multi-Objective Optimization Using Parallel Vector Evaluated Particle Swarm Optimization. In: International Conference on Artificial Intelligence and Applications (AIA 2004), 2004, 823-828
- [85]. 张利彪,周春光,刘小华,等. 求解多目标优化问题的一种多子群体进化算法. 控制与决策, 2007, 22(11):1313-1320
- [86]. Gregorio Toscano Pulido and Carlos A. Coello Coello. Using Clustering

- Techniques to Improve the Performance of a Particle Swarm Optimizer. In: GECCO'2004, Washington, USA, 2004, 225-237
- [87]. Carlos A. Coello Coello, Gregorio Toscano Pulido and Maximino Salazar Lechuga. Handling Multiple Objectives with Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation. 2004, 8(3):256-279
- [88]. 熊盛武,刘麟,王琼,史敏. 改进的多目标粒子群算法.武汉大学学报(理学版), 2005, 51(3):308-312
- [89]. S.Chamaani, S.A. Mirtaheri, M.Teshnehlal,et al. Modified Multi-Objective Particle Swarm Optimization for Electromagnetic Absorber Design. In: Electromagnetics Research, PIER. 2008: 353-366
- [90]. 杨俊杰,周建中,方仍存,等. 基于自适应网格的多目标粒子群优化算法. 系统仿真学报, 2008, 20(21):5843-5847
- [91]. Gregorio Toscano Pulido, Carlos A. Coello Coello and Luis Vicente Santana-Quintero. EMOPSO: A Multi-Objective Particle Swarm Optimizer with Emphasis on Efficiency. In: Evolutionary Multi-Criterion Optimization, 4th International Conference, Matshushima, Japan, 2007, 272-285
- [92]. 蒋浩,郑金华,陈良军. 一种求解多目标优化问题的粒子群算法. 模式识别与人工智能, 2007,20(5):606-610
- [93]. 胡德峰,张步涵,姚建光. 基于改进粒子群算法的多目标最优潮流计算. 电力系统及其自动化学报, 2007, 19(3):51-57
- [94]. Spears W M, Spears D F, Kerr W, et al. An overview of Physicomimetics. Lecture Notes in Computer Science-State of the Art Series, 2005, No.3342, pp.84-97
- [95]. Xie L P, Zeng J C. A Global Optimization Based on Physicomimetics Framework. In: The 2009 World Summit on Genetic and Evolutionary Computation (GEC'09), Shanghai, P. R. China, 2009
- [96]. Xie L P, Zeng J C, Cui Z H. Using Artificial Physics to Solve Global Optimization Problems. In: the 8th IEEE International Conference on Cognitive Informatics (ICCI 2009), Hong Kong, 2009
- [97]. Xie L P, Zeng J C. The Performance Analysis of Artificial Physics Optimization Algorithm Driven by Different Virtual Force. ICIC Express Letters (ICIC-EL), 2009, 4(1):239-244
- [98]. Xie L P, Zeng J C, Cui Z H. On mass effects to artificial physics optimization algorithm for global optimization problems. International Journal of Innovative Computing and Applications (IJICA), 2010, 2(2):69-76

- [99]. 谢丽萍. 基于拟态物理学优化的全局优化算法研究:[兰州理工大学博士学位论文].兰州:兰州理工大学, 2010
- [100]. Liping Xie, Ying Tan, Jianchao Zeng, et al. Artificial Physics Optimization: A Brief Survey. *International Journal of Bio-Inspired Computation*, 2010, 2(5): 291-302
- [101]. Carlos A. Coello Coello and Gregorio Toscano Pulido. Multi-Objective Structural Optimization Using a Micro-Genetic Algorithm. *Structural and Multidisciplinary Optimization*, 2005, 30(5):388-403
- [102]. Ziler E, Laumanns M. Test Problems and Test Data for Multiobjective Optimizers [EB/OL] (30. Jun. 2008)
<http://www.tik.ee.ethz.ch/sop/download/supplementary/testProblemSuite/>
- [103]. Rudolph G. On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. In: *Proceedings of the 15th IEEE Conference Evolutionary Computation*, Anchorage AK, 1998, 511-516
- [104]. Hanne Thomas. On the Convergence of Multi-Objective Evolutionary Algorithms. *European Journal of Operational Research*, 1999, 117(3):553-564
- [105]. 邹秀芬,刘敏忠,吴志健,康立山. 解约束多目标优化问题的一种鲁棒的进化算法. *计算机研究与发展*, 2004, 41(6):985-990
- [106]. 刘淳安,王宇平. 约束多目标优化问题的进化算法及其收敛性. *系统工程与电子技术*, Feb.2007, 29(2):277-280
- [107]. 周育人,闵华清,许孝元,等. 多目标演化算法的收敛性研究. *计算机学报*, Oct.2004, 27(10):1415-1424
- [108]. Ruhul Sarker, Hussein A. Abbass, and Samin Karim. An Evolutionary Algorithm for Constrained Multi-Objective Optimization Problems. In: *the 5th Australasia-Japan Joint Workshop University of Otago, Dunedin, New Zealand*, 2001
- [109]. Fernando Jiménez, Antonio F. Gómez-Skarmeta, Gracia Sánchez, and Kalyanmoy Deb. An Evolutionary Algorithm for Constrained Multi-objective Optimization. In: *Proceedings of the 2002 IEEE Congress on Evolutionary Computation (CEC'2002)*, Honolulu, Hawaii, 2002. 1133-1138
- [110]. Aguirre A H, Rionda S B, Coello C A C, Handling Constraints Using Multiobjective Optimization Concepts. *International Journal for Numerical Methods in Engineering*, 2004,59:1989-2017
- [111]. 王跃宣,刘连臣,牟盛静,等. 处理带约束的多目标优化进化算法. *清华大学学报(自然科学版)*, 2005, 45(1):103-106

- [112]. 孟红云,张小华,刘三阳. 用于约束多目标优化问题的双群体差分进化算法. 计算机学报, 2008, 31(2):228-235
- [113]. 尚荣华,焦李成,马文萍,等. 用于约束多目标优化的免疫记忆克隆算法. 电子学报, 2009, 37(6):1289-1294
- [114]. 孙超利. 面向机械优化设计的微粒群算法理论及其应用研究:[太原科技大学博士学位论文].太原:太原科技大学, 2010
- [115]. 张捍东,郑睿,岑豫皖. 移动机器人路径规划技术的现状与展望. 系统仿真学报, 2005, 17(2): 439-443
- [116]. 陈少斌. 自主移动机器人路径规划及轨迹跟踪的研究:[浙江大学博士学位论文].杭州:浙江大学, 2008
- [117]. 成伟明. 移动机器人自主导航中的路径规划与跟踪控制技术研究:[南京理工大学博士学位论文].南京:南京理工大学, 2007
- [118]. 朱大奇,颜明重. 移动机器人路径规划技术综述. 控制与决策, 2010, 25(7):961-967
- [119]. L. Srikanthan. High-Speed Environment Representation Scheme for Dynamic Path Planning. Journal of Intelligent and Robotic Systems, 2001, 32(3):307-319
- [120]. A. Amos, G. Wilfried. Analytic Path Planning Algorithms for Bipedal Robots without a Trunk. Journal of Intelligent and Robotic Systems, 2003, 36(2):109-127
- [121]. X.Deng, A.Mirzaian.Competitive Robot Mapping with Homogeneous Markers. IEEE Trans. on Robotics and Automation, 2002, 12(4):532-542
- [122]. Jaillet L, Simeon T. Path Deformation Roadmaps: Compact Graphs with Useful Cycles for Motion Planning. Int. J of Robotics Research, 2008, 27(11):1175-1188
- [123]. Cai C H, Ferrari S. Information-Driven Sensor Path Planning by Approximate Cell Decomposition. IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, 2009, 39(3):672-689
- [124]. YANG S X, MAX M. Neural Network Approaches to Dynamic Collision Free Trajectory generation. IEEE Trans. on Systems, Man, and Cybernetics, Part B, 2001, 31(3):302-318
- [125]. Ge S S, Cui Y J. Dynamic Motion Planning for Mobile Robots Using Potential Field Method. Autonomous Robots, 2002, 13(2):207-222
- [126]. Janglova D. Neural Networks in Mobile Robot Motion. International Journal of Advanced Robotic Systems, 2004, 1(1):15-22
- [127]. 刘成良,张凯,付庄,等. 神经网络在机器人路径规划中的应用研究. 机器人, 2001, 23(7):605~607

- [128]. Kao-Ting Hung, Jing-Sin liu, and Yau-Zen Chang. A comparative study of smooth path planning for a mobile robot by evolutionary multi-objective optimization. In: IEEE International Symposium on Computational Intelligence in Robotics and Automation, Jacksonville, FL, USA, June 20-23, 2007
- [129]. 周兰凤,洪炳熔. 用基于知识的遗传算法实现移动机器人路径规划. 电子学报, 2006, 34(5): 911-914
- [130]. Oscar Castillo, Leonardo Trujillo, and Patricia Melin. Multiple Objective Genetic Algorithms for Path-Planning Optimization in Autonomous Mobile Robots . Soft Computing, 2007, 11:269-279
- [131]. Gihan NAGIB and W. GHARIEB. Path Planning for a Mobile Robot Using Genetic Algorithms. In: Electrical, Electronic and Computer Engineering, (ICEEC 2004), 2004, 185-189
- [132]. Kamran H Sedighi, Kaveh Ashenayi, Theodore W. Manikas, et al. Autonomous Local Path-Planning for a Mobile Robot Using a Genetic Algorithm. In: Congress on Evolutionary Computation (CEC2004), 2004, Vol.2,1338-1345
- [133]. S. Mitsit, K-D Bouzakis, D. Sagris et al. Robot Path Planning Optimization, Free of Collisions, Using a Hybrid Algorithm. In: Proceedings of the 3rd International Conference on Manufacturing Engineering (ICMEN), Chalkidiki, Greece, 2008, 475-484
- [134]. 闵华清,毕盛,陈亚鹏. 基于多目标遗传算法的机器人避障研究. 哈尔滨工业大学学报, 2005, 37(7):914-917
- [135]. 秦元庆,孙德宝,李宁,等. 基于粒子群算法的一点机器人路径规划. 机器人, 2004, 26(3): 222-225
- [136]. 孙波,陈卫东,席裕庚. 基于粒子群优化算法的移动机器人全局路径规划. 控制与决策, 2005, 20(9): 1052-1055
- [137]. 曲道奎,杜振军,徐殿国,等. 移动机器人路径规划方法研究. 机器人, 2008, 30(2): 97-102
- [138]. 薛英花,田国会,李国栋. 利用改进的粒子群算法的机器人全局路径规划. 华中科技大学学报, 2008, 36(Sup1): 167-170
- [139]. 雷开友,邱玉辉,刘博勤,等. 基于改进粒子群算法的移动机器人全局路径规划. 西南师范大学学报, 2006, 31(2): 103-106
- [140]. 杨翎鹏,李少远. 模糊优化算法及其在视觉机器人路径规划中的应用. 控制与决策, 2002, 12(sup1):723-726
- [141]. 樊晓平,罗熊,易晟,等. 复杂环境下基于蚁群优化算法的机器人路径规划. 控制

与决策, 2004, 19(2): 166-170

- [142]. 于乃功, 王胜, 阮晓钢. 基于细胞自动机的移动机器人路径规划算法. 控制与决策, 2010, 25(7): 1055-1058

致 谢

本论文从开始选题到全文的终结,都是在我的导师曾建潮教授的精心指导和悉心关怀下完成的,论文倾注了导师的汗水与心血。他给我们创造了一个良好的学习和研究环境,在他时刻地鞭策与鼓励下,我的论文才得以顺利完成。四年的博士生涯,曾老师渊博的学识、勤奋忘我的工作精神、在学术研究中敏锐的洞察力使我获益非浅,他严谨的科学态度和求实的科研作风给我树立了很好的榜样。在此我要向导师致以最衷心的感谢和深深的敬意!

感谢兰州理工大学电信学院的辛勤工作的老师们,特别是郝书记、李院长、尹老师,感谢兰州理工大学研究生学院的郑老师、卢老师、康老师及学院的各位领导和老师,是他们在学习期间给我们提供了很多的帮助。

感谢实验室所有的兄弟姐妹及兰州理工大学 07 级博士班同学,从与他们的交流和讨论中收获了许多启发与灵感,而且生活上的关心与帮助使得我的博士生活更加丰富多彩。

感谢我的父母公婆,正是他们的默默奉献与大力支持,让我有充足的精力,能全身心的投入到研究中。感谢我的爱人,他的宽容与理解化作点点滴滴平淡而浓厚的爱,陪伴着我渡过了许多难关。最后感谢我的宝贝女儿,她平时童真而富有哲理的话语,给我带来了无穷的快乐与勇气。

本论文得到了国家自然科学基金(60975074)的资助,在此表示感谢。

附录 A 攻读博士期间发表的学术论文

- [1]. 王艳,曾建潮. 一种基于拟态物理学优化的多目标优化算法. 控制与决策, 2010, 25 (7) :1040-1044(EI: 20103213128998)
- [2]. Yan WANG, Jian-chao ZENG, Zhi-hua CUI, Xiao-juan HE. A Novel Constraint Multi-Objective Artificial Physics Optimization Algorithm and Its Convergence. International Journal of Innovative Computing and Applications, IJICA , 2011, 3(2):61-70. (EI: 20111613915139)
- [3]. 王艳,曾建潮. 解决多目标优化问题的拟态物理学优化算法. 计算机工程, 2010, 36 (20) .188-190
- [4]. 王艳,曾建潮. 多目标微粒群优化算法综述. 智能系统学报, 2010, 5: 377-384
- [5]. Yan WANG, Jian-chao ZEGN, Ying TAN. An Artificial Physics Optimization Algorithm for Multi-Objective Problems Based on Virtual Force Sorting Proceedings. In: International Conference on Swarm, Evolutionary and Memetic Computing, (SEMCCO2010), Springer LNCS 6466, Chennai, India, Dec 16-18, 2010, pp. 615-622, (EI: 20110213561408).
- [6]. WANG Yan, ZEGN Jian-chao. A Constraint Multi-Objective Artificial Physics Optimization Algorithm. In: 2010 Second International Conference on Computational Intelligence and Natural Computing. (CINC2010). Wuhan, China, 2010, 107-112 (EI: 20110213570957).

