

浙江大学

硕士学位论文

非支配排序遗传算法（NSGA）的研究与应用

姓名：高媛

申请学位级别：硕士

专业：控制理论与控制工程

指导教师：卢建刚

20060301

非支配排序遗传算法 (NSGA)

的研究与应用

摘 要

多目标优化问题一直是科学和工程研究领域的难题和热点问题。传统的解决方法在处理大维数、多模态等复杂问题上存在许多不足。为了解决这一问题,国内外研究学者在简单遗传算法的基础上,发展出了多种多目标优化遗传算法。本文研究的非支配排序遗传算法(Non-dominated Sorting Genetic Algorithm, NSGA)及其改进算法 NSGA-II 就是其中发展较快、优化效果较好的一种方法。本文对算法的基本原理进行了系统的学习和研究,在算法的应用研究方面作了大胆的尝试,扩展了算法的应用范围,并提出了一种新的解决函数拟合问题的方法。

本文所作的主要工作如下:

- 1) 系统地介绍了遗传算法的一般流程和基本理论,以及目前已有的多目标优化遗传算法及其存在的一些问题。论述了 NSGA 的基本原理和不足之处,并对其改进算法 NSGA-II 提出的快速非支配排序法、拥挤度及其比较算子、精英策略及 NSGA-II 的主要流程作了详细的研究。
- 2) 将 NSGA-II 引入变结构控制系统 (Variable Structure Control System)。针对多输入变结构控制系统参数仅能凭经验选取的缺陷,利用 NSGA-II 对变结构控制的参数和滑平面参数进行优化,从而抑制了多输出控制系统的超调量,减少了系统的动态响应时间,降低了抖振,同时增强了系统对参数摄动和外部扰动的鲁棒性,仿真结果证明了该方法的有效性。
- 3) 将 NSGA-II 与 SVM 结合,提出了一种基于 SVM 和 NSGA-II 的函数拟合方法。传统的函数拟合方法,如支持向量机回归方法,需要人为选择参数和核函数,经验因素对拟合结果的影响很大。而本文提出的方法,首次将 SVM 的结构风险最小化准则引入 NSGA-II 算法,并实现了根据核函数的计算复杂

度自动选取核函数形式。仿真结果表明，该算法得到的拟合函数与原函数的曲线偏差很小，与传统的 SVM 的拟合方法相比更具优越性。

关键词：多目标优化、遗传算法、非支配排序遗传算法（NSGA）、NSGA-II、Pareto、参数优化、变结构控制、支持向量机

Non-dominated Sorting Genetic Algorithm (NSGA) and Its Applications

Abstract

Multi-objective optimization is a difficult problem and a research focus in the fields of science and engineering. Classical multi-objective optimization methods have several shortcomings in solving high dimension, multi-modal problems. In order to solve these problems, researchers have developed many multi-objects optimization genetic algorithms based on Simple Genetic Algorithm. Among them, Non-dominated Sorting Genetic Algorithm (NSGA) and its improved method (NSGA-II) have been developed quickly and applied widely. In this paper, basic theory of NSGA and NSGA-II are studied. The application fields of algorithms are extended, and a new scheme of function approximation is proposed.

The main contents are as follows:

- 1) The basic theory and process of Genetic Algorithm are systematically introduced. The present Multi-objects optimization genetic algorithms and its disadvantages are described. The basic theory of NSGA and its shortcomings are discussed. Fast Non-dominated Sorting Approach, Crowding Distance, Crowded-Compared Operator and Elitist Approach proposed by NSGA-II and its process are studied.
- 2) NSGA-II is introduced into Variable Structure Control System (VSC System). Aiming at the shortcoming that the control parameter can be only chosen by experience, parameters of the VSC System are optimized by NSGA-II. The overshoot and chattering of the MIMO system are reduced, the dynamic response time is shortened, and robustness is guaranteed to system parameter variations.

and external disturbance. Simulation results show the validity of the scheme.

- 3) A scheme of function approximation based on Support Vector Machine (SVM) and NSGA II is proposed. When using SVM regression, which is one of the traditional methods of function approximation, the parameters and kernel function should be chosen by experience. Factor of experience affects results too much. In this paper, Structural Risk Minimization Principle of SVM was firstly combined with NSGA II, and kernel function could be automatically chosen according to its computational complexity. Simulation shows that the approximation function the scheme gained is closer to the original than the conventional SVM function approximation.

Key Words: Multi-objects optimization, Genetic Algorithm, Non-dominated Sorting Genetic Algorithm (NSGA), NSGA-II, Pareto, Parameters optimization, Variable Structure Control, Supported Vector Machine

第一章 绪 论

1.1 引言

优化问题自古以来就是倍受人们关注的问题，尤其是在科技高速发展的今天，在工程技术应用过程中，我们经常要处理复杂的多目标优化问题。因此，研究多目标优化问题的解决方法已经成为目前学术界的热点之一。遗传算法自出现以来在各个领域得到了广泛的应用，在解决简单的单目标优化问题方面取得了很好的成果，于是，研究人员便考虑使用遗传算法解决多目标优化问题。然而，面对复杂的多目标优化问题，传统的遗传算法就显得力不从心了，各种困难迎面而来。面对困境，各国的研究人员通过对遗传算法进行改造，发展出了多种用于解决多目标优化问题的遗传算法，如多目标加权遗传算法，基于向量评估的遗传算法 (VEGA)^[1]，小组决胜遗传算法(NPGA)^[2]，基于字典序的方法，基于博弈论的遗传算法，多性别遗传算法^[3]，目标向量法^[4]，最大—最小方法^[5,6]，关系定理法，无后代遗传算法，随机权重和精英法，非支配排序遗传算法(Non-dominated Sorting Genetic Algorithm, NSGA)^[7]及其改进算法 NSGA-II^[8]等等。

本章将简要介绍遗传算法的产生和发展情况，以及已有的多目标优化遗传算法及其存在的一些问题，并对其中的非支配排序遗传算法 NSGA 及其第二代改进算法 NSGA-II 作进一步的介绍。最后，对本文所作的工作做概括性的说明。

1.2 多目标优化问题

在我们生活和工作当中，人们总是按照“价廉物美”这一基本优化准则进行决策，即人们总是期望成本最小化，效益最大化。这实际上就是一个多目标优化与决策问题。在现实生活中，多目标优化问题无处不在，尤其是在科学实践和工程设计中，绝大多数的优化问题都涉及到多个目标的优化，这些目标往往相互耦合、相互竞争，每个目标具有不同的物理意义和量纲。例如：控制工程中的时域

指标与稳定裕度、系统带宽等频域特性的综合问题就是一个典型的多目标工程优化设计问题；航天器总体设计中的有效载荷、射程、推力、燃料质量等指标的综合也是一个多目标优化问题；而社会发展与国民经济的中长远发展计划的优化与决策问题则是更为复杂的多目标优化与决策问题^[9]。对这样的多个目标进行同时处理确实十分复杂，难度很大。

自 20 世纪 60 年代初期以来，多目标优化问题就受到研究人员越来越多的关注，但却一直没有出现有效的解决方法。虽然已经有许多经典的方法可以很好的解决单目标优化问题，但其最优解的定义是不能推广到多目标优化问题中的。在单目标优化问题中，通常得到的最优解都是全局最大或最小解，而在多目标优化问题中，多个目标函数需要同时进行优化，由于目标之间无法比较，又存在矛盾冲突，导致不一定存在同时满足所有目标函数最优的解。某个解可能在一个目标上是最优的但在另一个上是最差的。因此，多目标问题通常存在一个解的集合，它们之间不能简单的进行比较好坏。对这种解来说，不可能进一步优化某一个或几个目标而不损坏其他目标，这种解称作非劣最优解集^[9]，也就是所谓的 Pareto 最优解集。早在 1896 年法国经济学家 Pareto 就提出了这一观点^[10]。传统数学规划原理的多目标优化方法在实际工程优化问题中往往表现出一定的脆弱性，因此有必要研究高效实用的多目标优化与决策的算法及理论^[9]。

近几年来，将遗传算法应用于多目标优化问题成为研究热点，这种算法通常称作多目标优化进化算法或多目标优化遗传算法。由于遗传算法的基本特点是多方向和全局搜索，这使得带有潜在解的种群能够一代一代地维持下来。从种群到种群的方法对于搜索 Pareto 解来说是十分有益的。

1.3 遗传算法的产生和发展^[11]

遗传算法 GA (Genetic Algorithms) 是受生物学进化学说和遗传学理论的启发而发展起来的，是一类模拟自然生物进化过程与机制求解问题的自组织与自适应的人工智能技术，是一种借鉴生物界自然选择和自然遗传机制的随机的搜索算法，由 Holland 教授于 1975 年提出^[12]。

遗传算法起源于 60 年代对自然和人工自适应系统的研究，最早的雏形出现

在生物学家 Fraser 的 1962 年的论文中^[13]，他尝试使用仿真来模拟带有突变和选择交互作用的进化过程。1967 年，Bagley 在他的博士论文中首次提出了遗传算法（Genetic Algorithm）一词，发表了遗传算法应用方面的第一篇论文，讨论了遗传算法在自动博弈中的应用。他提出的选择、交叉和变异等操作已经与目前的遗传算法中的操作十分相近。他还注意到在遗传算法进化过程的不同阶段应采用不同的选择概率，有利于防止遗传算法的早熟现象。同时，他提出了遗传算法自动调整的概念，交叉和变异的概率融入染色体本身的编码中，随着进化的进行自动调整算法。他还引入了适应度定标（scaling）的概念，这也是目前遗传算法中常用的技术。

1975 年美国密执根大学 Holland 教授^[12]出版了他的第一本关于遗传算法的专著《Adaptation in Natural and Artificial System》，同年 De Jong 发表的博士论文《An analysis of behavior of a class of genetic adaptive system》一起被公认为是遗传算法的基础^[14]。Holland 的著作系统地阐述了遗传算法的基本理论和方法，并提出了奠定遗传算法理论基础的模式定理（schemata theory）和隐形并行性原理。模式定理揭示了群体中的优良个体的样本数将以指数级规律增长，从理论上保证了遗传算法是可以用来寻求最优可行解的一种优化方法，该理论也首次确认了结构重组遗传操作对以后的隐并行性的重要性。De Jong 的论文中结合模式定理进行大量的数值函数优化计算试验，建立了遗传算法的工作框架，并提出了代沟（generation gap）等新的遗传操作技术，定义了评价遗传算法的性能指标。他建立的一组测试函数至今仍被广泛应用，其中包括非凸函数、不连续函数、带有随机变量的函数以及高维函数，被称作 De Jong 五函数测试平台。他所作的工作可以看成是遗传算法发展过程中的一个里程碑。

到了 80 年代早期，遗传算法已经广泛的应用于各个领域当中。1983 年 Goldberg 将遗传算法应用于管道系统的优化和机器学习问题^[15, 16]。1984 年 Fitzpatrick, Grefenstete 和 Van Gucht 利用遗传算法处理了医学图像变换问题。80 年代中期，Axelord 和 Forrest 合作研究了博弈论中的经典问题——囚徒困境问题^[17, 18]。

到了 80 年代末，遗传算法的发展达到了高潮^[19-26]。人们对遗传算法的兴趣

日益增长,其原因有二:其一是工程领域中不断涌现出超大规模的非线性系统,特别是人工智能与控制领域。经典的优化方法不能够有效的求解这些系统中存在的优化问题,而这些问题恰恰是大量存在并亟需解决的,如神经网络连接权重及网络拓扑结构的优化、模糊系统中模糊规则的选取等等。其二,遗传算法本身就是一种模拟自然演化这一学习过程的求解问题的方法,它能够以独立的或与其它方法相结合的形式用于智能机器学习系统的设计中。

1989年Goldberg首先提出了基于Pareto最优概念计算个体适应度的方法,借助非支配等级概念和相应的选择算子使种群在优化过程中朝Pareto最优解的方向进化。并出版了专著《搜索、优化和机器学习中的遗传算法(Genetic Algorithms in Search, Optimization and Machine Learning)》^[27]。该书全面的论述了遗传算法的原理及其应用,奠定了现代遗传算法的科学基础。

1991年,Davis编辑出版了《遗传算法手册(Handbook of Genetic Algorithms)》一书,书中涵盖了遗传算法在工程技术和科学计算等方面的大量应用实例^[28],为推广遗传算法的应用作出了贡献。

1992年,Koza将遗传算法应用计算机程序的优化设计及自动生成,提出了遗传编程(Genetic Programming,简称GP)的概念,并成功的应用于人工智能、机器学习等方面。

经过十几年的努力,遗传算法不论是在应用研究上,算法设计上,还是在基础理论上,均取得了长足的发展,已经成为信息科学、计算机科学、运筹学和应用科学等诸多学科所共同关注的热点研究领域。

遗传算法虽然在过去的20年中得到了广泛的应用,但研究人员已经意识到,遗传算法采用简单的、固定不变的进化策略对复杂应用场合的效果并不理想,传统的遗传算法逐渐暴露出一些缺点。所以,为了提高遗传算法的性能,使其更好地应用于实际问题的解决中,研究者们开始对基本遗传算法进行改进,通过不同的遗传基因表达方式,不同的交叉和变异算子的选择,特殊算子的引用,以及不同的再生和选择方法,产生了以基本遗传算法为核心的各种算法。遗传算法的这些扩展和改进给一般问题特别是工业工程中的难以求解的优化问题带来了新的希望和方向。

1.4 多目标遗传算法的发展

由于多目标优化问题在科学和工程实践中普遍存在,但又缺少确实有效的解决方法,研究人员把目光投向了具有多方向和全局搜索特点的遗传算法。遗传算法的这一基本特点可以确保带有潜在解的种群能够一代一代地维持下来,这种从种群到种群的方法对于搜索 Pareto 解非常有益,因此,利用遗传算法解决多目标优化问题极具研究意义。于是,遗传算法应用于单目标问题之后的 20 多年以后,多目标遗传算法逐渐成为研究热点。

经典的多目标优化方法中,通常把多个目标函数整合成单目标,将多目标优化问题转变为单目标优化问题,然后采用单目标的优化技术求解。但这些方法存在以下几个缺点:

- a) 只能得到一个解。然而,在实际决策中决策者通常需要多种可供选择的方案。
- b) 多个目标函数之间量纲不同,难以统一。为了避免其中的一个目标函数支配其他目标函数,精确的给出所有目标函数的标量信息,就必须有每一个目标的全局先验知识,计算量巨大,很难实现。
- c) 加权值的分配带有较强的主观性。由于是人为规定各个目标函数的权值,因此带有很大的主观性。
- d) 加权的目标函数之间通过决策变量相互制约,拓扑结构十分复杂。
- e) 最终优化目标仅为各目标之和,各目标的优化进度不可操作[9]。

同时,这种方法一旦可行,也有它的优点,即它无需和决策者做任何交互就可以得到最优解或近优解。早期出现的线性加权法和变权重加权法都属于这一类方法。线性加权法是受 Kuhn 和 Tucker 的数值优化方法的启发而演变来的^[29],它给每个目标函数赋予不同的权重因子,对所有的目标函数进行整合。线性加权法简单、有效、实用,对它的研究和应用有很多。

事实上,在多目标优化问题中首次应用遗传算法的思想可以追溯到 1967 年, Rosenberg 在他的研究中提出了在模拟单细胞有机物的化学遗传特性中采用多属性研究方法^[30],也提到了可以考虑用遗传的搜索算法求解多目标优化问题。虽然

他最终只应用了单一属性方法, 他的研究却开创了遗传算法应用于多目标优化领域的研究。

直到 90 年代, 遗传算法才真正地应用于多目标优化领域。这些多目标遗传算法通常分为两类, 一类是采用 Pareto 机制的多目标演化算法, 也是目前的研究热点, 一类是非 Pareto 机制的多目标演化算法。

1985 年, 为了克服整合方法的缺点, David Schaffer 在 Grefenstette 的单目标优化程序 GENESIS 的基础上提出了基于向量评估的遗传算法(VEGA)。这是第一个多目标演化算法(MOEA), 但并不是基于 Pareto 最优概念的演化算法。虽然 VEGA 本质上与传统加权法一样不适合求解折衷目标具有凹面的多目标问题, 但是与普通固定加权和算法相比, VEGA 对不同物种能够维持比较多的种群代数, 因而增强了种群的多样性^[9]。然而, 在经过一定代数的进化以后, VEGA 通常会偏好收敛于某一区域的解, 而不是均匀的收敛于整个 Pareto 域, 因此只能求的局部意义上的非支配解。

1993 年, Fonseca 和 Fleming 提出了多目标优化算法(MOGA)^[17,31-33], 其主要思想是每个个体排序的序号由当前种群中支配它的个体的数量决定, 采用共享函数和小生境技术实现种群的多样化。MOGA 易于执行且效率较高, 因而在多目标优化领域得到了广泛的应用。如: 1995 年, 用于解决喷气发动机的多变量控制系统优化问题^[34]; 1997 年, Todd 和 Sen 采用改进的 MOGA 解决了大规模组合优化问题——集装箱布局(containership layouts)问题等等^[35]。然而, MOGA 也存在缺点: 小生境的大小对算法的影响较大。

1993 年, Horn 和 Nafpliotis 提出了一种基于 Pareto 支配定义的锦标赛选择机制方法——小组决胜遗传算法(NPGA)。算法从种群中随机选取多个个体(一般为 10 个)参与, 共同决定支配关系。当两个个体之间没有明显的支配或被支配关系时, 称作 tie, 通过适应度共享来决定锦标赛选择的结果^[2,36]。因为该算法的非劣最优解的选择是基于种群的部分而非全体, 因此其优点是能很快找到一些好的非劣最优解域, 并能维持一个较长的种群更新期, 缺点是除了设置共享参数外还需要选择一个适当的锦标赛规模, 限制了该算法的实际应用效果^[9]。

1995 年, 由 Srinivas 和 Deb 提出了非支配排序遗传算法 NSGA

(Non-dominated Sorting Genetic Algorithms)。算法是根据个体之间的支配与非支配关系分层实现的^[7]。NSGA 的优点是优化目标个数任选, 非劣最优解分布均匀, 并允许存在多个不同的等价解; 缺点是算法计算复杂度较高, 缺乏精英策略, 并对共享参数 σ_{share} 的依赖性较大^[9]。NSGA 在现实问题的求解中也得到了广泛的应用, Vedarajan 等人于 1997 年采用 NSGA 进行投资利润的最优化^[37]; Michielssen 和 Weile 于 1995 年采用 NSGA 设计电磁系统。

1999 年, Eckart Zitzler 提出了 Strength Pareto Evolutionary Algorithm, 简称 SPEA^[38], 在他的论文中解决了计算任务在不同体系结构的系统上的调度问题, 并和其他算法的解进行了详尽的比较。

2000 年, Deb 等人针对 NSGA 存在的缺点, 提出了改进算法 NSGA-II, 降低了算法的计算复杂度, 引入精英策略, 并不再需要指定共享参数 σ_{share} 。NSGA-II 保持了 NSGA 具有的优点, 改进了不足, 在算法应用上取得了极好的效果。

在这些算法当中, 多目标遗传算法(MOGA), 非支配排序遗传算法(NSGA)和小组决胜遗传算法(NPGA), 已成为采用 Pareto 机制的多目标演化算法研究工作的奠基石^[9]。此外, 还存在一些非 Pareto 机制的多目标演化算法, 如多性别遗传算法、目标向量方法、权重极小极大法、变化权重的极小极大法, 无后代遗传算法等等。

随着多目标演化算法越来越多的应用于各个领域, 对它的研究也将直接受到实际工程问题的推动。目前, 多目标优化技术仍面临着许多困难, 存在很多有待研究的问题, 如非劣最优解域收敛性分析、适应值赋值方法、种群更新终止条件及其稳定性分析, 以及实际多目标优化问题的演化求解等, 这些都是目前的研究热点^[9]。

1.5 非支配排序遗传算法概述

非支配排序遗传算法 NSGA (Non-dominated Sorting Genetic Algorithms) 是由 Srinivas 和 Deb 于 1995 年提出的^[7]。这是一种基于 Pareto 最优概念的遗传算法, 它是众多的多目标优化遗传算法中体现 Goldberg 思想最直接的方法。该算

法就是在基本遗传算法的基础上,对选择再生方法进行改进:将每个个体按照它们的支配与非支配关系进行分层,再做选择操作,从而使得该算法在多目标优化方面得到非常满意的结果。

Zitzler 与 Thiele 等^[39]曾对 NSGA、NPGA、VEGA,与 Hajela 和 Lin 的加权向量算法以及纯随机搜索算法,作了系统的定量实验比较。通过采用多背包问题扩展形式的九种不同规模设置作为数值型多目标测试问题,经过全面的比较分析,得到比较结果:NSGA 的性能最优,其次是 VEGA,而 NPGA 与 Hajela 和 Lin 的加权向量算法的全部实验结果不分上下^[9]。虽然这种比较结果不能无限制的外推,但是它在一定程度上说明了 NSGA 具有一定的优越性。因此,对该算法进行理论研究和应用研究具有十分重要的意义。

虽然 NSGA 与其它多目标优化遗传算法比较具有一定的优越性:优化目标个数任选,非劣最优解分布均匀,并允许存在多个不同的等价解。但它仍存在一些问题^[8]: a) 计算复杂度较高,当种群较大时,计算相当耗时; b) 没有精英策略,精英策略可以加速算法的执行速度,而且也能在一定程度上确保已经找到的满意解不被丢失; c) 需要指定共享半径 σ_{share} 。

2000 年,他们又提出 NSGA 的改进算法——带精英策略的非支配排序遗传算法(NSGA-II)^[8]。提出了快速非支配排序法,以降低算法的计算复杂度。采用拥挤度和拥挤度比较算子,代替需要指定共享半径的适应度共享策略,并在快速排序后的同级比较中作为胜出标准,使准 Pareto 域中的个体能扩展到整个 Pareto 域,并均匀分布,保持了种群的多样性。另外,还引入精英策略,扩大了采样空间。将父代种群与其产生的子代种群组合,共同竞争产生下一代种群,有利于保持父代中的优良个体进入下一代,并通过对种群中所有个体的分级存放,使得最佳个体不会丢失,迅速提高种群水平。

NSGA-II 改进了原来算法的不足之处,提高了算法的运算速度和鲁棒性,并保证了非劣最优解的均匀分布。因而,NSGA-II 比 NSGA 更具优越性,其在近几年被应用于众多行业并得到了很好的效果^[40,41],说明了它在众多的多目标优化遗传算法中更具应用价值。所以,扩大 NSGA 及其 NSGA-II 的应用领域,进行应用研究,具有很重要的现实意义。

1.6 本文所作的工作

1.6.1 基本思想

目前多目标优化遗传算法的研究主要有两个方向：理论研究和应用研究。本文通过对非支配排序的遗传算法及其改进算法的研究和学习，大胆地尝试将此算法应用于变结构控制系统。针对多输入控制系统输入参数不易确定，控制性能难以满足的特点，利用 NSGA-II 算法在处理多目标优化问题上的优势，以系统输出超调量、动态响应时间和抖振的最小化为算法的目标函数，对原本需要人为设置的变结构控制参数进行优化。

同时，本文还尝试将 NSGA-II 与支持向量机回归方法相结合发展一种新的解决函数逼近问题的方法。通过引入 SVM 中的结构风险最小化准则，核函数的计算复杂度，及函数逼近的偏差的范数作为多目标优化问题的目标函数，实现了核函数形式和其他重要参数的自动选取。仿真结果表明，该方法比传统的 SVM 方法更具优越性。

本文所作的研究开拓了 NSGA-II 的应用领域，具有现实意义。

1.6.2 内容安排

总体上，本文分为六章。

第一章概述了遗传算法的发展历史和现状，讨论了目前用于解决多目标优化问题的一些方法及存在的一些问题，并简要介绍了非支配排序遗传算法的发展和应用情况，对本文所作的工作做了大概的交代。

第二章系统的介绍了遗传算法的一般流程、基本原理、编码方式、适应度函数以及 3 个基本算子。

第三章详细介绍了非支配排序遗传算法 (NSGA) 的基本原理，论述了该算法存在的缺陷，并介绍了它的改进算法——带精英策略的非支配排序遗传算法 (NSGA-II)。

第四章针对变结构控制中多输入控制系统输入参数不易确定，控制性能难以

满足的特点，利用改进的非支配排序遗传算法（NSGA-II）处理多目标优化问题的优势，对变结构控制参数进行优化。文中进行了仿真试验，试验结果表明：该方法有效的抑制了系统的超调量，缩短了动态响应时间，降低了抖振，满足了控制性能的要求。

第五章提出了一种基于 NSGA-II 与 SVM 的函数逼近方法。通过引入 SVM 中的结构风险最小化准则，核函数的计算复杂度，及函数逼近的偏差的范数作为多目标优化问题的目标函数，实现了核函数形式和其他重要参数的自动选取。文中通过仿真试验，证明了该方法比传统的 SVM 方法更有效。

第六章对本文所作的工作进行了总结，并对非支配排序遗传算法及其改进算法今后的研究方向进行了展望。

第二章 遗传算法基本原理和方法

2.1 遗传算法的基本原理

遗传算法 (Genetic Algorithms, GA) 是一类借鉴生物界自然选择和自然遗传机制的随机的搜索算法, 由 Holland 教授于 1975 年提出。

其基本思想来源于达尔文(Darum)的进化论和孟德尔(Mendel)的遗传学说。达尔文的进化论认为: 每个物种都是朝着越来越适应环境的方向进化的; 物种的每个个体都继承了其父代的基本特征, 但又不完全与之相同; 更能适应环境的个体在进化过程中生存下来, 其特征也被保留下来, 体现了适者生存的原理。与此相应, 孟德尔的遗传学认为: 遗传是以基因的形式包含在染色体中的; 每个基因有自己特定的位置并控制着个体的某个特性, 而由这些基因控制的特性使个体对环境具有一定的适应性; 通过基因杂交和基因突变可能产生对环境适应性更强的后代, 通过优胜劣汰的自然选择, 适应值高的基因结构被保存下来^[42]。

遗传算法正是受这两种学说的启发而发展起来的。通过计算机编程, 将待求问题表示成二进制码串或数码串 (也称作个体), 并将产生的一群串 (称作种群) 置于问题的求解环境中, 根据适者生存的原则, 从种群中选出适应环境的串进行复制, 且通过交叉、变异两种基因操作产生新一代更适应环境的种群, 经过这样一代一代地不断进化, 收敛到一个最适应环境的串上, 便求得了问题的最优解。

2.2 遗传算法的一般流程

Holland 教授提出的遗传算法, 现在一般被称为简单遗传算法 (Simple Genetic Algorithms, 简称 SGA) 或基本遗传算法, 其基本流程如图 2.1 所示^[10]。具体求解步骤如下:

- 1) 参数编码: 遗传算法一般不直接处理问题空间的参数, 因此在算法开始进行

之前，首先要选择合适的编码方式对待优化的参数进行编码。通常采用二进制编码，将参数转换成为 0 和 1 组成的数字串。

- 2) 产生初始种群：随机地产生一个由 n 个个体组成的种群，该种群代表一些可能解的集合。遗传算法的任务是种群出发，模拟生物进化的过程进行优胜劣汰，最后得出满足优化要求的种群和个体。
- 3) 设计适应度函数：把问题的目标函数转换成合适的适应度函数，并根据适应度函数计算种群中的每个个体的适应度，为种群进化的选择提供依据。
- 4) 优化准则：也可称作终止条件，是用来判断算法是否可以终止的标准。可以设定进化的最大代数，当进化到最大代数时，算法终止运行。也可以设定期望的适应度函数值，只有当种群中存在个体能达到期望值时，算法才可以结束。通常情况下，这两种方法同时作为优化准则使用。
- 5) 选择(复制)操作：按一定概率从群体中选择个体，作为双亲用于繁殖后代，产生新的个体。在此操作中，适应于生存环境的优良个体将有更多的机会繁殖后代，这使得优良特性能够遗传到下一代。
- 6) 交叉操作：随机地选择用于繁殖的每一对个体的同一基因位，将其染色体在此基因位断开并相互交换。
- 7) 变异操作：以一定的概率从群体中选择若干个个体。对于选中的个体，随机选择某一位进行取反操作。

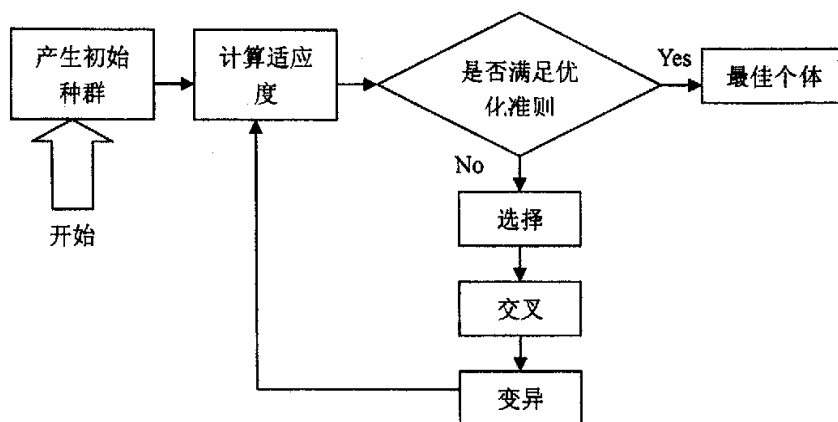


图 2.1 基本遗传算法流程图

对产生的新一代群体重新进行评价、选择、杂交和变异。一代一代循环往复，使种群中最优个体的适应度和平均适应度不断提高，直至最优个体的适应度满足优化准则或最优个体的适应度和平均适应度不再提高，则迭代过程收敛，算法结束。遗传算法的选择和交叉算子赋予了它强有力的搜索能力，变异算子则使算法能搜索到问题解空间的每一个点，以确保算法能达到全局最优。

遗传算法提供了一种求解复杂系统优化问题的通用框架，它不依赖于问题的领域和种类。对一个需要进行优化计算的实际应用问题，一般可以按照上述步骤来构造求解该问题的遗传算法。由上述步骤可以看出，构造遗传算法时需要考虑的两个主要问题是可行解的编码方法和遗传算子的设计，这也是设计遗传算法的两个关键步骤。

2.3 遗传算法的编码方式

如何进行编码是遗传算法中的关键问题。编码方式有很多种，例如：二进制编码（binary encoding）、格雷码编码（Gray encoding）、实数编码（real-number encoding）、整数或字母排列编码（literal permutation encoding）等。

最初，Holland^[12] 采用了二进制字符串进行编码，其方式简单，易于理解。但是，由于 Hamming 悬崖的存在，二进制编码对于函数优化问题存在严重缺陷。Hamming 悬崖指的是表现型空间中距离很小的个体对可能有很大的 Hamming 距离。例如，个体对 011111 和 100000 属于表现型空间中的相邻点（最小 Euclidean 距离点），但它们却在基因型空间具有最大的 Hamming 距离，即从 011111 变到 100000 需要同时改变个体的所有位。通过交叉和变异实现翻越 Hamming 悬崖的可能性非常小^[43]。因此，采用二进制编码时可能出现这样的情况：在已经找到的可行解的附近存在最优解的点，但由于它们之间的 Hamming 距离太大，通过交叉变异仍无法找到该点，因而最优解在进化过程中被丢失了。

为了解决这一问题，格雷码编码应运而生。格雷码有这样一个特点：任意两个整数的差是这两个整数所对应的格雷码之间的 Hamming 距离。二进制码与格雷码的比较，如表 2.1 所示^[44]。

表 2.1 二进制码与格雷码的比较

十进制码	0	1	2	3	4	5	6	7
二进制码	0000	0001	0010	0011	0100	0101	0110	0111
格雷码	0000	0001	0011	0010	0110	0111	0101	0100
十进制码	8	9	10	11	12	13	14	15
二进制码	1000	1001	1010	1011	1100	1101	1110	1111
格雷码	1100	1101	1111	1110	1010	1011	1001	1000

二进制码和格雷码之间可以相互转换。假设一个二进制码为 $B = b_m b_{m-1} \cdots b_2 b_1$ ，其对应的格雷码为 $G = g_m g_{m-1} \cdots g_2 g_1$ 。由二进制码到格雷码的转换公式为^[44]：

$$\begin{cases} g_m = b_m \\ g_i = b_{i+1} \oplus b_i, i = m-1, m-2, \dots, 1 \end{cases} \quad (2.1)$$

由格雷码到二进制码的转换公式为：

$$\begin{cases} b_m = g_m \\ b_i = b_{i+1} \oplus g_i, i = m-1, m-2, \dots, 1 \end{cases} \quad (2.2)$$

虽然格雷码相对于二进制码而言，提高了遗传算法的局部搜索能力，但是对于工业工程领域里的许多问题来说，采用二进制编码或格雷码编码都很难满足它们需要的精度。为了解决这个问题，人们提出了实数编码，因为实数编码在基因型空间中的拓扑结构与表现型空间中的拓扑结构是一致的，因此个体的基因值可以用某一范围内的实数表示。实数编码的优点是：可以表示较大范围内的数，表示精度很高，便于处理多变量的优化。

整数和字母排列编码，或称为序号编码。对于组合优化问题，如旅行商问题、背包问题，其关键是要寻找满足约束项目的最佳排列或组合，因此字母排列编码对于这类问题是最有效的方法。

2.4 适应度函数及其尺度变换

遗传算法在进化搜索中基本不利用外部信息, 仅以适应度函数作为区分种群个体好坏的标准。一般而言, 适应度函数是由目标函数变换而来的。因此, 适应度函数选择的好坏直接影响算法的优劣, 引入适应值调节和资源共享策略可以加快收敛速度和跳出局部最优点。对适应值进行调节就是通过变换来改变原适应值的比例关系, 常用的比例关系有线性变换, 幂函数变换和指数变换等^[42]。

2.4.1 适应度函数^[44]

个体适应度函数 (fitness function) 通常是目标函数的一个变换。若目标函数为最小化问题, 则

$$Fit(f(x)) = \begin{cases} c_{\max} - f(x), & f(x) < c_{\max} \\ 0, & \text{else} \end{cases} \quad (2.3)$$

其中, $f(x)$ 为目标函数, c_{\max} 为目标函数的最大值估计, $Fit(f(x))$ 为适应值。

若目标函数为最大化问题, 则

$$Fit(f(x)) = \begin{cases} f(x) - c_{\min}, & f(x) > c_{\min} \\ 0, & \text{else} \end{cases} \quad (2.4)$$

其中, $f(x)$ 为目标函数, c_{\min} 为目标函数的最小值估计, $Fit(f(x))$ 为适应值。

2.4.2 适应度函数的尺度变换

在遗传进化的初期, 通常会产生一些竞争力极强的超常个体, 它们控制了选择过程, 影响了算法的全局优化性能。因此, 在进化初期必须避免某一个个体适应度过大, 通过对适应度进行尺度变换 (fitness scaling), 改变原适应值的比例关系。常用的尺度变换方法有线性变换法、幂函数变换法和指数变换法。

线性变换法中, 假设原来的适应度函数为 f , 变换后的适应度函数为 f' , 则线性变换表示为:

$$f' = a * f + b \quad (2.5)$$

其中, a 和 b 分别表示变换系数和常数项。

根据以下两个条件确定系数 a 和 b 的值: 原适应度的平均值等于变换后的适应度平均值; 变换后适应度的最大值等于原适应度平均值的指定倍数 c , 以控制适应度最大的个体在下一代中的复制数。则 a 、 b 可以表示为:

$$a = \frac{(c-1)f_{avg}}{f_{max} - f_{avg}}, b = \frac{(f_{max} - cf_{avg})f_{avg}}{f_{max} - f_{avg}} \quad (2.6)$$

其中, f_{avg} 表示平均适应度, f_{max} 表示最大适应度, c 表示控制参数。如图 2.2 所示^[44]。

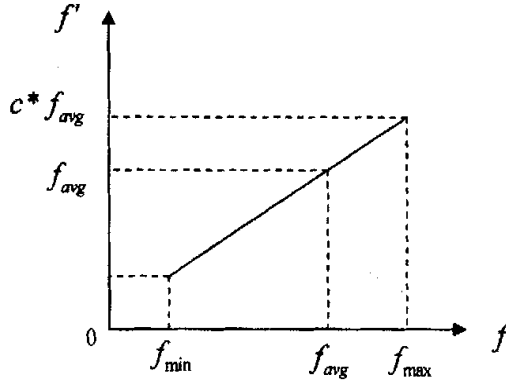


图 2.2 适应度函数的线性变换

线性变换法通过改变适应度之间的差距, 保持了种群的多样性, 且计算简单, 易于实现。但当种群内某些个体适应度远远低于平均值时, 则变换后的适应值可能为负。为了避免这种情况, 可以进行另一种变换:

$$a = \frac{f_{avg}}{f_{avg} - f_{min}}, b = \frac{-f_{avg}f_{min}}{f_{avg} - f_{min}} \quad (2.7)$$

其中, f_{min} 表示最小适应度, 其与同上。如图 2.3 所示^[44]。

除了线性变换法, 另外还有两种常用的变换方法: 幂函数变换法, 其变换公式为: $f' = f^k$; 指数变换法, 其变换公式为: $f' = e^{-af}$ 。

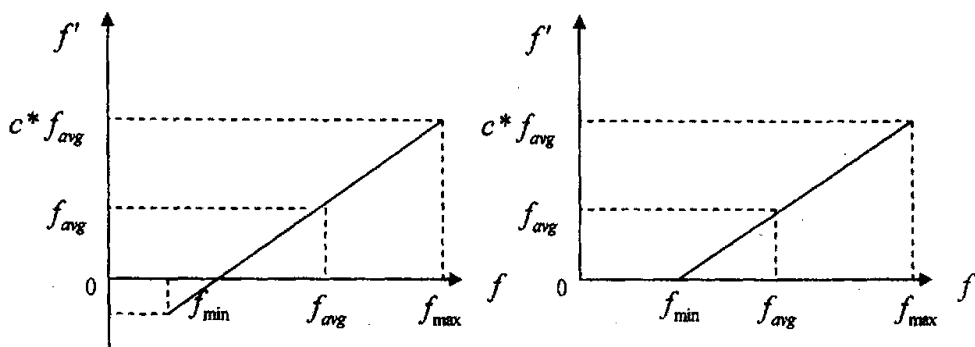


图 2.3 线性变换时适应度出现负值的情况及相应的调整

2.5 遗传算法的基本算子

2.5.1 选择算子

选择操作体现了自然界中适者生存的思想，它提供了遗传算法的驱动力，其任务就是将遗传搜索引导到搜索空间中有前途的区域中。在进化过程中，驱动力不能太大，否则遗传搜索将过早地终止；也不能太小，否则进化过程将进行得十分缓慢。通常情况下，在遗传算法进行的早期需采用较小的选择压力，以对搜索空间进行广度搜索，而在晚期则推荐采用较大的选择压力，以限制搜索空间。在过去的 20 年中提出、验证并比较了许多选择方法。通常采用的选择方法如下^[43]：

(1) 轮盘赌选择 (roulette wheel selection) 是由 Holland 提出的最知名的选择方式^[12]，其特点是随机采样。根据每个染色体适应值的比例，确定该个体的选择概率或生存概率。建立一个轮盘赌模型来表示这些概率，通过旋转轮盘选出个体并放入新种群中，旋转的次数等于种群的大小。

(2) $(\mu + \lambda)$ 选择 ($(\mu + \lambda)$ selection)，与轮盘赌选择不同，它是确定性选择过程。该方法从父代和子代中选取最好的个体，但不允许从种群中选取相同染色体，因此许多研究人员倾向于采用此方法来处理组合优化问题。

(3) 锦标赛选择 (tournament selection) 是由 Goldberg、Korb 和 Deb 提出的。该方法随机选择一些个体，并从中选出最好的进行繁殖，因此同时包含了随机性和确定性的特征。

(4) 稳态复制 (steady-state reproduction) 是由 Whitley 和 Syswerda 提出的。该方法用子代替换 n 个最差的父代 (n 是子代的个数), 可以看作另一种形式的确定性选择。

(5) 比例变换与排序选择。比例变换方法将原始目标函数值映射为正实数, 根据这些正实数计算每个个体的生存概率。这种方法维持了个体相对适应值比例间的合理差异, 并在算法早期通过抑制某些超级个体过快收敛来达到了限制竞争的目的。但是其参数选取随着问题的不同而不同。Backer 提出的排序选择方法, 通过对种群进行从好到差的排序, 并根据每个染色体的排序 (而不是原始适应值) 来确定其选择概率。该方法得到了类似于比例变换的效果, 同时又避免了比例变换参数选取的问题。

(6) 共享 (sharing) 方法。针对多峰函数优化问题, Goldberg 和 Richardson 提出了共享方法。共享函数是根据个体在某个距离内与其它个体相邻程度来决定该个体适应值下降多少的。由于适应值的下降, 在拥挤的峰周围的个体的复制率受到抑制, 而其他个体则容易产生后代。共享方法维持了种群的多样性。

2.5.2 交叉算子^[44]

交叉是遗传算法中产生新个体的主要方式, 体现了自然界中信息交换的思想。交叉操作通过把两个父代个体的部分结构加以替换重组, 从而产生新的个体。根据编码方法的不同, 具体的交叉方式也不同, 二进制编码以及格雷码编码一般称为交叉, 实数编码称为重组, 整数或字母排列编码称为重排列。

(1) 交叉方式

单点交叉: 又称为简单交叉。在个体串中随机选择一个交叉点, 交换两个个体在此点后面的部分, 从而产生了两个新个体

两点交叉: 与单点交叉类似, 只是交叉点为两个。

多点交叉: 是单点交叉和两点交叉的推广, 交叉点为随机选择的多个点, 从头向尾, 每遇到一个点就交换一下此点后的部分。

均匀交叉: 对所有点随机选择 0 或 1, 交换两个父个体中标 1 的点的基因。

受限交叉：在父个体不同的位置中随机选择一点，交换此点后面的部分。

举例说明，如图 2.4 所示：

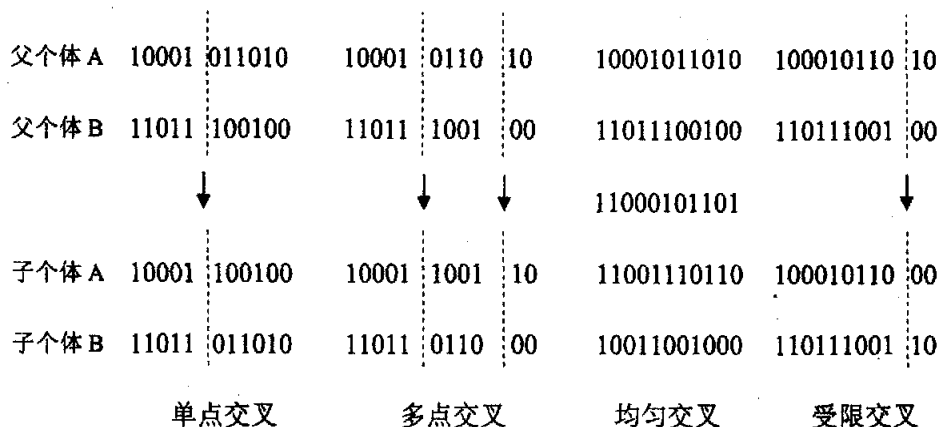


图 2.4 二进制码的几种主要交叉方法

(2) 重组方式

离散重组：对每一个位置，随机从父个体中取对应位置的值。

中间重组：子个体 = 父个体 A + α * (父个体 B - 父个体 A)，其中 α 为比例因子，由 $[-d, 1+d]$ 上均匀分布的随机数产生。中间重组， $d = 0$ 。

线性重组：类似于中间重组，对所有变量只有一个 α 值。

扩展线性重组：为线性重组增加了一个随机的扰动：大扰动的概率较小，小扰动的概率较大。

(3) 重排列方式

部分匹配交叉：在两个父个体上统一地随机选取两个交叉点，交叉点之间的部分定义为匹配段。互换匹配段中两个个体对应的部分，实现交叉操作。

循环交叉：利用一个交替映射找到一个环，然后交换所有不在环上位置的串。

次序交叉：次序交叉按匹配段中对方的基因，将本个体中相应的基因映射成空穴；利用一个从第二个交叉点开始的滑动来填充空穴；最后匹配段交叉。

2.5.3 变异算子^[44]

变异是指子代基因按小概率扰动产生的变化，它模拟了生物进化过程中的偶然基因突变的现象。按个体编码方式分为二进制变异、实数变异以及序号变异。

(1) 二进制变异

二进制变异法：随机产生一行与编码长度相同的(0, 1)之间的随机数，如果这个随机数小于指定的 P_m ，就把相应位置上的个体从 1 变成 0，或从 0 变成 1。

例如：

	↓
发生变异的位置	0000010000
变异前个体	1001011010
变异后个体	1001001010

(2) 实数变异

一致变异法：其变异的最大幅度不随进化的代数发生变化。

非一致变异法：其变异的最大幅度随着进化代数的增加而逐渐减小。

多变量非一致变异法：除了变异最大幅度随进化的代数增加逐渐减小之外，其变异方向也是随机的。

(3) 序号变异

交换变异：随机选择两点，将两点的值交换。

倒序变异：随机选择两点，将两点之间的基因值颠倒顺序。

2.6 遗传算法的特点及研究内容

2.6.1 遗传算法的特点

遗传算法是一类可用于复杂系统优化的稳健的搜索算法，它具有以下几个特点：

- (1) 遗传算法的处理对象不是参数本身，而是对参数集进行了编码的个体。这种编码操作使得遗传算法可直接对结构对象进行操作，具有良好的可操作性和广泛的应用领域。
- (2) 遗传算法直接利用适应值作为搜索信息，基本上不需要搜索空间的知识或其它辅助信息。其搜索过程既不受优化函数的连续性约束，又避开了函数求导，因而适用于任何大规模、高度非线性的、不连续多峰函数的优化以及无解析表达式的目标函数的优化，具有很强的通用性；而且可以把搜索范围集中到适应度较高的部分搜索空间中，从而提高了搜索效率。
- (3) 遗传算法采用多点搜索或者说是群体搜索，具有很高的隐含并行性。这一特点使得遗传算法具有较好的全局搜索性能，减少了陷于局部最优解的风险。
- (4) 遗传算法是一种自适应搜索技术，其选择、交叉、变异等操作都是以一种概率方式来进行，从而增加了搜索过程的灵活性，同时能以很大的概率收敛于最优解，具有较好的全局优化求解能力。
- (5) 遗传算法的基本思想简单，运行方式和实现步骤规范，便于具体使用。

2.6.2 研究内容

遗传算法的研究主要包括两个方面：算法应用研究和算法理论研究。虽然遗传算法目前已经被广泛的应用于各个领域，但是科学工作者们对它的理论研究从未停止。其理论研究的主要内容包括：分析遗传算法的编码方法、适应值函数、3个遗传算子、遗传算法的参数选择、全局收敛性和搜索效率的数学基础、欺骗问题、收敛性分析、并行计算、遗传算法的局部改进及混合遗传算法等^[45]。

其中，以研究遗传算法的改进发展最快。通过采用不同的遗传基因表达方式，选择不同的交叉和变异算子，引入特殊算子，以及不同的再生和选择方法，产生了以基本遗传算法为核心的各种算法。这些算法中，又以用于解决多目标优化问题的遗传算法发展最为迅速。VEGA, MOGA, NPGA, 以及NSGA等都是在基本遗传算法的基础上发展起来的，目前已被广为应用。

2.7 本章小结

本章系统的介绍了遗传算法的基本原理和一般流程，对算法的编码方式、适应度函数及其尺度变换方法和 3 个基本算子做了较详细的讨论。最后总结了遗传算法的基本特点，并对其研究内容作了简要的介绍。

第三章非支配排序遗传算法(NSGA)及 NSGA-II

3.1 多目标优化问题的数学描述

工程中经常会遇到多准则或多设计目标下的设计和决策问题, 这些目标往往是相悖的, 要找到满足这些目标的最佳设计方案, 就要解决多目标和多约束的优化问题, 即多目标优化 (Multi-Objective Optimization, MO) 问题。

通常考虑的多目标优化问题, 可以定义为在一组约束条件下, 极大化 (或极小化) 多个不同的目标函数, 其一般形式为^[7]:

$$\begin{cases} \text{Maximize / Minimize } [f_1(X), f_2(X), \dots, f_n(X)] \\ \text{s.t.} \\ g_j(X) \leq 0, j = 1, 2, \dots, J \\ h_k(X) = 0, k = 1, 2, \dots, K \end{cases} \quad (3.1)$$

其中 $X = (x_1, x_2, \dots, x_p)$ 是一个 p 维向量, $f_i(X)$ 是目标函数, $i = 1, 2, \dots, n$, $g_j(X) \leq 0$ 和 $h_k(X) = 0$ 是系统约束, $j = 1, 2, \dots, J$, $k = 1, 2, \dots, K$ 。

3.2 基本概念

3.2.1 Pareto 支配关系

对于最小化多目标问题, n 个目标分量 $f_i (i = 1, \dots, n)$ 组成的向量 $\bar{f}(\bar{X}) = (f_1(\bar{X}), f_2(\bar{X}), \dots, f_n(\bar{X}))$, 任意给定两个决策变量 $\bar{X}_u, \bar{X}_v \in U$:

当且仅当, 对于 $\forall i \in \{1, \dots, n\}$, 都有 $f_i(\bar{X}_u) < f_i(\bar{X}_v)$, 则 \bar{X}_u 支配 \bar{X}_v 。

当且仅当, 对于 $\forall i \in \{1, \dots, n\}$, 有 $f_i(\bar{X}_u) \leq f_i(\bar{X}_v)$, 且至少存在一个 $j \in \{1, \dots, n\}$, 使 $f_j(\bar{X}_u) < f_j(\bar{X}_v)$, 则 \bar{X}_u 弱支配 \bar{X}_v 。

当且仅当, $\exists i \in \{1, \dots, n\}$, 使 $f_i(\bar{X}_u) < f_i(\bar{X}_v)$, 同时, $\exists j \in \{1, \dots, n\}$, 使 $f_j(\bar{X}_u) > f_j(\bar{X}_v)$, 则 \bar{X}_u 与 \bar{X}_v 互不支配。

3.2.2 Pareto 最优解定义

多目标优化问题与单目标优化问题有很大差异。当只有一个目标函数时, 人们寻找最好的解, 这个解优于其他所有解, 通常是全局最大或最小, 即全局最优解。而当存在多个目标时, 由于目标之间存在冲突无法比较, 所以很难找到一个解使得所有的目标函数同时最优, 也就是说, 一个解可能对于某个目标函数是最好的, 但对于其他的目标函数却不是最好的, 甚至是最差的。因此, 对于多目标优化问题, 通常存在一个解集, 这些解之间就全体目标函数而言是无法比较优劣的, 其特点是: 无法在改进任何目标函数的同时不削弱至少一个其他目标函数^[9]。这种解称作非支配解 (nondominated solutions) 或 Pareto 最优解 (Pareto optimal solutions), 其定义如下:

对于最小化多目标问题, n 个目标分量 $f_i (i=1, \dots, n)$ 组成的向量 $\bar{f}(\bar{X}) = (f_1(\bar{X}), f_2(\bar{X}), \dots, f_n(\bar{X}))$, $\bar{X}_u \in U$ 为决策变量, 若 \bar{X}_u 为 Pareto 最优解, 则需满足^[10]:

当且仅当, 不存在决策变量 $\bar{X}_v \in U$, $v = f(\bar{X}_v) = (v_1, \dots, v_n)$ 支配 $u = f(\bar{X}_u) = (u_1, \dots, u_n)$ 即不存在 $\bar{X}_v \in U$ 使得下式成立:

$$\forall i \in \{1, \dots, n\}, v_i \leq u_i \wedge \exists i \in \{1, \dots, n\} | v_i < u_i \quad (3.2)$$

Pareto 最优概念是由法国经济学家 Pareto 于 1896 年提出的。它是建立在集合论基础上对多目标解的一种向量评估方式。传统的数学规划法与模拟退火算法是以单点搜索为特征的串行算法, 无法利用 Pareto 最优概念对解进行评估。因此, Pareto 最优概念提出百余年仍无传统算法意义上的相关算法。基于种群操作的演化算法可以隐并行地搜索解空间的多个解, 并能利用不同解之间的相似性提高其并行求解问题效率, 与 Pareto 最优概念相结合, 则有可能产生真正基于 Pareto 最优概念的多目标优化的演化算法, 实现对非劣最优解集的搜索^[9]。

3.3 非支配排序遗传算法 (NSGA)

1995 年, Srinivas 和 Deb 提出了非支配排序遗传算法 (Non-dominated Sorting Genetic Algorithms, NSGA) [7]。这是一种基于 Pareto 最优概念的遗传算法, 是众多的多目标优化遗传算法中体现 Goldberg 思想最直接的方法。

3.3.1 基本原理

NSGA 与简单的遗传算法的主要区别在于: 该算法在选择算子执行之前根据个体之间的支配关系进行了分层。其选择算子、交叉算子和变异算子与简单遗传算法没有区别[7]。

在选择操作执行之前, 种群根据个体之间的支配与非支配关系进行排序。首先, 找出该种群中的所有非支配个体, 并赋予他们一个共享的虚拟适应度值, 得到第一个非支配最优层; 然后, 忽略这组已分层的个体, 对种群中的其它个体继续按照支配与非支配关系进行分层, 并赋予它们一个新的虚拟适应度值, 该值要小于上一层的值, 对剩下的个体继续上述操作, 直到种群中的所有个体都被分层。

算法根据适应度共享对虚拟适应值重新指定, 比如指定第一层个体的虚拟适应值为 1, 第二层个体的虚拟适应值应该相应减少, 可取为 0.9, 依此类推。这样, 可使虚拟适应值规范化, 保持优良个体适应度的优势, 以获得更多的复制机会, 同时也维持了种群的多样性。由于采用了适应度共享, 对于在共享半径 σ_{share}

内的个体其适应度相应减少为 $f(x) = \frac{f'(x)}{\sum_{y \in P} s(d(x, y))}$ 。式中: x, y 为个体; $f(x)$

为个体 x 共享后的适应度值; $f'(x)$ 为个体 x 共享前的适应度值; s 为共享函数; d 为距离函数; P 为种群; α 为常数[44]。

共享函数 s 表示个体 x 和小生境群体中其他个体的关系[17],

$$s(d(x, y)) = \begin{cases} 1 - \left(\frac{d(x, y)}{\sigma_{share}}\right)^\alpha & \text{若 } d(x, y) < \sigma_{share} \\ 0 & \text{其他} \end{cases} \quad (3.3)$$

其中 σ_{share} 为共享半径。根据距离函数 $d(x, y)$ 的定义不同, 共享方式可分为: 编

码空间中的适应度共享，决策变量空间中的适应度共享，以及目标函数空间中的适应度共享。

3.3.2 一般流程

NSGA 采用的非支配分层方法，可以使好的个体有更大的机会遗传到下一代；适应度共享策略则使得准 Pareto 面上的个体均匀分布，保持了群体多样性，克服了超级个体的过度繁殖，防止了早熟收敛。算法流程如图 3.1 所示^[7]。

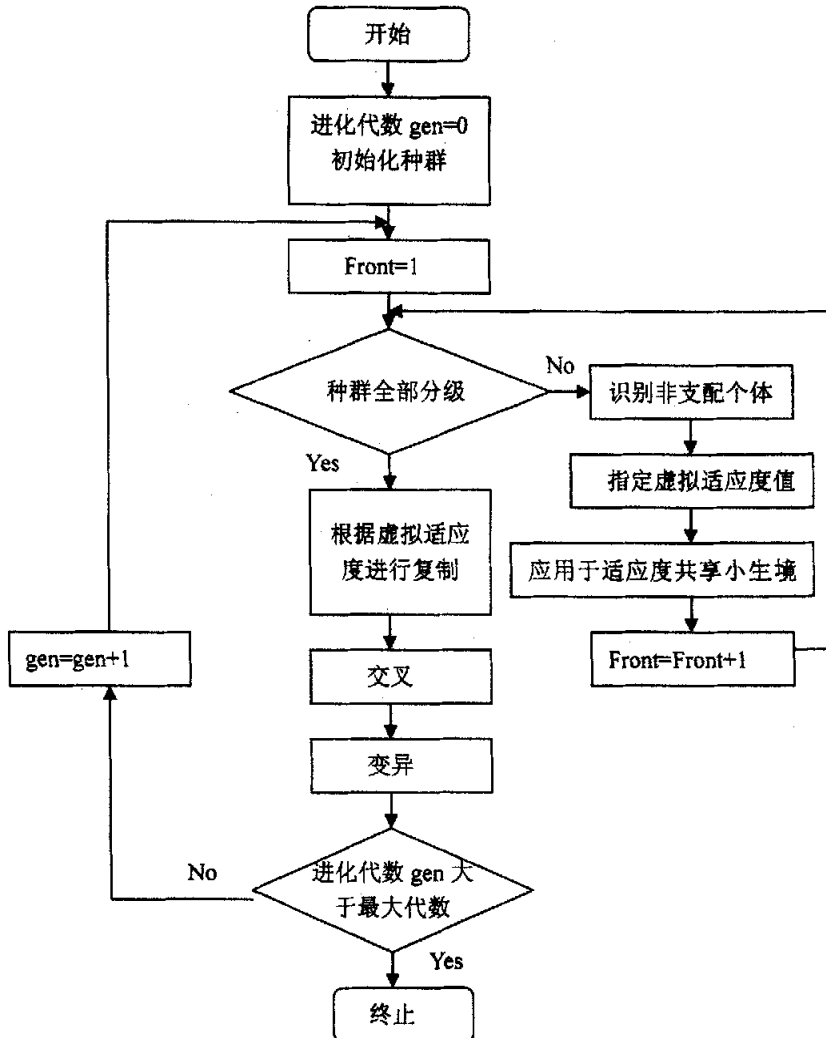


图 3.1 NSGA 工作原理流程图

3.4 带精英策略的非支配排序的遗传算法 (NSGA-II) [8]

非支配排序遗传算法 (NSGA) 在许多问题上得到了应用, 但 NSGA 仍存在一些问題:

- a) 计算复杂度较高, 为 $O(mN^3)$ (m 为目标函数个数, N 为种群大小), 所以当种群较大时, 计算相当耗时。
- b) 没有精英策略: 精英策略可以加速算法的执行速度, 而且也能在一定程度上确保已经找到的满意解不被丢失。
- c) 需要指定共享半径 σ_{share} 。

2000 年, Deb 又提出 NSGA 的改进算法——带精英策略的非支配排序遗传算法 (NSGA-II)。NSGA-II 针对以上的缺陷通过以下三个方面进行了改进:

- a) 提出了快速非支配排序法, 降低了算法的计算复杂度。由原来的 $O(mN^3)$ 降到 $O(mN^2)$, 其中, m 为目标函数个数, N 为种群大小。
- b) 提出了拥挤度和拥挤度比较算子, 代替了需要指定共享半径的适应度共享策略, 并在快速排序后的同级比较中作为胜出标准, 使准 Pareto 域中的个体能扩展到整个 Pareto 域, 并均匀分布, 保持了种群的多样性。
- c) 引入精英策略, 扩大采样空间。将父代种群与其产生的子代种群组合, 共同竞争产生下一代种群, 有利于保持父代中的优良个体进入下一代, 并通过对种群中所有个体的分层存放, 使得最佳个体不会丢失, 迅速提高种群水平。

3.4.1 快速非支配排序方法

NSGA-II 对第一代算法中非支配排序方法进行了改进: 对于每个个体 i 都有以下两个参数 n_i 和 S_i , n_i 为在种群中支配个体 i 的解个体的数量, S_i 为被个体 i 所支配的解个体的集合。首先, 找到种群中所有 $n_i = 0$ 的个体, 将它们存入当

前集合 F_1 ，然后对于当前集合 F_1 中的每个个体 j ，考察它所支配的个体集 S_j ，将集合 S_j 中的每个个体 k 的 n_k 减去 1，即支配个体 k 的解个体数减 1（因为支配个体 k 的个体 j 已经存入当前集 F_1 ），如果 $n_k - 1 = 0$ 则将个体 k 存入另一个集 H 。最后，将 F_1 作为第一级非支配个体集合，并赋予该集合内个体一个相同的非支配序 i_{rank} ，然后继续对 H 作上述分级操作并赋予相应的非支配序，直到所有的个体都被分级。其计算复杂度为 $O(mN^2)$ ， m 为目标函数个数， N 为种群大小。

上述非支配排序伪代码如下^[8]：

函数 sort(P)

对每一个 $i \in P$

$S_i = \phi$; $n_i = 0$;

对每一个 $j \in P$

若 i 支配 j ， 则 $S_i = S_i \cup \{j\}$

否则 $n_i = n_i + 1$

若 $n_i = 0$ ， 则 $i_{rank} = 1$; $F_1 = F_1 \cup \{i\}$;

$p = 1$

p 为非支配层数，初始值为 1

当 $F_p \neq \phi$ 时

$H = \phi$

对每个 $i \in F_p$

对每个 $j \in S_i$

$n_j = n_j - 1$

若 $n_j = 0$ ， 则 $j_{rank} = p + 1$; $H = H \cup \{j\}$

$p = p + 1$

$$F_p = H$$

3.4.2 确定拥挤度

在原来的 NSGA 中, 我们采用共享函数以确保种群的多样性, 但这需要由决策者指定共享半径 σ_{share} 的值。为了解决这个问题, 我们提出了拥挤度概念: 在种群中的给定点的周围个体的密度, 用 i_d 表示, 它指出了在个体 i 周围包含个体 i 本身但不包含其他个体的最小的长方形, 如图 3.2 所示^[8]。

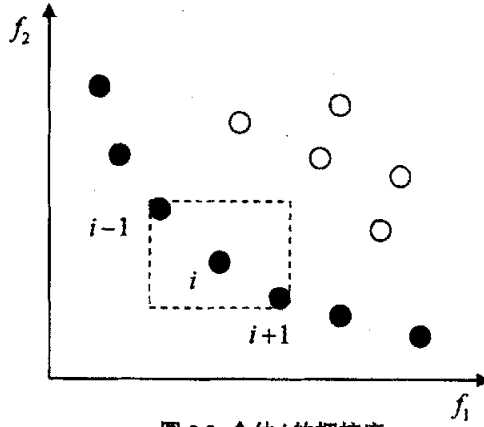


图 3.2 个体 i 的拥挤度

拥挤度的计算伪代码如下 (其中 I 为种群中的非支配集)^[8]:

函数 crowding-distance(I)

$l = |I|$ $\backslash\backslash$ 集合 I 中解个体的个数

对每个 i , $I[i]_d = 0$ $\backslash\backslash$ 每个个体 i 的拥挤度初始值都为零

对每个目标函数 m

$I = \text{sort}(I, m)$

$I[1]_d = I[l]_d = \infty$

从 $i = 2$ 到 $l - 1$ 循环

$I[i]_d = I[i]_d + (I[i+1]m - I[i-1]m) / (f_m^{\max} - f_m^{\min})$

这里, $I[i]m$ 表示集合 I 中的第 i 个个体对于第 m 个目标函数的值。 $\text{sort}(I, m)$ 是

指在目标函数 m 下对个体进行非支配排序。算法的复杂性取决于排序的复杂性，最极端情况下，当所有的个体都在一个非支配集里时，计算复杂度为 $O(mN \lg N)$ 。

3.4.3 拥挤度比较算子

从图 3.2 中我们可以看出 i_d 值较小时表示该个体周围比较拥挤。为了维持种群的多样性，我们需要一个比较拥挤度的算子以确保算法能够收敛到一个均匀分布的 Pareto 面上。

由于经过了排序和拥挤度的计算，群体中每个个体 i 都得到两个属性：非支配序 i_{rank} 和拥挤度 i_d ，则定义偏序关系为 $<_n$ ：当满足条件 $i_{rank} < j_{rank}$ ，或满足 $i_{rank} = j_{rank}$ 且 $i_d > j_d$ 时，定义 $i <_n j$ 。也就是说：如果两个个体的非支配排序不同，取排序号较小的个体（分层排序时，先被分离出来的个体）；如果两个个体在同一级，取周围较不拥挤的个体。

3.4.4 NSGA-II 算法主流程

首先，随机初始化一个父代种群 P_0 ，并将所有个体按非支配关系排序且指定一个适应度值，如：可以指定适应度值等于其非支配序 i_{rank} ，则 1 是最佳适应度值。然后，采用选择、交叉、变异算子产生下一代种群 Q_0 ，大小为 N

NSGA-II 算法主流程如下^[8]：

$$R_t = P_t \cup Q_t$$

$$F = \text{sort}(R_t)$$

$$P_{t+1} = \emptyset$$

从 $i=1$ 开始

 计算 F_i 中个体的拥挤度

$$P_{t+1} = P_{t+1} \cup F_i$$

$$i = i + 1$$

直到 $|P_{t+1}| + |F_t| \leq N$

$sort(F_t, \prec_n)$

$P_{t+1} = P_{t+1} \cup F_t[1:(N - |P_{t+1}|)]$

$Q_{t+1} = new(P_{t+1})$ \\ 通过遗传算子产生新种群

$t = t + 1$

如图 3.3 所示^[8], 首先将第 t 代产生的新种群 Q_t 与父代 P_t 合并组成 R_t , 种群大小为 $2N$ 。然后 R_t 进行非支配排序, 产生一系列非支配集 F_i 并计算拥挤度。由于子代和父代个体都包含在 R_t 中, 则经过非支配排序以后的非支配集 F_1 中包含的个体是 R_t 中最好的, 所以先将 F_1 放入新的父代种群 P_{t+1} 中。如果 F_1 的大小小于 N , 则继续向 P_{t+1} 中填充下一级非支配集 F_2 , 直到添加 F_3 时, 种群的大小超出 N , 对 F_3 中的个体进行拥挤度排序 ($sort(F_3, \prec_n)$), 取前 $N - |P_{t+1}|$ 个个体, 使 P_{t+1} 个体数量达到 N 。然后通过遗传算子 (选择、交叉、变异) 产生新的子代种群 Q_{t+1} 。

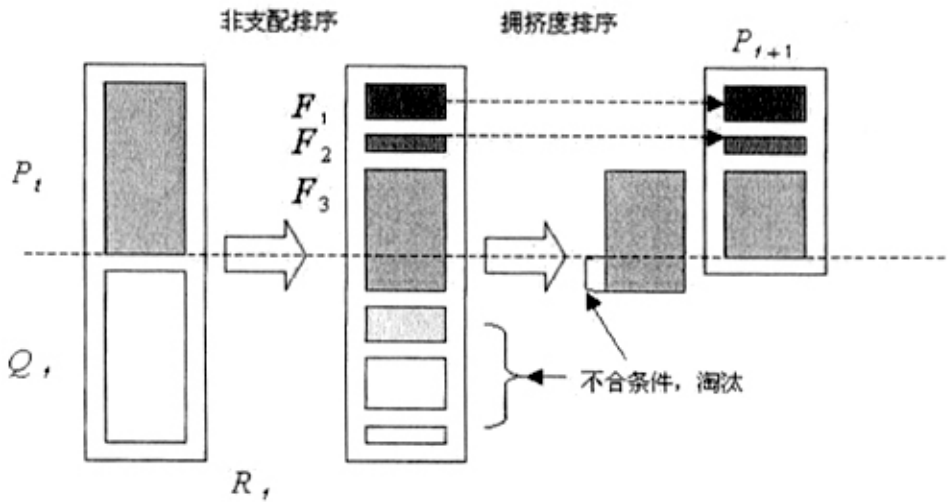


图 3.3 NSGA-II 流程

算法的整体复杂性为 $O(mN^2)$, 由算法的非支配排序部分决定。当排序产生

的非支配集的个体数目足够填充 P_{t+1} 时, 就不必再继续对剩下的部分排序了。非支配解的多样性由拥挤度比较算子保证, 不需要额外的共享参数。通过对当前解和种群中所有个体的分级存放, 使得最佳个体不会丢失。

3.5 本章小结

通过介绍非支配排序遗传算法 (NSGA) 及其改进算法 NSGA-II 的基本原理和流程, 我们了解到 NSGA-II 解决了 NSGA 中存在的 3 个问题: 降低了计算复杂度; 引入精英策略; 采用拥挤度及其比较算子代替了共享半径 σ_{share} 。使得 NSGA-II 在处理多目标优化问题上有更好的性能。

第四章 NSGA-II 在变结构控制系统中的应用^[47]

变结构控制^[48-50] (Variable Structure Control), 是一种实现线性和非线性系统鲁棒控制的方法。本章将 NSGA-II 引入变结构控制系统, 针对多输入控制系统输入参数不易确定, 控制性能难以满足的特点, 利用 NSGA-II 算法在处理多目标优化问题上的优势, 对变结构控制参数进行优化, 抑制了系统的超调量, 缩短了动态响应时间, 降低了抖振。仿真结果证明了该方法的有效性。

4.1 变结构控制研究概述

20 世纪 50 年代, 前苏联学者 Emelyanov 首先提出了变结构控制系统 (Variable Structure Control System, VSCS) 的概念, 并且逐步形成了控制系统的综合方法, Utkin 等在此基础上进一步发展、完善了变结构控制理论, 使其成为控制科学中的一个重要的分支。

变结构控制是一种实现线性与非线性系统鲁棒控制的方法。由于系统的结构在控制的过程中会发生变化, 所以被称作变结构控制, 也被称作滑模变结构控制。这种控制策略与常规控制的根本区别在于控制的不连续性, 它利用一种特殊的滑模控制方式, 迫使系统沿给定的状态轨迹滑动到期望点, 这种运动称之为滑动模态或“滑模”运动。由于给定的状态轨迹与系统参数及外部扰动无关, 因而系统在滑模面上运动时具有比鲁棒性更加优越的不变性。变结构控制是近二十年来研究较多的一类控制方法, 尤其近年来该方面的文献大量出现, 取得了丰硕的研究成果。

4.1.1 变结构控制系统的基本概念^[51]

定义 4.1 [滑模变结构控制]

设有一非线性系统:

$$\dot{x} = f(x, u, t) \quad x \in R^n, u \in R^m, t \in R \quad (4.1)$$

和切换函数向量

$$S = S(x), \quad S \in R^m \quad (4.2)$$

寻求变结构控制:

$$u_i(x) = \begin{cases} u_i^+(x) & \text{当 } S_i(x) > 0 \\ u_i^-(x) & \text{当 } S_i(x) < 0 \end{cases} \quad (4.3)$$

这里的变结构体现在 $u^+(x) \neq u^-(x)$, 使得设计的系统能满足以下三个条件:

- (1) 滑动模态的存在;
- (2) 满足到达条件;
- (3) 滑模运动渐进稳态并具有良好的动态特性;

由式(4.1)和式(4.2)所描述的且能满足上面条件(1), (2), (3)的系统称之为变结构控制系统, 由于滑动模态是变结构所特有的现象, 所以又称之为滑动模态控制 (Sliding Mode Control, SMC), 通俗叫做滑模变结构控制。

定义 4.2 [切换平面] 是指变结构控制系统的切换函数在相平面上形成的平面, 也被称作滑平面, 该平面对应的方程记为 $S(x)$ 。当系统为二阶系统时, 该平面为直线; 为三阶系统时, 该平面成为一个三维平面; 当系统高于三阶时, 该平面成为一个超维平面。

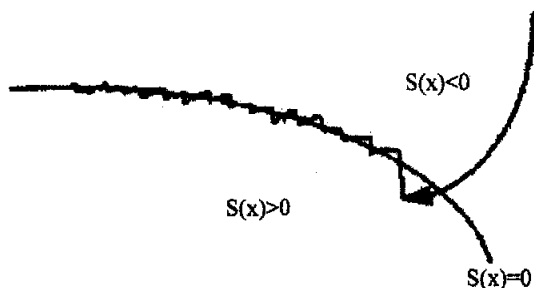


图 4.1 切换面上的滑模运动

如图 4.1 所示, 当实现变结构控制后, 切换平面使系统的状态向该平面运动, 呈现出一种“吸引”的现象, 但由于系统存在惯性, 系统的状态将在切换平面上下做小幅度的高频振动, 同时沿着切换平面向状态零点运动。

定义 4.3[滑动模态] 系统状态沿定义 4.2 的切换平面运动, 即为滑模运动, 或称为滑动模态。滑动模态完全决定于切换面的选择, 与系统本身无关。

定义 4.4[到达条件] 是使切换平面 $S(x)=0$ 以外的相轨迹都能在有限的时间内到达切换面的条件。

4.1.2 变结构控制系统的不变性和抖振问题

由于滑动模态的本质特点是对内部参数的摄动和外部扰动作用具有不变性, 或称不灵敏性、自适应性, 因此, 变结构控制系统的鲁棒性很强。

抖振是影响变结构控制技术应用的主要原因。理论上的滑动模态是降维光滑运动, 但在具体实现变结构控制时, 理想的开关特性不可能实现, 主要是由于时间上的延迟和空间上的滞后等原因使得滑动模态呈抖动形式, 从而在光滑的滑动上叠加了自振, 使系统状态在相平面上的轨迹呈现锯齿形状, 这种现象被称为抖振。抖振是变结构控制的致命缺点, 它严重妨碍了滑模控制获得的预期效果, 影响了控制的精确性, 增加能量的消耗并可能对系统造成损害。因此, 变结构控制的研究重点已经转移到如何消弱并防止抖振的研究上来。

4.2 变结构控制基本原理

4.2.1 变结构控制的三个要素

(1) 滑动模态的存在条件

Utkin 首先提出了滑动模态存在的充分条件^[52]:

$$\lim_{s \rightarrow 0^+} \dot{S} < 0 \text{ 及 } \lim_{s \rightarrow 0^-} \dot{S} > 0 \quad (4.4)$$

或者

$$\lim_{s \rightarrow 0^+} \dot{S} < 0 < \lim_{s \rightarrow 0^-} \dot{S} \quad (4.5)$$

式(4.5)也可合并为

$$\lim_{S \rightarrow 0} \dot{S}S < 0 \quad (4.6)$$

(2) 滑动模态得到达条件^[51]

变结构控制系统的到达条件比较简单,最初由前苏联学者 Emelyanov 提出了定义:

$$\begin{cases} \dot{S} < 0 & \text{当 } S > 0 \\ \dot{S} > 0 & \text{当 } S < 0 \end{cases} \quad (4.7)$$

或者合并写为

$$\dot{S}S < 0 \quad (4.8)$$

式(4.8)也可称作广义滑模条件。显然,满足了广义滑模条件必然同时满足滑模存在性和到达条件。

Chan 和我国学者高为炳后来又提出了用趋近率来保证达到条件^[53],这样还能保证趋近模态的动态品质。目前已有的几种趋近率为^[54]:

a) 等速趋近率

$$\dot{S} = -\varepsilon \operatorname{sgn}(S), \varepsilon > 0 \quad (4.9)$$

b) 指数趋近率

$$\dot{S} = -\varepsilon \operatorname{sgn}(S) - KS, \varepsilon > 0, K > 0 \quad (4.10)$$

c) 幂次趋近率

$$\dot{S} = -K |S|^\alpha \operatorname{sgn}(S), \text{ 其中 } K > 0, 1 > \alpha > 0 \quad (4.11)$$

d) 一般趋近率

$$\dot{S} = -\varepsilon \operatorname{sgn}(S) - f(S), \varepsilon > 0 \quad (4.12)$$

其中, $f(0) = 0$; 当 $S \neq 0$ 时, $Sf(S) > 0$

(3) 滑动运动稳定的条件

当满足条件(1)和(2)时,基本上就可以满足滑动稳定条件了。

4.2.2 基本原理

考虑如下的多输入状态方程:

$$\dot{x} = Ax + Bu \quad (4.13)$$

在这里, $x = [x_1 \ x_2 \ \dots \ x_n]^T \in R^n$, $u = [u_1 \ u_2 \ \dots \ u_m]^T \in R^m$,

$A = \{a_{ij}, i = 1, 2, \dots, n, j = 1, 2, \dots, n\} \in R^{n \times n}$, $B = \{b_{ij}, i = 1, 2, \dots, n, j = 1, 2, \dots, m\} \in R^{n \times m}$ 。

设滑模函数为:

$$s = c^T x \quad (4.14)$$

这里 $c = [c_1 \ c_2 \ \dots \ c_n]^T \in R^n$ 为滑平面参数, 一般情况下 $c_n = 1$ 。

所以

$$\dot{s} = c^T Ax + c^T Bu \quad (4.15)$$

假设 $B = [B_1 \ B_2 \ \dots \ B_m]$, 这里 $B_i = [b_{i1} \ b_{i2} \ \dots \ b_{in}]^T$ 。根据线性系统理论, 线性系统的单个输入对应的输出可以叠加, 所以针对某单一输入 u_i 的输出 $x^{(i)}$ 及对应的滑模函数 $s^{(i)}$, 有:

$$\dot{s}^{(i)} = c^T Ax^{(i)} + c^T B_i u_i \quad (4.16)$$

设 $\dot{s}^{(i)} = 0$, 可得其等效控制为:

$$u_{i,eq} = -(c^T B_i)^{-1} c^T Ax^{(i)} \quad (4.17)$$

为了保证控制算法的收敛性, 增加附加控制:

$$u_{i,d} = -\varepsilon_i \operatorname{sgn} s^{(i)} \quad (4.18)$$

所以有:

$$\begin{aligned} u_i &= u_{i,eq} + u_{i,d} \\ &= -(c^T B_i)^{-1} c^T Ax^{(i)} - \varepsilon_i \operatorname{sgn} s^{(i)} \end{aligned} \quad (4.19)$$

假设 $x^{(i)} = q_i x$, 其中 $0 < q_i < 1$ 且 $\sum_{i=1}^m q_i = 1$, 考虑到(4.14), 可得:

$$s^{(i)} = cx^{(i)} = q_i cx = q_i s \quad (4.20)$$

由(4.19)同时考虑到 $q_i > 0$, 可得:

$$u_i = -q_i (c^T B_i)^{-1} c^T Ax - \varepsilon_i \operatorname{sgn} s \quad (4.21)$$

由于变结构控制的滑平面参数 c , 切换控制量 ε_i 以及 q_i 的具体值一般根据经验人为设定, 需要较好的先验知识, 因此很难保证系统控制性能。本文引入 NSGA-II 用以优化参数, 通过多目标优化方法对待定参数进行优化, 可以保证取得的参数能获得良好的系统性能。下面将介绍基于 NSGA-II 的变结构控制方法。

4.3 基于 NSGA-II 的变结构控制方法

将(4.21)式代入(4.13)中可得变结构反馈控制系统:

$$\dot{x} = \left[I - \sum_{i=1}^m q_i B_i (c^T B_i)^{-1} c^T \right] Ax - \sum_{i=1}^m B_i \varepsilon_i \operatorname{sgn} s \quad (4.22)$$

所以:

$$\begin{aligned} s\dot{s} &= sc^T \dot{x} \\ &= sc^T \left[I - \sum_{i=1}^m q_i B_i (c^T B_i)^{-1} c^T \right] Ax - s \sum_{i=1}^m c^T B_i \varepsilon_i \operatorname{sgn} s \\ &= s \left[c^T - \sum_{i=1}^m q_i c^T \right] Ax - \sum_{i=1}^m c^T B_i \varepsilon_i |s| \\ &= - \sum_{i=1}^m c^T B_i \varepsilon_i |s| \end{aligned} \quad (4.23)$$

选 $\varepsilon_i > 0$, 可知当选择 c , 使所有 $c^T B_i > 0$ 时, 上式小于零, 即可保证系统的稳定性。

重新考虑系统(4.13), 有:

$$\dot{x} = Ax + Bu(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m, c_1, c_2, \dots, c_{n-1}, q_1, q_2, \dots, q_{m-1}) \quad (4.24)$$

由于切换控制量 ε_i ($i=1,2,\dots,m$)，变结构控制的滑平面参数 c_j ($j=1,2,\dots,n-1$) 和 q_k ($k=1,2,\dots,m-1$) 的具体值一般根据经验选取，因此很难保证系统控制性能。为了得到更为合理的参数，保证系统输出的超调量、动态响应时间和抖振都尽量小，采用多目标优化方法对参数进行优化，不需要人为干预直接获得各项参数，客观、方便。根据文献[7]可知，NSGA-II 算法与其它多目标优化算法相比其收敛速度快，种群个体在解空间分布均匀，算法的鲁棒性好，借助 NSGA-II 优化系统参数，将种群个体表示为 $(\varepsilon_1, \dots, \varepsilon_m, c_1, \dots, c_{n-1}, q_1, \dots, q_{m-1})$ ，以最小化系统的超调量、动态响应时间，以及抖振为目标函数。

设 T 为使输出稳定的足够大的常数，可得如下目标函数：

1) 使所有系统输出的超调量最小化

$$\sup \left\{ \frac{x_i}{-x_i(0)} \right\}, i=1,2,\dots,n \quad (4.25)$$

2) 使所有系统输出动态响应时间最小化

$$\{t, \text{当}|s(t)| \leq \delta\}, i=1,2,\dots,n \quad (4.26)$$

其中 δ 为人为指定的常数，当输出小于 δ 时认为已进入稳态。

3) 系统抖振最小化

$$\int_0^T s^2 dt \quad (4.27)$$

约束条件：

$$\sum_{i=1}^{m-1} q_i < 1, 0 < q_i < 1 \quad (4.28)$$

缩短系统动态响应时间必然会影响系统超调量增加，使抖振加大，所以通过多目标优化算法优化参数，可以找到满足三个目标函数的 Pareto 解集。算法只需运行一次即可得到多组参数，设计者的可根据不同要求选择参数。下面就这一方法给出算例及仿真结果。

4.4 仿真结果

考虑如下的控制系统:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (4.29)$$

其种群个体表示为 $(\varepsilon_1, \varepsilon_2, c_1, q_1)$, 采用实数编码。由于 $0 < q_1 < 1$, 已满足约束条件式(4.28), 算法中不需要再加约束条件。算法参数如表 4.1 所示。表 4.2 为经过算法优化得到的具有代表性一组 Pareto 最优解。

表 4.1 算法参数

参数	数值	参数	数值
种群中个体数	100	交叉率	0.9
遗传代数	500	变异率	0.5
ε_1 取值范围	[0.001, 10]	ε_2 取值范围	[0.001, 10]
c_1 取值范围	[0.001, 10]	q_1 取值范围	(0, 1)

表 4.2 一组 Pareto 最优解

ε_1 值	0.001	0.001	0.002	0.001	0.003	0.002	0.001	0.001	0.003
ε_2 值	0.001	0.001	0.001	0.001	0.006	0.003	0.003	0.002	0.001
c_1 值	0.001	2.614	0.006	0.001	0.002	0.006	0.006	0.001	1.587
q_1 值	0.472	0.09	0.006	0.003	0.001	0.002	0.002	0.003	0.472

选取 $\varepsilon_1 = 0.003$, $\varepsilon_2 = 0.001$, $c_1 = 1.587$, $q_1 = 0.472$, 假设系统参数有 25% 的摄动, 同时有 $[-0.5, 0.5]$ 的外界扰动, 得到图 4.2, 即系统在无参数摄动、外界扰动时(标称系统)与系统存在参数摄动与外界扰动时(受扰系统)的仿真结果。其中虚线标称系统的状态响应曲线, 实线为受扰系统的状态响应曲线。

由图 4.2 中可以看出, 选用优化后的参数, 系统输出在 0.5 秒内进入稳定状态, 而最大超调量小于 0.1, 同时两条曲线基本重合, 证明该系统具有较强的鲁

棒性。

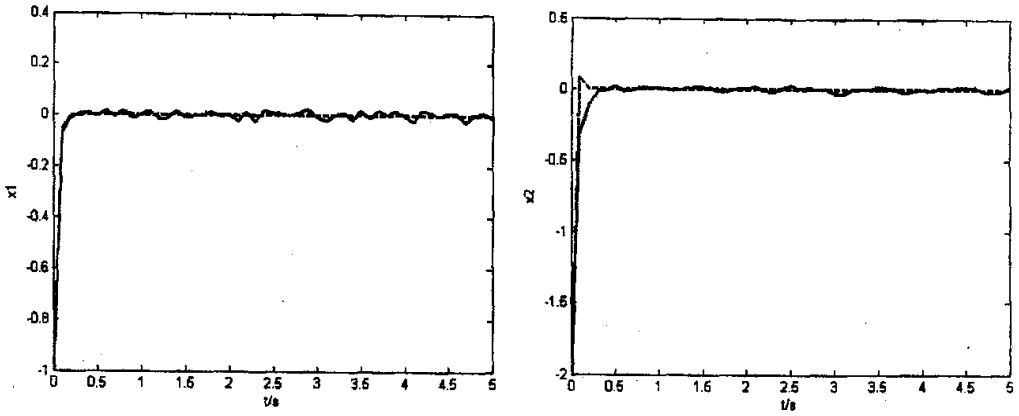


图 4.2 利用 NSGA-II 算法优化变结构控制参数后的状态响应曲线

--- 标称系统的状态响应曲线;

— 受扰系统的状态响应曲线;

图 4.3 中是当其它条件不变, 而 $\varepsilon_1=1$, $\varepsilon_2=1$, $c_1=1$, $q_1=0.3$ 时的曲线。

可以看出, 状态的最大超调量大于 0.2, 同时抖振也大于 0.1, 可知优化参数后的系统效果明显优于参数任选时的效果。

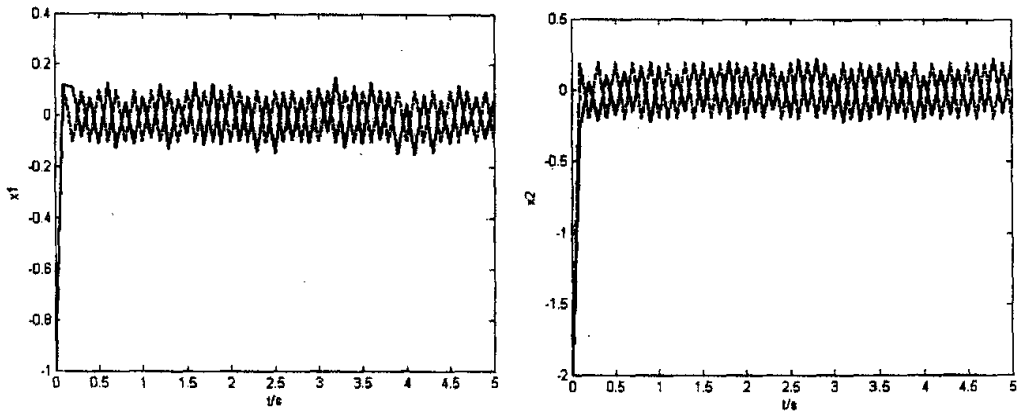


图 4.3 在参数任选时的状态响应曲线

--- 标称系统的状态响应曲线;

— 受扰系统的状态响应曲线;

4.5 本章小结

本章将多目标优化算法 NSGA-II 引入变结构控制系统，以系统输出超调量、动态响应时间和抖振的最小化为算法的目标函数，对原本需要人为设定的参数 $\varepsilon_i (i=1, \dots, m)$ 、 $c_j (j=1, \dots, n-1)$ 和 $q_k (k=1, 2, \dots, m-1)$ 进行优化。该方法克服了变结构控制系统输入参数不易确定，需依据经验选取参数的缺陷，抑制了系统的超调量，缩短了动态响应时间，降低了抖振。相比于传统的多目标优化算法，NSGA-II 只需运行一次就可以得到足够多的参数备选方案，满足了不同使用者对同一系统的不同要求。仿真结果证明了该方法的有效性。

第五章 基于 NSGA-II 与回归支持向量机的函数逼近方法^[55]

从动态系统的识别到控制,从故障检测到优化及分类,在众多的工程应用中,函数逼近是解决这些问题的核心^[56]。这类问题通常给出一个集合 $S = \{(x^1, y_1), \dots, (x^m, y_m)\} \subseteq R^n \times R$, 其中 $x^i \in R^n$ 为输入数据, $y_i \in R$ 为观测数据, 即输出数据^[57]。函数逼近的目的就是要找到一个函数 $f(x)$ 能够恰当的描述这些输入变量和观测数据之间的关系,从而准确的预测新的输入数据 x 对应的观测值 y 。近些年来,已有许多研究人员采用简单遗传算法(Simple Genetic Algorithm),提出了一些解决方案^[58, 59],在一定程度上解决了函数逼近和曲线拟合的问题。然而这些方法存在一些缺点:

- a) 固定的多项式形式限制了函数预测的精度。多项式的最高次数越高,精度越高,而精度越高,算法的运算复杂度也相应的增加了。
- b) 算法运行一次只能得到一个解,而这个解可能是局部最优解而不是全局最优解。也就是说,遗传算法仅运行一次得到的函数 $f(x)$ 不一定是最能表达 x 与 y 之间关系的函数,那么,函数 $f(x)$ 将无法准确预测新的输入数据 x 的输出值。

另外,支持向量机回归算法(Support Vector Machine Regression, SVMR)如今也被广泛应用于函数逼近方面^[60]。但是, SVMR 的参数通常需要根据经验选取,可能需要尝试多次才能得到较好的结果,经验因素对结果的影响很大。一些研究^[61, 62]为了实现参数的自动选取,引入遗传算法对参数进行优化。然而,在这些算法中,核函数的形式仍然需要人为选择。虽然核函数的形式对最终得到的解并没有很大影响,但核函数的计算复杂度对算法的运算时间的影响却是不可忽

略的,尤其是当算法应用于实时过程控制时,时间因素是不可忽略的。所以,在尽可能不影响预测精度的情况下,应尽量简化核函数的形式,采用计算复杂度较小的核函数。

鉴于 NSGA-II 在解决多目标优化问题上的优势,为了解决上述方法中存在的问题,本章提出了一种基于 SVM 与 NSGII 函数逼近方法。通过引入支持向量机 (Support Vector Machine, SVM) 中的结构风险最小化准则 (Structure Risk Minimization, SRM), 核函数的计算复杂度, 及函数逼近的偏差的范数作为多目标优化问题的目标函数, 实现了核函数形式和其他重要参数的自动选取。采用 NSGA-II 对 3 个目标函数进行优化。仿真结果表明, 该方法比传统的 SVM 方法更具优越性。

5.1 支持向量机概述

支持向量机是统计学习理论的一种实现方法,它较好地实现了结构风险最小化思想。支持向量机理论源于 Vapnik 在 1963 年提出的用于解决模式识别问题的支持向量方法^[63]。该方法从训练集中选择一组特征子集,使得对特征子集的线性划分等价于对整个数据集的分割。这组特征子集称为支持向量 (Support Vector, SV)。在此后近 30 年中,对 SV 的研究主要集中在对分类函数的改进和函数预测上。1971 年, Kimeldorf 提出使用线性不等约束重新构造 SV 的核空间^[64], 解决了一部分线性不可分的问题,为以后的 SVM 研究开辟了道路。1995 年, Vapnik 提出了统计学习理论^[65],较好地解决了线性不可分的问题,正式奠定了 SVM 的理论基础^[66]。

5.1.1 结构风险最小化准则

经验风险最小化准则已经在神经网络等学习机器中得到了广泛的应用。该准则是从样本无穷大假设条件下出发的,在有限样本的情况下,它无法解决学习机器的复杂性和推广能力之间的矛盾,因此,采用该准则的学习机器会出现过学习现象。结构风险最小化准则利用推广能力界的理论解决了这一矛盾。

定义函数 $Q(z, \alpha), \alpha \in \Lambda$ 的集合 S 具有一定的结构,这一结构是由一系列嵌套

的函数子集 $S_k = \{Q(z, \alpha), \alpha \in \Lambda_k\}$ 组成的。它们满足 $S_1 \subset S_2 \subset \dots \subset S_k \dots$ ，其中结构中的元素满足下面两个性质^[67]：

- a) 每个函数集 S_k 的 VC 维 h_k 是有限的，因此， $h_1 \leq h_2 \leq \dots \leq h_k \dots$ 。
- b) 结构的任何元素 S_k 包含一个完全有界函数的集合 $0 \leq Q(z, \alpha) \leq B_k, \alpha \in \Lambda_k$ 。

因此，在同一个子集中置信区间相同。在每个子集中寻找最小经验风险，选择最小化经验风险与置信区间之和最小的子集，就可以达到期望风险的最小。这个子集中使经验风险最小的函数就是所要求的最优函数。这种思想称作结构风险最小化（Structure Risk Minimization，简称 SRM 准则）。

SRM 准则定义了在对给定数据逼近的精度和逼近函数的复杂性之间的一种折衷。随着子集 S_k 序号的增加，经验风险的最小值减少，决定置信范围的项增加，SRM 准则通过对子集 S_k 的选择将这两者都考虑在内。如图 5.1 所示，在函数子集 S_2 中，数据逼近的精度和逼近函数的复杂性之间取得了一种最佳折衷，这时的模型有最好的推广能力。

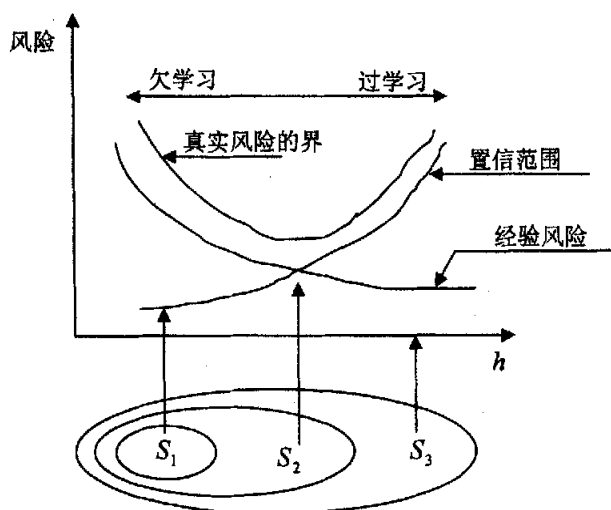


图 5.1 最小化经验风险的界

结构风险最小化准则为我们提供了一种不同于经验风险最小化的更科学的学习机器设计准则。但这个准则的实现并不容易，关键是如何构造函数子集的结构，然而对于这一问题目前尚无一般性理论。

5.1.2 支持向量机基本思想

支持向量机在形式上类似一个神经网络，输出是中间节点的线性组合，每个中间节点对应一个支持向量。其结构如图 5.2 所示

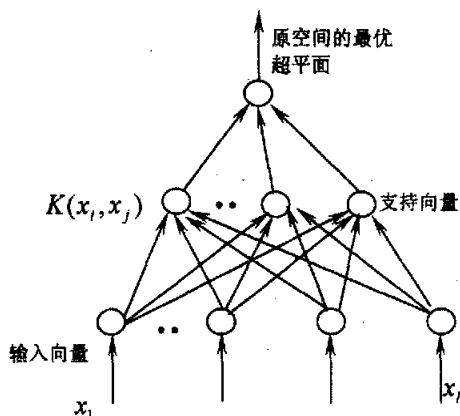


图 5.2 支持向量机的结构

支持向量机的实现是基于如下的思想：通过非线性映射 $\phi(\cdot)$ 把 n 维向量映射到高维特征空间中，使向量在高维空间中的映像具有线性关系，以便在这个空间中构造线性最优决策函数。在构造最优决策函数时，利用结构风险最小化准则，需要计算内积。而支持向量机巧妙地利用原空间的核函数取代了高维特征空间中的内积运算，即 $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ ，也被称为内积的回旋，用核函数实现了输入输出之间的非线性关系，避免了灾难。因此，构造支持向量机由构造线性最优决策函数和内积的回旋两个过程组成。在此过程中，我们甚至无需知道映射 $\phi(\cdot)$ 的具体形式。

核函数方法避免了非线性映射 $\phi(\cdot)$ 的具体形式，避免了特征空间中的内积运算。该方法已经成为现代机器学习中的一种重要的方法，被看作学习理论上的一次革命和转折点。它架起了线性学习问题和非线性学习问题之间的一座桥梁，使

得在线性学习领域所取得的成果可以推广到非线性学习领域。因而，核方法的提出在机器学习领域的影响是深远的。

5.1.3 支持向量机的特点

支持向量机方法是建立在统计学习理论的VC维理论和结构风险最小化准则基础上的，是集优化、核、最佳推广能力等特点于一身的统计学习理论的一种实现方法。根据有限样本信息，在模型的复杂性（对特定训练样本的学习精度）和学习能力（无错误地识别样本的能力）之间寻求最佳折衷，以期获得最好的推广能力。其主要特点如下：

1. 专门针对有限样本情况，得到现有信息下的最优解而不仅仅是样本数趋于无穷大时的最优解。
2. 采用结构风险最小化准则，具有非常好的推广能力。
3. 最终将转化成为一个二次型寻优问题。理论上，可以得到全局最优解，解决了神经网络方法中无法避免的局部极值问题。
4. 巧妙地解决了维数问题，其算法复杂度与样本无关，非常适合于处理非线性问题。只需选取不同的核函数，就可实现多项式逼近、径向基函数（RBF）方法、多层感知器网络等许多现有学习算法。

随着理论的不完善，支持向量机的应用逐渐成为各国研究者的研究重点。目前，支持向量机算法在模式识别、回归估计、概率密度函数估计等方面已有广泛应用。

5.2 支持向量机回归算法原理

5.2.1 线性可分情况

Vapnik 利用分类支持向量机的思想来解决回归问题，设给定的输入样本为 u 为 n 维向量， r 个训练样本及其输出值可表示为：

$$(u_i, v_i) (i=1, 2, \dots, r), \{u_i \in R^n, v_i \in R\}$$

采用 ε -不灵敏损失函数, 即:

$$L(v, f(u, \alpha)) = L(|v - f(u, \alpha)|_\varepsilon) = |v - f(u, \alpha)|_\varepsilon$$

$$|v - f(u, \alpha)|_\varepsilon = \begin{cases} 0 & \text{若 } |v - f(u, \alpha)| \leq \varepsilon \\ |v - f(u, \alpha)| - \varepsilon & \text{其他} \end{cases} \quad (5.1)$$

这种损失函数描述这样一种情况, 即如果预测值和实际值之间的差别小于 ε , 则损失等于零, 其示意图如下:

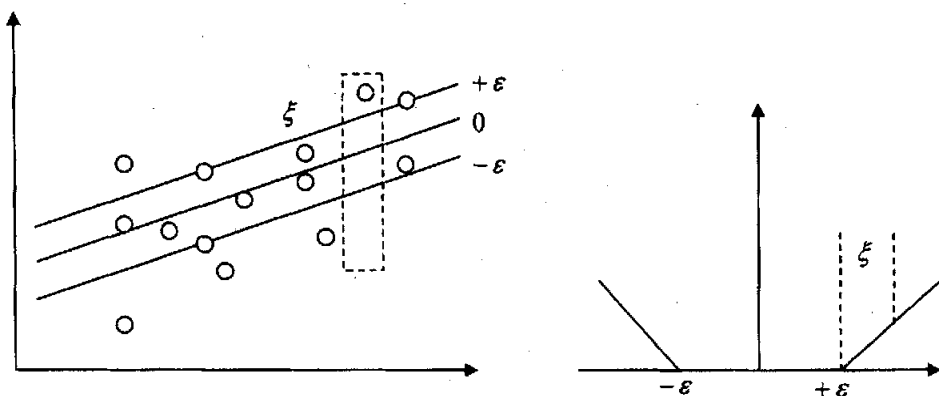


图 5.3 ε -不灵敏函数

在图 5.3 中, 右边的图是 ε -不灵敏函数示意图, 横坐标是拟合误差 $|v - f(u, \alpha)|$, 纵坐标表示损失函数值 $|v - f(u, \alpha)|_\varepsilon$; 左边的图形表示在 ε -不灵敏函数作用下线性拟合的情况。首先考虑线性回归的情况, 即用线性回归函数 $f(u) = \langle w \cdot u \rangle + b$ 来拟合训练数据 $\{u_i, v_i\}, i = 1, \dots, r, u_i \in R^n, v_i \in R$ 。如果所有训练数据都可以在精度 ε 下无误差地用线性函数拟合, 即

$$\begin{cases} v_i - \langle w \cdot u_i \rangle - b \leq \varepsilon \\ \langle w \cdot u_i \rangle + b - v_i \leq \varepsilon \end{cases} \quad i = 1, \dots, r \quad (5.2)$$

但实际上并不是如此, 总会有大于 ε 的拟合误差, 考虑到允许拟合误差的情况, 引入松弛因子 $\xi_i \geq 0$ 和 $\xi_i^* \geq 0$ 。回归支持向量机的优化问题提法如下:

$$\begin{aligned} \text{Min} \quad & J = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^r (\xi_i + \xi_i^*) \\ \text{s.t.} \quad & \begin{cases} v_i - \langle w \cdot u_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w \cdot u_i \rangle + b - v_i \leq \varepsilon + \xi_i^* \end{cases} \quad i = 1, \dots, r \end{aligned} \quad (5.3)$$

上述优化问题的对偶问题是:

$$\begin{aligned}
 \max W(\alpha, \alpha^*) &= -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \langle u_i, u_j \rangle + \sum_{i=1}^l v_i (\alpha_i^* - \alpha_i) - \varepsilon \sum_{i=1}^l (\alpha_i^* + \alpha_i) \\
 \text{s.t. } \sum_{i=1}^l (\alpha_i - \alpha_i^*) &= 0 \\
 0 \leq \alpha_i, \alpha_i^* &\leq C \quad i=1, \dots, r
 \end{aligned} \tag{5.4}$$

这样可以利用二次型规划方法来解出 α 、 α^* ，同时利用 KKT 条件可以计算出 b ，在最优点拉格朗日乘子和约束条件的乘积为零：

$$\begin{aligned}
 \alpha_i(\varepsilon + \xi_i - v_i + \langle w, u_i \rangle + b) &= 0 \\
 \alpha_i^*(\varepsilon + \xi_i^* + v_i - \langle w, u_i \rangle - b) &= 0 \\
 (C - \alpha_i)\xi_i &= 0 \\
 (C - \alpha_i^*)\xi_i^* &= 0
 \end{aligned} \tag{5.5}$$

如果 $\alpha_i, \alpha_i^* \in (0, C)$ ，则 $\xi_i = \xi_i^* = 0$ ，则有：

$$b = v_i - \langle w, u_i \rangle - \varepsilon \quad \alpha_i \in (0, C) \tag{5.6}$$

或

$$b = v_i - \langle w, u_i \rangle + \varepsilon \quad \alpha_i^* \in (0, C) \tag{5.7}$$

根据上面的推导，可以得到所求的回归函数：

$$\begin{aligned}
 f(u) &= \langle w, u \rangle + b \\
 &= \sum_{i=1}^l (\alpha_i^* - \alpha_i) \langle u_i, u \rangle + b
 \end{aligned} \tag{5.8}$$

α_i 、 α_i^* 中只有少部分不为 0，它们对应的样本就是支持向量，一般是在函数变化比较剧烈的位置上的样本。

5.2.2 线性不可分情况

数据样本 $(u_i, v_i) (i=1, 2, \dots, r)$ ， $\{u_i \in R^n, v_i \in R\}$ 同 5.2.1 节，对于线性不可分情况，假设存在非线性映射 $\phi(u_i)$ ，能够找到一个函数 $f(\phi(u), w)$ ，使得函数与所有样本观测值 v_i 之间的距离不超过 ε ，并且尽可能的平滑。

考虑:

$$v = f(\varphi(u), w) = \langle w \cdot \varphi(u) \rangle + b \quad (5.9)$$

其中, $w \in R^n$, $\langle \cdot \rangle$ 表示点积, $b \in R$ 为偏差.

引入松弛变量 ξ_i 和 ξ_i^* , 由 SVM 得到的最优解可表示为:

$$\begin{aligned} \min & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^k (\xi_i + \xi_i^*) \\ \text{s.t.} & \begin{cases} v_i - \langle w \cdot \varphi(u_i) \rangle - b \leq \varepsilon + \xi_i \\ \langle w \cdot \varphi(u_i) \rangle + b - v_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (5.10)$$

其中, 平衡因子 C 为常数且 $C > 0$, 它决定了 f 的光滑度与偏差大于 ε 的和之间的平衡关系.

类似地, 定义:

$$|v - f(\varphi(u), w)| = \begin{cases} |v - f(\varphi(u), w)| - \varepsilon, & |v - f(\varphi(u), w)| > \varepsilon \\ 0, & \text{其它} \end{cases} \quad (5.11)$$

参照拉格朗日乘子和 KKT 条件, 可得到回归方程为:

$$\begin{aligned} \min Q(\alpha) &= \frac{1}{2} \sum_{i=1}^r (\alpha_i - \alpha_i^*) (\alpha_i - \alpha_i^*) \langle \varphi(u_i), \varphi(u_i) \rangle \\ &+ \sum_{i,j=1}^r \alpha_i (\varepsilon - v_i) + \sum_{i,j=1}^r \alpha_i^* (\varepsilon + v_i) \\ \text{s.t.} & \begin{cases} \sum_{i=1}^r (\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, 2, \dots, r \end{cases} \end{aligned} \quad (5.12)$$

则所求的逼近函数可表示为:

$$\begin{aligned} f(u, w) &= \langle w \cdot \varphi(u) \rangle + b \\ &= \sum_{i=1}^{nsv} (\alpha_i - \alpha_i^*) \langle \varphi(u_i) \cdot \varphi(u) \rangle + b \end{aligned} \quad (5.13)$$

其中, u_i 为支持向量, nsv 为支持向量的个数, α_i 为系数, b 为初值. 定义核函数为:

$$K(u_k, u_i) = \varphi(u_k)^T \varphi(u_i) \quad (5.14)$$

则回归函数可表示为：

$$f(u) = \sum_{i=1}^{nsv} \alpha_i K(u, u_i) + b \quad (5.15)$$

5.3 基于 NSGA-II 与回归支持向量机的函数拟合方法

本节提出的基于 NSGA-II 与 SVM 的优化算法，结合了 NSGA-II 在解决多目标优化问题方面的优势，和 SVM 解决曲线拟合的经典准则，将结构风险最小化准则引入算法，作为目标函数进行优化。不同形式的核函数其运算时间各有不同，当数据运算量大时，核函数的形式对算法整体的运算时间将不可忽略，本节提出了相应的解决方案，实现了自动选择运算复杂度小的核函数。

5.3.1 确定目标函数

令 $\beta_i = \frac{\alpha_i}{C}$, $\beta_i^* = \frac{\alpha_i^*}{C}$, 则(5.12)式变为：

$$\begin{aligned} \min J_1 = & \frac{1}{2} C^2 \sum_{i,j=1}^r (\beta_i - \beta_i^*)(\beta_j - \beta_j^*) \langle x_i, x_j \rangle \\ & + C \sum_{i=1}^r \beta_i (\varepsilon - y_i) + C \sum_{i=1}^r \beta_i^* (\varepsilon + y_i) \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^r (\beta_i - \beta_i^*) = 0 \\ 0 \leq \beta_i, \beta_i^* \leq 1, i=1, 2, \dots, r \end{cases} \end{aligned} \quad (5.16)$$

满足(5.16)式的解必然满足结构风险最小化准则，所以选择(5.16)式作为多目标优化问题的一个目标函数，其中 $\beta_1 \dots \beta_r, \beta_1^* \dots \beta_r^*, C, \varepsilon, b$ 为待优化参数。

另外，我们将曲线偏差的范数 $\sum_{i=1}^r (y_i - f(x_i))^2$ 引入，作为另一个目标函数。

通过(5.17)式对偏差的范数进行最小化，以保证得到的拟合曲线与原曲线尽可能的接近。

$$\min J_2 = \sum_{i=1}^r (y_i - f(x_i))^2 \quad (5.17)$$

考虑到当数据运算量大时,核函数的运算复杂度将大大影响算法的整体运算时间,因此,我们将核函数的运算复杂度作为第三个目标函数。通过(5.18)式对运算复杂度进行最小化操作,以保证所选取的核函数尽量简单,其中 K 为待优化参数,代表核函数的类型,采用二进制编码。

$$\min J_3 = K \quad (5.18)$$

下面我们就来介绍一下,如何选取核函数,根据核函数的运算复杂度,对不同类型不同参数的核函数进行编码。

5.3.2 选取核函数

在支持向量机中,常用的核函数有以下几种:

- (1) 线性核: $K(x, x_i) = x \cdot x_i$
- (2) 多项式核: $K(x, x_i) = (1 + x \cdot x_i)^p, p = 1, 2, \dots, N$
- (3) 高斯基 RBF 核: $K(x, x_i) = \exp(-\frac{\|x - x_i\|^2}{2\sigma^2})$
- (4) 指数基 RBF 核: $K(x, x_i) = \exp(-\frac{\|x - x_i\|}{2\sigma^2})$
- (5) 样条核: $K(x, x_i) = \sum_{r=0}^p x^r x_i^r + \sum_{s=1}^N (x - \gamma_s)^p (x_i - \gamma_s)^p$
- (6) B 样条核: $K(x, x_i) = B_{2N+1}(\|x - x_i\|)$
- (7) Fourier 级数核: $K(x, x_i) = \frac{\sin[(N + \frac{1}{2})(x, x_i)]}{\sin[\frac{1}{2}(x - x_i)]}$
- (8) 感知器核: $K(x, x_i) = \tanh(\beta(x_i \cdot x) + b)$

在这里,我们以前三个核函数为例,进行运算复杂度的比较。

设样本数为 r , 三种核函数的算法复杂度均为 $O(r^2)$ 。采用线性核函数时,

系统要作 r^2 次乘法；采用多项式核函数时，则作 r^2p 次乘法和 r^2 加法（仅考虑 p 为整数时）；采用高斯基 RBF 核函数时，则作 $5r^2$ 乘法、 r^2 加法和 r^2 次指数运算。

通过理论分析，我们仅能得出三种核函数运算复杂度的大致顺序，即：线性核函数、多项式核函数、高斯基 RBF 核函数。由于不同的运算占用 CUP 的时间不尽相同，而同种核函数选取不同的参数，其计算时间也是有差异的。因此，我们选取具有代表性的核函数和参数进行计算复杂度的比较仿真。

在 Windows2000 系统下，采用 MATLAB6.5，对线性核函数（无参数）、多项式核函数（参数 $p=1,2,3,4,5$ ）、高斯基 RBF 核函数（参数 $\sigma=0.1,0.2,0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0$ ），在无干扰的情况下计算它们在样本数为 10000 时的 CUP 运行时间，进行比较。

根据上述仿真结果，用 4 位二进制编码 K 代表不同参数下的三种核函数，并约定随着 K 值的增加，核函数的计算复杂度增加，编码如表 5.1 所示。

表 5.1 K 的含义

K	Kernel from	Parameters	K	Kernel from	Parameters
0000	Linear	\	1000	RBF	0.5
0001	Poly	1	1001	RBF	1.0
0010	Poly	2	1010	RBF	1.5
0011	Poly	3	1011	RBF	2.0
0100	Poly	4	1100	RBF	2.5
0101	Poly	5	1101	RBF	3.0
0110	RBF	0.1	1110	RBF	3.5
0111	RBF	0.2	1111	RBF	4.0

5.3.3 个体表示方法

根据 3 个目标函数的待优化参数，其种群个体表示可为： $(\beta_1 \cdots \beta_r, \beta_1^* \cdots \beta_r^*, C, \varepsilon, b, K)$ 。由于在采用 NSGA-II 算法对种群个体进行初始化时，要同时满足(5.16)的约束条件 $\sum_{i=1}^r (\beta_i - \beta_i^*) = 0$ 和 $0 \leq \beta_i, \beta_i^* \leq 1, i=1,2,\dots,r$ ，因此，我们采取了以下操作：

(1) 参数 $\beta_i, i=1,2,\dots,r, \beta_i^*, i=1,2,\dots,r-1$, 为 $[0,1]$ 区间随机产生的数。

(2) 由 $\sum_{i=1}^r (\beta_i - \beta_i^*) = 0$, 得 $\beta_r^* = \sum_{i=1}^r \beta_i - \sum_{i=1}^{r-1} \beta_i^*$

(3) 若 $\beta_r^* < 0$, 则 $\beta_i = 1 - \beta_i^*, i=1,2,\dots,r, \beta_i^* = 1 - \beta_i^*, i=1,2,\dots,r$

(4) 若 $\beta_r^* > 1$, 则 $\beta_i = \frac{\beta_i}{\beta_r^*}, i=1,2,\dots,r, \beta_i^* = \frac{\beta_i^*}{\beta_r^*}, i=1,2,\dots,r$

因此, 种群个体的长度可由原来的 $2r+4$ 简化为 $2r+3$, 表示为 $(\beta_1 \cdots \beta_r, \beta_1^* \cdots \beta_{r-1}^*, C, \varepsilon, b, K)$, 采用实数编码, 其中 r 为训练样本数。

5.4 仿真结果

设函数:

$$f(x) = 0.5 \cdot \frac{\sin(\frac{30}{\pi}x)}{\frac{30}{\pi}}; x \in [-1, 1] \quad (5.19)$$

在 $[-1, 1]$ 区间均匀选取 101 个点作为函数输入, 其函数样本值:

$$y_i = f(x_i) + \rho, i=1, \dots, 101 \quad (5.20)$$

其中 ρ 为 $mean=0$ 和标准方差 $\sigma=0.04$ 的附加高斯白噪声。

采用带精英策略的非支配排序遗传算法 (NSGA-II), 参照表 5.2 选取参数。经过 500 代的优化运算, 算法自动选取参数为 $\sigma=0.2$ 的高斯基 RBF 核函数, $C=2.615702e-001$, $\varepsilon=2.145446e-001$ 的一组数据。所得拟合函数曲线 $\hat{f}(x)$ 与原函数曲线 $f(x)$ 比较, 如图 5.4 所示。采用传统的 SVM 对该曲线进行拟合, 根据经验选取核函数和其他参数, 如表 5.3 所示。所得拟合函数曲线与原函数曲线比较, 如图 5.5, 5.6, 5.7, 5.8 所示。

定义函数拟合偏差的范数为:

$$EN = \sum [\hat{f}(x_i) - f(x_i)]^2 \quad (5.21)$$

EN 越小, 表示算法得到的拟合函数曲线与原函数曲线的偏差就越小, 函数拟合的效果就越好。表 5.3 列出了不同算法在不同参数选择下得到的函数拟合 EN 值。

表 5.2 NSGA-II 的参数

Random Seed	0.2
Population size	200
Generation size	500
Real variable number	205
Probability of crossover (real code)	0.9
Probability of mutation (real code)	0.01
Binary variable number	1
String length (binary code)	4
Probability of crossover (binary code)	0.9
Probability of mutation (binary code)	0.5
Boundary of $\beta_1 \cdots \beta_r, \beta_1^* \cdots \beta_{r-1}^*$	[0,1]
Boundary of C	[0,10000]
Boundary of ε	[0,1]
Boundary of b	[-1,1]
Boundary of K	[0,15]

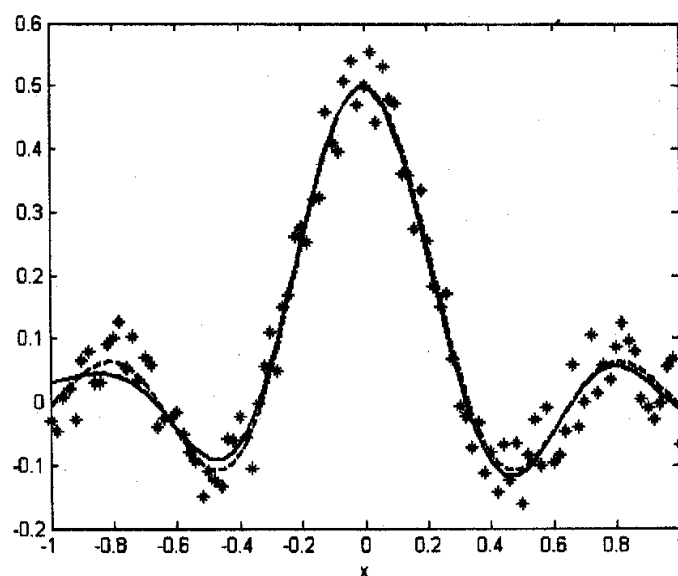


图 5.4 采用基于 SVM 与 NSGA-II 的函数拟合方法得到的结果

- Curve of $\hat{f}(x)$ achieved
- Original Curve of $f(x)$
- * Samples created by (5.20)

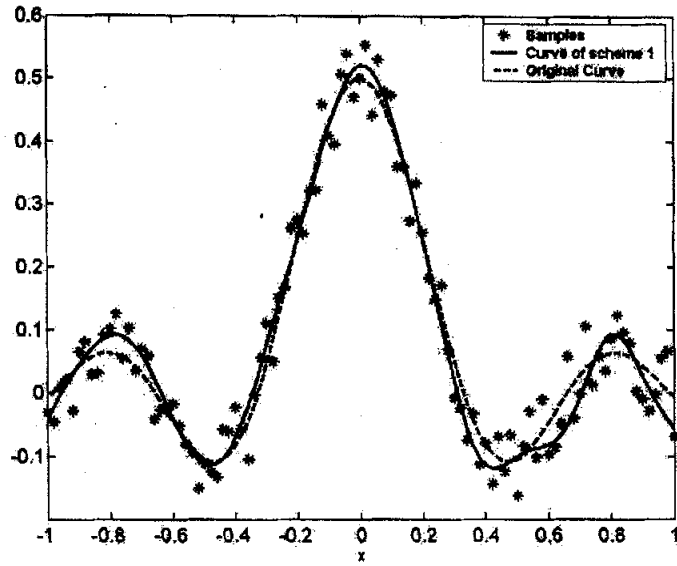


图 5.5 采用传统 SVM 的曲线拟合方法 1 得到的结果

— Curve of conventional SVM scheme 1
 --- Original Curve of $f(x)$
 * Samples created by (5.20)

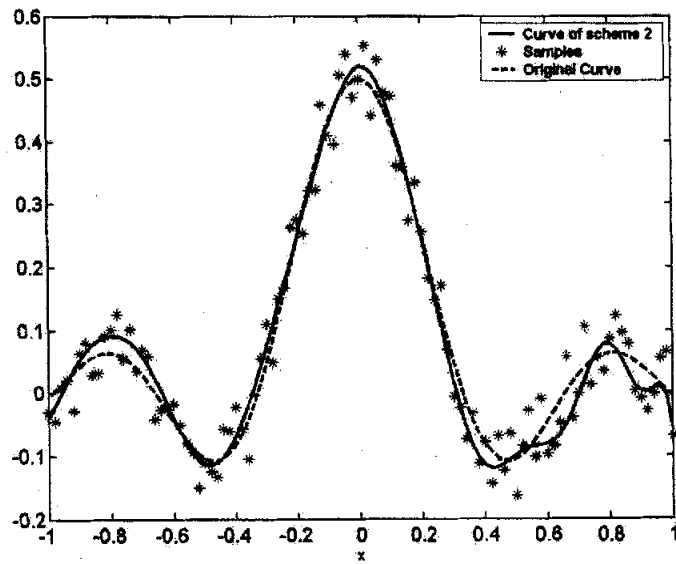


图 5.6 采用传统 SVM 的曲线拟合方法 2 得到的结果

— Curve of conventional SVM scheme 2
 --- Original Curve of $f(x)$
 * Samples created by (5.20)

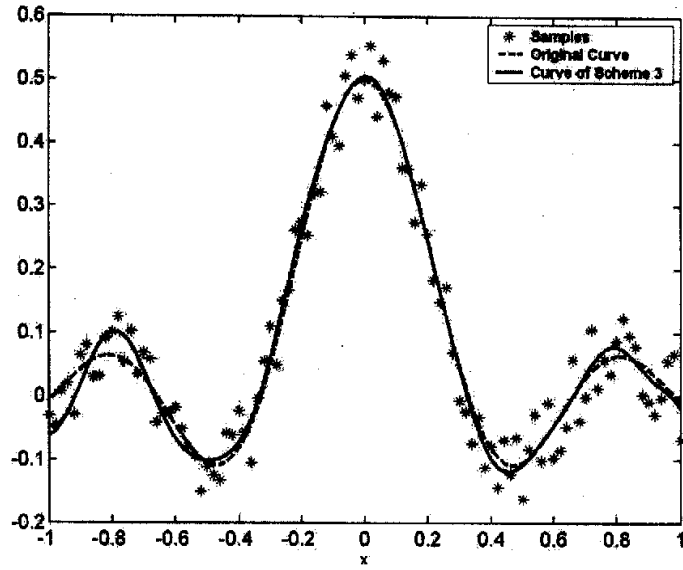


图 5.7 采用传统 SVM 的曲线拟合方法 3 得到的结果

— Curve of conventional SVM scheme 3
 --- Original Curve of $f(x)$
 * Samples created by (5.20)

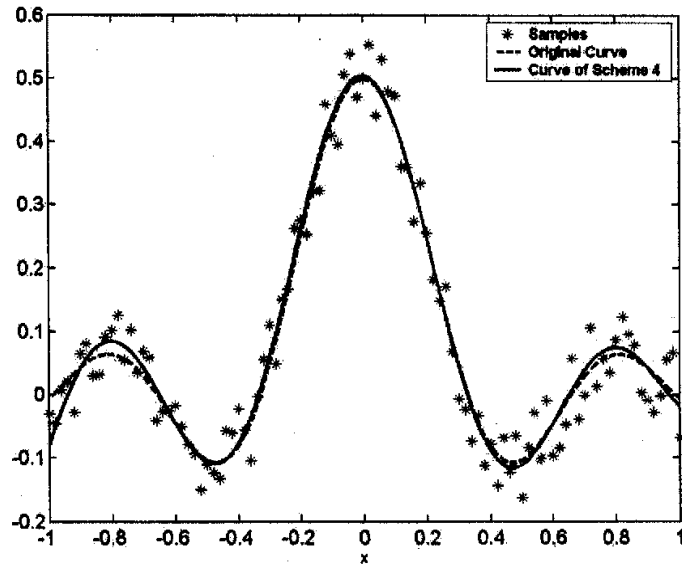


图 5.8 采用传统 SVM 的曲线拟合方法 4 得到的结果

— Curve of conventional SVM scheme 4
 --- Original Curve of $f(x)$
 * Samples created by (5.20)

表 5.3 不同方法不同参数下的 EN 值比较

	Kernel function and its parameter	C	ε	EN
Scheme of this paper	RBF 0.2	0.262	0.215	0.013856
Conventional SVM scheme 1	RBF 0.2	1000	0.01	0.054331
Conventional SVM scheme 2	RBF 0.2	10000	0.01	0.050503
Conventional SVM scheme 3	RBF 0.2	1000	0.05	0.030168
Conventional SVM scheme 4	RBF 0.5	1000	0.05	0.018721

由此可见, 传统的 SVM 方法需要不断的根据经验修正参数选择, 以更接近原函数。不同的操作人员由于经验不同, 选择的参数也不同, 因此, 人为因素将直接影响拟合结果。而基于 NSGA-II 和 SVM 的函数拟合方法, 完全是由算法自身进行参数选择, 尽可能的排除了人为因素。由表 5.3 所示, 该算法只需运行一次, 所得拟合曲线的 EN 值就已经很小了, 拟合的效果非常好, 而根据经验选取参数的 SVM 方法则需要多次尝试才有可能得到的较小 EN 值。

5.5 本章小结

本章提出了一种基于 NSGA-II 和 SVM 的函数拟合方法。传统的函数拟合方法, 如支持向量机回归的拟合方法, 需要人为选择参数和核函数, 经验因素对曲线拟合的结果有很大影响。而本章采用的方法, 将 SVM 的结构风险最小化准则、偏差的范数, 以及核函数的计算复杂度, 作为多目标优化问题的三个目标函数, 从而实现了 SVM 参数及核函数的自动选取。仿真结果表明, 该算法得到的拟合函数与原函数的曲线偏差很小, 与传统的 SVM 的函数拟合方法相比, 具有显著的优越性。

第六章 总结与展望

6.1 总结

遗传算法作为一种启发式的搜索算法,已经得到了广泛的应用,尤其是面向多目标优化问题的遗传算法应用更加广泛。本文研究的非支配排序遗传算法(NSGA)及其改进算法NSGA-II就是其中发展较快、优化效果较好、应用较广的一种方法。本文对算法的基本原理进行了系统的学习和研究,在算法的应用研究方面作了大胆的尝试,扩展了算法的应用范围,并提出了一种新的解决函数拟合问题的方法。

本文在广泛查阅国内外文献的基础上,做了的如下工作:

系统地介绍了遗传算法的一般流程和基本理论,以及目前已有的多目标优化遗传算法及其存在的一些问题。论述了NSGA的基本原理和不足之处,并对其改进算法NSGA-II提出的快速非支配排序法、拥挤度及其比较算子、精英策略及NSGA-II的主要流程作了详细的介绍。

将NSGA-II引入变结构控制系统。针对多输入控制系统输入参数不易确定,控制性能难以满足的特点,利用NSGA-II算法在处理多目标优化问题上的优势,以系统输出超调量、动态响应时间和抖振的最小化为算法的目标函数,对变结构控制的参数和滑平面参数进行优化。仿真结果证明了该方法能够克服参数需依据经验选取的缺陷,抑制了多输出控制系统的超调量,缩短了动态响应时间,降低了抖振,同时增强了系统对参数摄动和外部扰动的鲁棒性。相比于传统的多目标优化算法,NSGA-II只需运行一次就可以得到足够多的参数备选方案,满足了不同使用者对同一系统的不同要求。

将NSGA-II与SVM结合,提出了一种基于SVM和NSGA-II的函数拟合方法。传统的函数拟合方法,如支持向量机回归方法,需要人为选择参数和核函数,

经验因素对拟合结果的影响很大。而本文第五章提出的方法,首次将 SVM 的结构风险最小化准则引入 NSGA-II 算法,并实现了根据核函数的计算复杂度自动选取核函数形式。仿真结果表明,将 SVM 的结构风险最小化准则、偏差的范数,以及核函数的计算复杂度,作为多目标优化问题的三个目标函数,采用 NSGA-II 进行优化得到的拟合函数与原函数的曲线偏差很小,比传统的 SVM 的拟合方法更具优越性。

6.2 展望

多目标优化问题一直是科学和工程研究领域的难题和热点问题,针对传统的解决方法在处理大维数、多模态等复杂问题上存在的许多不足,国内外研究学者在简单遗传算法的基础上,发展出了多种多目标优化遗传算法,并广为应用。NSGA 与其它的多目标优化遗传算法相比,具有一定的优越性:优化目标个数任选, Pareto 最优解分布均匀,允许存在多个不同的等价解。NSGA-II 进一步解决了 NSGA 本身存在的问题,使得算法的性能更优越,因此具有重要的研究和应用价值。

本文所作的工作虽然为 NSGA-II 这种多目标优化遗传算法的应用研究开拓了新的方向,拓展了应用思路,但仍有许多有待改进的地方。如第四章提出的将 NSGA-II 应用于变结构控制系统,其应用研究针对的是线性系统,该方法可望进一步扩展到非线性系统;又如第五章提出的基于 SVM 和 NSGA-II 的函数拟合方法,其中的根据核函数计算复杂度问题选取核函数形式的方法仍有待讨论,文中的方法仅为核函数形式的自动选取提供了一种新的思路,后续工作仍大有可为。

现在,在工业过程自动化领域,越来越多的权威专家认为,自动化的目标不仅仅是保证产品的质量,还应该积极倡导“节能降耗”与“绿色化生产”的理念。这实际上就是一类关系到我们国家未来发展战略的多目标优化重大命题。因此,在多目标优化问题无处不在的今天,进一步发展 NSGA-II 的创新理论和应用方法,为工业化的应用实践提供重要的理论与技术支撑,具有十分重大的现实意义。

参 考 文 献

- [1] Schaffer J. D., Multiple objective optimization with vector evaluated genetic algorithms [A]. *Genetic Algorithms and their Applications: Proceeding of the First International Conference on Genetic Algorithms* [C], Lawrence Erlbaum, pp. 93-100, 1985.
- [2] Horn J, Nafpliotis N., Multiobjective Optimization using the Niche Pareto Genetic Algorithm. *Technical Report IlliGAI Report 93005*, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993
- [3] Lis J., Eiben A. E., A multi-sexual genetic algorithm for multi-objective optimization [A], Fukuda T., Furuhashi T., *Proceedings of the 1996 International Conference on Evolutionary Computation, IEEE* [C], Nagoya, Japan, pp. 59-64, 1996.
- [4] Darrell W., Evaluating evolutionary algorithms [J], *Artificial Intelligence*, 1996, 85: 245-276.
- [5] Wienke P. B, Lucasius C., Kateman G., Multicriteria target vector optimization of analytical procedures using a genetic algorithm [J], *Analytica Chimica Acta*, 1992, 265 (2): 211-225
- [6] Tseng C. H., Lu T. W., Minimax multiobjective optimization in structural design [J], *International Journal for Numerical Methods in Engineering*, 1990, 30: 1213 ~ 1228
- [7] Srinivas N., Deb K., Multiobjective function optimization using nondominated sorting genetic algorithms [J], *Evolutionary Computation*, 1995,2(3): 221-248
- [8] Deb K, Agrawal S., Pratap A., et al., A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II [A], *Proc of the Parallel Problem Solving from Nature VI Conf*[C], Paris, 2000: 849-858
- [9] 谢 涛, 陈火旺, 多目标优化与决策问题的演化算法[J], 中国工程科学, vol. 4, no. 2, pp: 59-68, Feb. 2002

- [10]王小平, 曹立明, 遗传算法——理论、应用与软件实现[M], 西安: 西安交通大学出版社, 2002.
- [11]关志华, 面向多目标优化问题的遗传算法的理论及应用研究[D], 天津: 天津大学博士学位论文, 2002 年 5 月
- [12]Holland, J., *Adaptation in Natural and Artificial Systems* [M], *University of Michigan Press*, Ann Arbor, MI, 1975; *MIT Press*, Cambridge, MA, 1992
- [13]Faser, A.S., Simulation of genetic systems [J], *Journal of Theoretical Biology*, 1962(2), pp. 329-346
- [14]DeJong A. K. An Analysis of the Behavior of a Class of Genetic Adaptive Systems [D], *Ph. D. thesis*, University of Michigan, 1975
- [15]Goldberg D. E., Deb K., A comparison of selection schemes used in genetic algorithms. In G. E. Rawlins Ed., *Foundations of Genetic Algorithms*, San Mateo, California: Morgan Kaufmann, pp. 69-93, 1991
- [16]Goldberg D. E., Richardson J., Genetic algorithm with sharing for multimodal function optimization. In J. J. Grefenstete Ed., *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum, pp. 41-49, 1987
- [17]Fonseca C. M., Fleming P. J., Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest Ed., *Proceedings of the Fifth International Conference on Genetic Algorithms* (San Mateo, California, 1993), University of Illinois at Urbana-Champaign: Morgan Kaufman Publishers, pp. 416-423, 1993
- [18]Powell D., Skolnick M. M., Using genetic algorithms in engineering design optimization with non-linear constraints. In S. Forrest Ed., *Proceedings of the Fifth International Conference on Genetic Algorithms* (San Mateo, California, Jul 1993), University of Illinois at Urbana-Champaign: Morgan Kaufmann Publishers, pp. 424-436, 1993
- [19]Cavichio D. J., Reproductive Adaptive Plans. In: *Proc. Of the ACM 1972 Annual conf.*, pp. 1-11, 1972

- [20]Michalewicz Z., et. al., Genetic Algorithms and Optimal Control Problem, *Proc. Of 290th IEEE Conf. On Decision and Control*, pp. 1664-1666, 1990
- [21]Davidor Y., A Naturally Occurring Niche and Species Phenomenon: The Model and First Result., *Proc. Of 4th Int. Conf. On Genetic Algorithms*, Morgan Kaufmann, pp. 257-263, 1991
- [22]Vose M. D., Modeling Simple Genetic Algorithms, *Foundations of Genetic Algorithms II*, Morgan Kaufmann Publishers, pp. 63-73, 1993
- [23]Schwefel H. P., Evolution and Optimum Seeking, *John Wiley&Sons*, 1995.
- [24]Srinivas M, Patnaik L.M., Adaptive Probabilities of Crossover and Mutation in Gas, *IEEE Trans. On SMC*, 1994, 24(4): 656-667
- [25]Nilsson N. J., Artificial Intelligence: A New Synthesis, *Morgan Kaufmann&机械工业出版社*, 1999
- [26]Kay C. T, Yun L., Multi-Objective Genetic Algorithm Based Time and Frequency Domain Design Unification of Linear Control System, *Research Report, University of Glasgow*, UK, 1995.
- [27]Goldberg. D. E., Genetic Algorithm in Search, Optimization and Machine Learning [M], *Addison-Wesley*, 1989.
- [28]Davie L.D., Handbook of Genetic Algorithm [M], *Van Nostrand Reinhold*, 1991.
- [29]Kuhn H. W., Tucker A W. Nonlinear programming. In J. Neyman Ed., *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability (Berkeley, California, 1951)*, University of California Press, pp. 481-492, 1951.
- [30]Rosenberg R. S., Simulation of genetic populations with biochemical properties, *Ph. D. thesis*, University of Michigan, Ann Harbor, Michigan, 1967.
- [31]Rodriguez-Vázquez K., Fonseca C. M., Fleming P. J., Multiobjective Genetic Programming: A Nonlinear System Identification Application. In J. R. Koza Ed., *Late Breaking Papers at the Genetic Programming 1997 Conference (Stanford University, California, July 1997)*, Stanford Bookstore, pp. 207-212, 1997.

- [32]Fonseca C. M., Fleming P. J., Nonlinear System Identification with Multiobjective Genetic Algorithms, *Proceedings of the 13th World Congress of IFAC*, pp. 187-192, 1996.
- [33]Fonseca C. M., Fleming P. J., An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, 1995, 3(1): 1-16
- [34]Chipperfield A., Fleming P., Gas Turbine Engine Controller Design using Multiobjective Genetic Algorithms. In A. M. S. Zalzal Ed., *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA'95*, pp. 214-219, 1995.
- [35]Todd D. S., Sen P. A., Multiple Criteria Genetic Algorithm for Containership Loading. In T Back Ed., *Proceedings of the Seventh International Conference on Genetic Algorithms*, Michigan State University: Morgan Kaufmann Publishers, pp. 674-681, 1997.
- [36]Horn J., Nafploitis N., Goldberg D. E., A niched Pareto genetic algorithm for multiobjective optimization. In Michalewicz, Z., editor, *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE Service Center, Piscataway, New Jersey, pp. 82-87, 1994.
- [37]Vedarajan Q. Chan L. C., Goldberg D. E., Investment Portfolio Optimization using Genetic Algorithms. In J. R. Koza Ed., *Late Breaking Papers at the Genetic Programming 1997 Conference*, Stanford Bookstore, pp. 255-263, 1997.
- [38]Zitzler E., Evolutionary algorithms for multiobjective optimization: Methods and applications. *Doctoral thesis ETH NO. 13398*, Zurich: Swiss Federal Institute of Technology (ETH), Aachen, Germany: Shaker Verlag, 1999.
- [39]Zitzler E., Thiele L., Multiobjective optimization using evolutionary algorithms-A comparative case study [A]. Eiben A E., et. al. *Parallel Problem Solving from Nature-PPSN V* [M], Berlin Springer, 1998.
- [40]李满林, 杜雷, 闻英友, 王玉娜, 王光兴, 多目标优化遗传算法在移动网络规划中的应用[J], 控制与决策, vol.18, no.4, pp. 441-444, 448, Jul. 2003.
- [41]杨青, 汪亮, 叶定友, 基于多目标遗传算法的固体火箭发动机面向成本优化设计[J], 固体火箭技术, vol. 25, no. 4, pp. 16-20, 2002.

- [42]姚文俊, 遗传算法及其研究进展[J], 计算机与数字工程, vol. 32, no. 4, pp. 41-43, 2004.
- [43]玄光男, 程润伟, 遗传算法与工程优化[M], 北京: 清华大学出版社, 2004.
- [44]徐瑛, 遗传算法的改进及应用实现[D], 北京: 北京理工大学硕士学位论文, Jun. 2003.
- [45]郑立平, 郝忠孝, 遗传算法理论综述[J], 计算机工程与应用, 2003(21): 50-53, 96.
- [46]关志华, 非支配排序遗传算法(NSGA)算子分析[J], 管理工程学报, vol. 18, no. 1, pp. 56-60, Jan. 2004.
- [47]高媛, 卢建刚, 基于非支配排序遗传算法(NSGA-II)的变结构控制方法[J], 中国科学技术大学学报, vol.35, 增刊, pp.328-333, Nov. 2005.
- [48]Tarek M. M. Nasab, "A New Variable Structure Control Design With Stability Analysis for MIMO Systems" [J], *IEEE Trans. on Fuzzy Systems*, pp. 785-788, 2003.
- [49]Shaocheng Tong, HanXiong Li, Fuzzy Adaptive Sliding-Mode Control for MIMO Nonlinear Systems [J], *IEEE Trans. on Fuzzy Systems*, Vol. 11, No.3, pp. 354-360, 2003.
- [50]Wei-Bing Gao, Discrete-Time Variable Structure Control Systems [J], *IEEE Transactions On Industrial Electronics*, 42(2), 117-121. 1995.
- [51]陈潮填, 变结构控制抖振问题的研究及仿真[D], 广州: 广东工业大学硕士学位论文, May. 2004.
- [52]Utkin V. I., Sliding modes and their application in disconuous systems [J], *Automat. Remote Control*, 1974, 21:1898-1907.
- [53]Chan S.P., and Gao W.B., Approach algorithm for smooth tracking with variable structure controller [J], *American Control Conference*, Newyork, pp. 78-88, 1989.
- [54]高为炳, 变结构控制的理论及设计方法[M], 第 1 版, 北京: 中国科学出版社, pp.243-254, 1990

- [55] Yuan Gao, Jiangang Lu, A New Scheme of Function Approximation Based on SVM and NSGA II [J], *ICSCA2006*, accepted.
- [56] G. A. Rovithakis, I. Chalkiadakis, M. E. Zervakis, High-Order Neural Network Structure Selection for Function Approximation Applications Using Genetic Algorithms [J], *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 34, no. 1, pp: 150-158, Feb. 2004.
- [57] Yuh-Jye Lee, Wen-Feng Hsieh, and Chien-Ming Huang. ε -SSVR: A Smooth Support Vector Machine for ε -Insensitive Regression [J]. *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 5, pp. 678-685, May. 2005.
- [58] 陈小平, 于盛林, 刘文波, 基于遗传算法的三次样条函数拟合[J], 数据采集与处理, vol. 15, no. 2, pp. 138-141, Jun. 2000.
- [59] 周鸣争, 基于遗传算法的曲线拟合及应用[J], 安徽机电学院学报, vol. 15, no. 3, pp. 1-5, Sep. 2000.
- [60] 朱国强, 刘士荣, 俞金寿, 支持向量机及其在函数逼近中的应用[J], 华东理工大学学报, vol. 28, no. 5, pp.555-559, 568, Oct. 2002.
- [61] 杨旭, 纪玉波, 田雪, 基于遗传算法的 SVM 参数选取[J], 辽宁石油化工大学学报, vol. 24, no. 1, pp. 54-58, Mar. 2004.
- [62] 赵洪波, 基于遗传算法的进化支持向量机研究[J], 绍兴文理学院学报, vol. 24, no. 9, pp. 25-28, Sep. 2004.
- [63] Vapnik V., Lerner A Pattern Recognition using Generalized Portrait, *Automation and Remote Control*, 1963, 24(6): 774-780.
- [64] Kimeldorf. G., Wahba. G., Some results on Tchebycheffian spline functions, *J. Math. Anal. Applic.*, 1971, 33(1): 82-95.
- [65] V. N. Vapnik, The Nature of Statistical Learning Theory. *New York: Springer-Verlag*, 1995.
- [66] 萧嵘, 王继成, 张福炎, 支持向量机理论综述, 计算机科学, vol. 27, no. 3, pp. 1-3, 2000.

- [67] Vapnik V., An overview of statistical learning theory, *IEEE Transaction on Neural Networks*, Vol. 10, no.5, pp. 988-999, Sep. 1999.

致 谢

值此论文付梓之际，内心无限感慨。回想在攻读硕士学位的过程中，得到了来自多方面的帮助，这些帮助是我完成硕士论文的主要基础。

首先，我要感谢我的导师卢建刚副教授，不论在生活上，学习上，还是科研上，都给予我悉心的关怀和指导。导师敏锐的学术洞察力，渊博的知识，勇于创新的胆识，严谨的科研作风使我受益匪浅，是我终生学习的楷模。

此外，衷心感谢林庆女士在我攻读硕士学位期间在生活学习上所给予的周到、热忱的帮助。

感谢实验室的兄弟姐妹，还有我们 9-6096 寝室同学等对我的帮助和支持，在学习过程中与他们所进行的诸多有益的讨论，使我获益非浅。同时对所有关心和帮助过我的其他老师、同学和朋友表示诚挚的感谢，谢谢你们与我一起度过这段美好的时光。

最后，我要特别感谢我的家人和朋友。一直以来，他们给予我在生活和工作上的理解和关怀，支持和鼓励，这些远非言语所能表达，如果没有他们，我不可能顺利完成这两年多的学习，当然也不会有本文的完成。谢谢！

高 媛

2006 年 2 月

于 求 是 园

作者攻读硕士学位期间完成的学术论文

- [1] 高媛, 卢建刚. 基于非支配排序遗传算法 (NSGA-II) 的变结构控制方法, 中国科学技术大学学报 (自动化专辑), 已发表
- [2] Yuan Gao, Jiangang Lu, "A New Scheme of Function Approximation Based on SVM and NSGA II", The International Conference on Sensing, Computing and Automation (ICSCA2006), Chongqing, China, May 2006. (录用, 待 SCI 收录)

科研情况

◇ “十五”国家科技攻关重大项目 No.2004BA210A01 (2004.10-至今)

项目描述:《科学仪器的研制与开发》

责任描述:多目标优化算法研究

◇ 国家 973 计划项目 No.2002CB312200 (2003.09-2005.12)

项目描述:复杂生产制造过程实时、智能控制与优化理论和方法研究。

责任描述:多目标优化算法研究

◇ 国家自然科学基金项目 No.20206027 (2003.09-2005.12)

项目描述:模拟移动床色谱分离手性药物对映体的应用基础研究。

责任描述:多目标优化算法研究